THE UNIVERSITY *of* EDINBURGH

Addressing Concept Sparsity in Medical Text with Medical Ontologies

Matúš Falis

Doctor of Philosophy

Centre for Doctoral Training in Biomedical Artificial Intelligence
Institute for Language, Cognition and Computation
School of Informatics
University of Edinburgh

2025

Abstract

Clinical Document Coding is the task of summarising unstructured or semi-structured medical text by assigning labels (codes) from a structured knowledge base – *e.g.*, the International Classification of Diseases (ICD) – corresponding to medical concepts, such as conditions or procedures, to clinical documents.

Clinical Document Coding is currently performed by humans. As the task requires time and effort that could be used elsewhere in healthcare, it is desirable to (at least partially) automate it. Hence, a variety of systems have been developed for this task ranging from rule-based to deep-neural-network solutions. Neural solutions often ignore the rich concept representation within the ontology, *i.e.*, the ontological structure, or the code descriptions. Furthermore, while effective in a variety of tasks, neural networks are notorious for requiring large amounts of training data and struggling in low-resource scenarios, unless designed with a focus on low-resource performance.

Concepts in clinical coding follow a big-head long-tail distribution – with few frequent (big head) and a large number of infrequent labels (long tail) – reflecting how common different conditions and procedures are. Few concepts, such as Hypertension and Type 2 Diabetes are very common, while many, such as the Marburg Virus Disease, are rare. This type of distribution is observed in data, with many concepts being infrequent or absent within even the largest publicly available datasets. This data sparsity issue is even more pronounced due to the demanding requirements on the amounts of data for training effective neural network models.

This thesis strives to incorporate the rich ontological structure and background knowledge into the model development and evaluation process for coding discharge summaries with the ICD. The thesis makes the following contributions: (1) a hierarchical evaluation metric; (2) a hierarchical error analysis tool; (3) rule-based data augmentation and synthesis through adjustments to existing texts; and (4) exploration of data augmentation through generating synthetic text via a Large Language Model (GPT-3.5).

The thesis presents hierarchy-aware evaluation approaches. Firstly, Count-Preserving Hierarchical Evaluation (*CoPHE*) compares the number of gold standard labels against predicted number of labels on different levels of the hierarchy. Beyond being able to assign partial credit to mispredictions based on their proximity to gold standard labels within the ontology, CoPHE is also capable of capturing over- or under-prediction within subgraphs of the hierarchy. Secondly, the popular confusion matrix visualisation and analysis approach commonly used in strongly-labelled scenarios was ex-

tended to the weakly-labelled scenario of ICD coding. This approach – Weak Hierarchical Confusion Matrices – allows understanding whether errors in prediction commonly arise due to assigning a different concept from the same family of concepts, or over/under-prediction.

Ontology-guided data augmentation and synthesis was employed to address the data sparsity issue. The thesis explores the possibility of addressing this issue via enhancing concept variability through synonym replacement for relevant concepts identified with pre-existing named entity recognition and linking tools, and introducing concepts previously unseen in the training data.

Finally, the thesis explores the possibility of creating synthetic discharge summaries with the aid of Large Language Models for the purpose of data augmentation for few-shot (appearing rarely in the training data) and zero-shot labels (absent from the training data). GPT-3.5 was prompted to generate discharge summaries based on diagnosis and procedure codes coming from real patient records. In models trained on augmented MIMIC-IV, concepts that appeared within the original training set albeit with few instances were found to have benefited from the further generated data. The method necessitates further refinement to be reliable in the zero-shot scenario. Clinical staff evaluated the generated discharge summaries and compared them to real data with similar labels. Synthetic discharge summaries correctly list individual concepts, but fail to note interaction among them. The resulting overall narrative is thus often flawed. The generated data may be useful for training neural network models, but would not be acceptable in a clinical setting.

In summary, the thesis contributes methods of evaluating performance with regard to the structure of the ontology, and augmentation approaches to mitigating the effects of concept sparsity using both the ontological structure, and the textual descriptions of concepts within the ontology.

Lay Abstract

Clinical Document Coding is the task of summarising medical records by assigning specific codes from a standardised system (*e.g.*, an ontology) like the International Classification of Diseases (ICD). These codes represent medical conditions or procedures.

Currently, human experts perform this time-consuming task. To save time and resources in healthcare, researchers aim to automate this process by developing automated coding systems – such as with machine learning models like artificial neural networks. However, these neural networks often overlook the detailed information within the coding systems – their structure and the codes' descriptions. Additionally, they need large amounts of data to learn from, which is challenging to obtain, especially for rare conditions and procedures.

Similar to their real-life counterparts, some codes are assigned frequently, while others are rare. Some of these concepts may be so rare that they never appear in the data at all. This creates a problem because datasets often lack enough examples of the rare conditions and procedures, making it difficult to train effective neural network models.

This thesis proposes new methods to improve the automation of coding discharge summaries with the ICD with a focus on better use of knowledge in ontologies. The main contributions include: (1) a hierarchical evaluation metric: a new way to evaluate automatic coding systems by considering the hierarchy of medical codes; (2) a hierarchical error analysis tool: a tool to analyze coding errors based on the hierarchy and relationships among codes; (3) rule-based data augmentation: techniques to create more training data by modifying existing texts; (4) synthetic text generation: using a Large Language Model (GPT-3.5) to generate new training data based on a list of conditions and procedures.

The thesis introduces hierarchy-aware evaluation methods, such as Count-Preserving Hierarchical Evaluation (CoPHE), which assesses the accuracy of predicted codes by comparing them at different levels of the hierarchy. It also extends the popular confusion matrix method to the setting of the coding task in order to better understand common errors in coding from the perspective of code groups (or families).

To address data scarcity, the thesis proposes ontology-guided data augmentation. This involves making small changes to the original text by replacing terms with synonyms to enhance variability and introducing new (previously absenting) concepts into

the dataset. Additionally, GPT-3.5 was used to generate new synthetic discharge summaries to supplement training data. While these generated summaries can help train models, they are not yet reliable enough for clinical use due to flaws in the overall narrative.

In summary, the thesis explores new ways to develop better methods for automated Clinical Document Coding – not through introducing new machine learning models, but rather by leveraging knowledge in medical ontologies, in particular the structure of the ontology and the codes' descriptions. It does so by proposing new ways to evaluate models and supplement their learning by generating additional data.

Acknowledgements

'But you've left out one of the chief characters: Samwise the stouthearted. "I want to hear more about Sam, dad. Why didn't they put in more of his talk, dad? That's what I like, it makes me laugh. And Frodo wouldn't have got far without Sam, would he, dad?"' 'Now, Mr. Frodo,' said Sam, *'you shouldn't make fun. I was serious.' 'So was I,'* said Frodo, *'and so I am.'*

J. R. R. Tolkien, The Two Towers

To produce this thesis, as they say, it took a village. To take full credit for it would feel wrong, so before we dive into the technical details, let us briefly appreciate everyone who contributed.

First and foremost, the research presented within this thesis was not conducted alone. I would like to thank Doctor Hang Dong for being my collaborator pretty much from the beginning of the PhD. I would like to thank Aryo Pradipta Gema for being an amazing work spouse, his admirable enthusiasm for working in clinical natural language processing, and especially for taking time from his busy schedule to collaborate with me. I would like to thank my clinical collaborators, Doctor Rose S Penfold, Doctor Michael Holder, Doctor Siddharth Basetti, and especially Doctor Luke Daines for their patience with my crazy ideas and tight deadlines. I would like to thank the non-supervisory reviewers that participated in my annual reviews – Doctor Honghan Wu and Doctor Pasquale Minervini – and the anonymous reviewers who evaluated my manuscripts for all the constructive feedback on my work. Special thanks go to Professor Goran Nenadić and Professor Ian Simpson for examining the thesis and facilitating a fruitful discussion during the defense. I would like to thank my clinical supervisor Professor William Whiteley for enduring my technical jargon and encouraging me to think about the clinical and industrial applications of my project. Above all I would like to thank Doctor Beatrice Alex and Doctor Alexandra Birch for taking me on as a student and sticking with me throughout this adventure. Beyond their patience and excellent academic advice, they often had more kindness and understanding for me than I did. I like to think that under their tutelage I became not only a better researcher, but also at least a bit more human.

Secondly, I would like to thank those who helped me become a researcher in the first place. I thank Doctor Thorsten Merten for letting me work with him on a natural language processing project for two months during my undergrad; Professor Steve

Renals for his great patience during his supervision of my undergraduate thesis; Doctor Brian Mohr for hiring me into my first full-time position as a research engineer; Doctor Keith Goatman for being willing to work with me on medical image analysis despite my expertise being in text; Doctor Alison O'Neil for introducing me to clinical natural language processing in general and ICD coding in particular; and Doctor Aneta Lisowska for being excited about even the tiniest bit of machine learning research we did together and never giving up on me as a friend. I would also like to thank Doctor Krishna Bulusu and Vladimir Poroshin for having me as a research intern at AstraZeneca during my PhD studies; and Doctor Arlene Casey, Doctor Matthew Iveson, and Professor Heather Whalley for their patience with me during the writeup.

The research presented in this thesis was partly conducted during the COVID-19 pandemic. This period was difficult physically, mentally, and emotionally and I believe that I would not have managed to progress within the project without the support of my friends. I would especially like to thank Domas Linkevičius for our morning progress calls and Mariana Ochodková for our weekly check-ins, you guys kept me going. I would like to thank Tomáš Košlab, David Dutko, Martin Opatovský, Filip Brakl, and Michal Ševove for providing distractions throughout lockdowns and beyond and keeping me from going insane.

During the first year of the project I have developed a life-changing health condition (which has since been utilised as an example clinical concept in my publications and this thesis). When the general practitioner phoned – the very visibly unwell – me to drop whatever I was doing and call by the university health center, as they (turns out very correctly) suspected I had a serious problem, I distinctly remember persuading them to give me two more hours, so I could submit a manuscript (which became the first accepted publication of the thesis). I would like to thank the staff at the Richard Verney Health Center and the Royal Infirmary of Edinburgh for putting me back together, being patient with me and available on the phone whenever I had questions, and encouraging me to learn to live with my disease, rather than merely survive.

Speaking of being alive, I would like to thank all the amazing people in the Edinburgh University Swing Dance Society and Edinbop for making sure I got my music fix and enough exercise without seriously hurting myself. I would further like to thank the Informatics Forum Boardgames Committee, especially Leonardo Castorina and Dominik Grabarczyk for our long-term co-operation in getting a bunch of nerds together to have civilised fun on a regular basis.

I cannot imagine how I would conduct this research, write this thesis, hell, have

anything resembling a healthy social life without tea. I would like to thank Mio Shudo for letting me observe her art of tea and learn from her philosophy. The most important part of every tea ceremony, every dance, every piece of research, no matter its successful execution, is to do it from your heart – for our friends, for our family, for people we meet for the first time, for people we might never meet again. To amply thank every person I have had tea with over the course of the PhD individually would probably result in a thesis of its own. Hence, please accept this collective version of my gratitude: Every meeting is a chance; I am glad I took one on you and that you took one on me.

Special thanks go to Ola Olšinová for being the most enthusiastic stylist and tea hobbit, and her endless support of my interests; Martin Krištien and Mattias Appelgren for their many visits and sharing their experience with both the PhD and life in general; Nikita Moghe for our open conversations and consultations on auspicious dates for thesis submission; and Stefi Tirkova for not letting traditions die. Further to this, I would like to thank my colleagues and the management team of the Center for Doctoral Training in Biomedical AI (especially Isabelle Hanlon and Ekaterina Churkina), and my officemates for enduring my special flavour of charisma.

Last but not least, there is family. Family is a core concept within the thesis and appears as a word in some way, shape, or form 146 times in its main body. I would like to thank my family for their support throughout my life, including all of their financial support during my undergraduate studies, learning to accept my prolonged absences, and trying to understand my work.

To all the mad ones.

Mad to live. Mad to talk. Mad to be saved.

Table of Contents

Acronyms

CM *Clinical Modification*. 13, 19, 57

CoPHE *Count-Preserving Hierarchical Evaluation*. 46, 53, 65, 114, 125

CUI *Concept Unique Identifier*. 20, 83

DA *Data Augmentation*. 77

FS *few-shot*. 27, 75

ICD *International Classification of Diseases*. 3, 9, 13, 14, 41, 75, 125, 128

LLM *Large Language Model*. 95, 99, 125, 132

LMTC *Large-Scale Multi-Label Text Classification*. 9, 33, 39, 41, 46, 61, 75, 96, 125

NER+L *Named Entity Recognition and Linking*. 9, 30, 39, 46, 48, 59, 76, 126

NLG *Natural Language Generation*. 97

NLP *Natural Language Processing*. 75, 95

OOF *Out of Family*. 63, 89, 114, 126

PCS *Procedure Coding System*. 13, 19

PLM *Pre-trained Language Model*. 98

SNOMED CT *Systematic Nomenclature of Medicine and Clinical Terms*. 20–22

UMLS *Unified Medical Language System*. 19, 21, 22, 80

WHCM Weak Hierarchical Confusion Matrices. 46, 67, 114, 125

ZS *zero-shot*. 27, 75

Chapter 1

Introduction

Medicine is essential to the functioning of society. Its primary aim is to study and improve patients' health so they may live longer, healthier, and more productive lives. Failure to provide ample aid to the patient may lead to diminishment or loss of bodily or mental function, or the death of the patient, and hence the stakes are high.

Communication plays an important role within medicine – be it for recording patient history, communication between specialists, or for the purposes of drawing statistics of care. To do this effectively, communication protocols and efficient data standards have been developed – *e.g.*, the DICOM format for radiology scans (Bidgood Jr et al., 1997). A large proportion of the communication of medical data is presented in the form of natural (human) language, and most often recorded as text. Historically, this has been done via handwritten notes, but with the development of hospital system infrastructure, Electronic Health Records (EHRs) are becoming the new standard. The free-text component of an EHR contains a plethora of information about the patient that may be absent or recorded in a limited fashion in the structured parts of the EHR (Wu et al., 2022). Some of these documents may come from a particular test/specialisation *e.g.*, radiology reports. On the other hand, *discharge summaries* focus on the larger scope of a patient's journey during a hospital stay. They can be seen as semi-structured with sections or paragraphs addressing the relevant aspects of admission, diagnosis, treatment, and discharge (*e.g.*, history of present illness, discharge diagnoses, discharge instructions).

A discharge summary is dense with useful data – beyond the presenting complaints and diagnoses at discharge they can contain test results, prescription data, and an overall narrative of the patient's stay. The need to facilitate efficient data recording and communication led to the development of medical coding systems, such as the *Inter-*

national Classification of Diseases (ICD), and the task of medical document coding.

The ICD is an ontology of clinical conditions and procedures presented within a tree structure. Each concept is represented as a node with the concept's code and verbal description and can have outgoing edges to descendant nodes. Concepts on higher levels (closer to the root of the tree) are abstract (*e.g.*, *mental disorders*, *neoplasms*, *operations on the endocrine system*), and further expanded by their descendant concepts (*e.g.*, *organic psychotic conditions*, *other psychoses*, *neurotic disorders, personality disorders, and other nonpsychotic mental disorders*, and *intellectual disabilities* descend from *mental disorders*). These expansions can happen over multiple levels until a leaf node (one without further descendants) is reached. An example of a path from the root to a leaf code (*290.20: Senile dementia with delusional features*) in clinical modification of the ninth revision of the ICD (ICD-9-CM) is presented in Figure 1.1

A coding system, alongside coding guidelines, is used by a specially-trained human coder to produce structured data based on the free-text discharge summary. Starting from the semi-structured or unstructured text, the coder assigns one or more codes from a coding system (corresponding to relevant concepts *e.g.*, diagnoses or procedures) on the document level (Dong et al., 2022a). In practice this means that the output of a coder's work is a list of codes attached to the document indicating the presence of the corresponding concepts within the case and the document. Presence here does not mean mere mention within the text, as mentions can be negated, stated as normal/unremarkable, or be irrelevant to the case. Hence, the coding task is not one of diagnosis, but rather of summary of concepts relevant to the case in a structured format (Dong et al., 2022a). This structured information is easier to store, move, and draw statistics from. ICD codes have been used for a variety of purposes, including reimbursement in certain healthcare systems, performance comparison, and semi-automation of communication with patients (*e.g.*, for contacting at-risk patients during the COVID-19 pandemic).

Clinical coding is error-prone and laborious (Dong et al., 2022a). The human resources spent on the task could be used elsewhere within the healthcare system. For these reasons, the involvement of machine learning in the process of clinical document coding has long been debated (Stanfill et al., 2010). Thanks to the transition to EHRs with readily machine-readable text (without the necessity of a translation step from hand-written to digital form through optical character recognition) hospitals produce a large amount of data that could be used for training modern data-driven solutions based on neural network models. However, access to data is not trivial, as the privacy

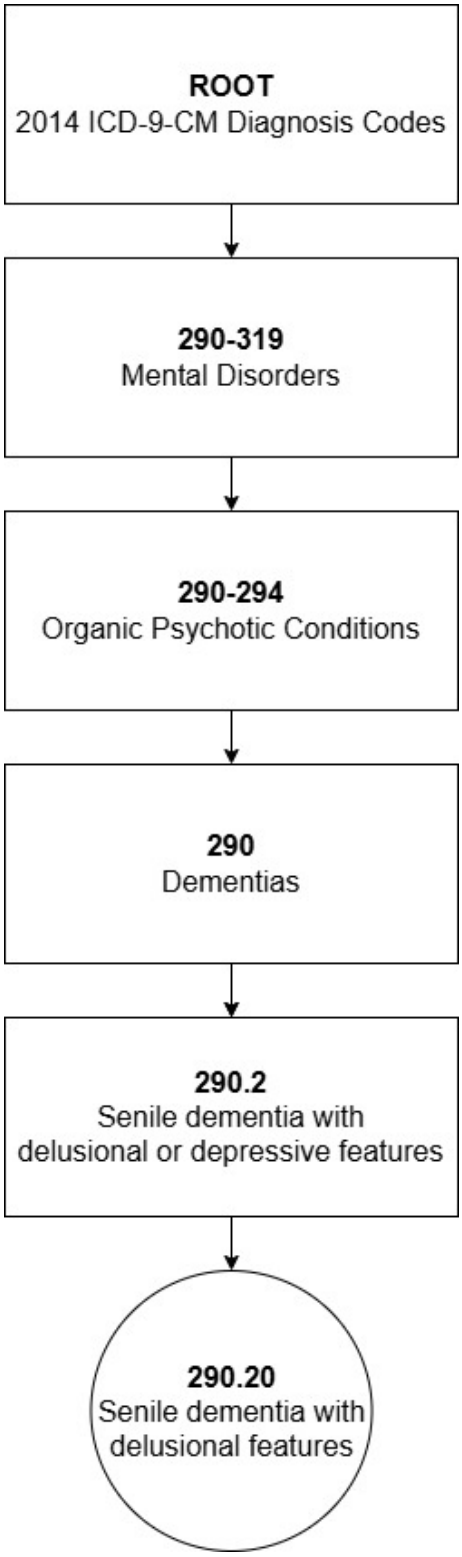

Figure 1.1: The path from the root of ICD-9-CM to the leaf code *290.20: Senile dementia with delusional features*. Rectangular nodes represent non-leaf nodes which can have further descendant nodes not included in this path.

of a patient's medical record is protected by law. This leads to restrictions on access to EHRs and the need for de-identification procedures (Kovačević et al., 2024). Hence, despite a large amount of relevant data existing, the task constitutes a low-resource scenario (Wu et al., 2022). What is more, medical ontologies, such as the ICD, contain a vast array of concepts following a big-head long-tail distribution that are relevant for medical coding, but may be infrequent or absent from the limited amount of data available (Rios and Kavuluru, 2018a). This can be due to the data collection parameters – *e.g.*, differences of frequency of certain conditions given the location from which the data was sourced (*e.g.*, shark bites in Switzerland), the timeframe of data collection (*e.g.*, absence of COVID-19 data before the COVID-19 pandemic), rare conditions not encountered during collection, or rare enough that they would identify the patient making retaining access to the exact condition while de-identifying patient-identifiable data impossible.

Facilitated by the release of the ICD-9 coded MIMIC-III dataset (Johnson et al., 2016), followed by ICD-9 and ICD-10 coded MIMIC-IV in 2023 (Johnson et al., 2023), automated ICD coding with neural models has received much attention in recent years (*e.g.*, Mullenbach et al. (2018); Chalkidis et al. (2019b); Vu et al. (2020a)). These early attempts use the ontology primarily as the label space in a large-scale multi-label classification task, with some integration into the models' architecture. The aim of this thesis is to investigate the potential of further integration of external knowledge stored in medical ontologies within the task of ICD coding into the model development process – particularly training and evaluation. To achieve this we utilise the ontology beyond the mere function of a flat label space by employing the ontological structure, the standard descriptions of concepts within the ontology, and the connection of the ICD with other medical ontologies, such as the UMLS or Snomed CT.

The contributions of this thesis are:

1. CoPHE – a hierarchical evaluation metric tracking counts of predictions and gold standard labels on different levels of the hierarchy for large-scale multi-label text classification capturing over- and under-prediction within scenarios with hierarchical label spaces.

2. WHCM – a weak hierarchical confusion matrix analysis tool for ICD coding allowing the analysis of co-occurrence of incorrect predictions with missed gold-standard labels within and outside of code families. Similar methods were developed by other researchers in different domains in parallel.

3. A rule-based data augmentation and synthesis method combining pre-existing named entity recognition and linking engines with term replacement based on ontologies in order to (1) provide a variety of verbal representations for clinical concepts already present within a training set with the use of synonyms; and (2) introduce concepts missing from the training data by replacing common yet less specific concepts with related uncommon specific ones.

4. An exploration of viability of the Large Language Model GPT-3.5 in the context of ICD-10 coding as a model performing the coding task; its usefulness as a generator of synthetic discharge summaries with a focus on uncommon labels for augmenting training sets of local neural network models; and the quality of the generated discharge summaries according to clinical professionals with experience in writing such documents.

The thesis is structured as follows: Chapter 2 provides a comprehensive background on the concepts relevant throughout the thesis – medical document coding, medical terminologies and ontologies, medical document coding datasets (MIMIC-III and MIMIC-IV) along with common issues appearing in them, the tasks of Named Entity Recognition and Linking and Large-Scale Multi-Label Text Classification along with methods developed for these tasks in the clinical domain. Chapter 3 presents existing LMTC evaluation approaches, discusses issues within them, and introduces two evaluation approaches developed as part of the thesis – Count-Preserving Hierarchical Evaluation (published in Falis et al. (2021)) and Weak Hierarchical Confusion Matrices (published in Falis et al. (2022)). Chapter 4 presents background on data augmentation in natural language processing and introduces ontology-driven rule-based data augmentation (synonym replacement for relevant medical concepts) and synthesis (further specifying of "unspecified" concepts) (published in Falis et al. (2022)). Chapter 5 extends the research conducted on rule-based data augmentation and synthesis to natural language generation with Large Language Models (published in Falis et al. (2024)). The chapter begins with a background on language modelling with neural networks, with a focus on Pre-trained Language Models, and Large Language Models. Further details are provided on Large Language Models in clinical natural language processing applications. The chapter then presents our exploration of GPT-3.5 in the context of ICD-10 coding as (1) a generator of synthetic discharge summaries in order to augment the training data for local neural network models and (2) as a model to directly perform ICD-10 coding (comparing performance on real and self-generated data). Furthermore,

the quality of the synthetic data was evaluated by clinical experts with experience in medical document coding. Chapter 6 summarises the thesis' conclusions, discusses the potential applications and ethical implications of our research, and suggests directions for future work.

Chapter 2

Background

This chapter provides a thorough background to the task of automated clinical document coding. Section 2.1 introduces the general task of medical document coding as it is currently preformed by humans. Section 2.2 discusses medical terminologies and ontologies, with a spotlight on the different ontologies relevant to the thesis, especially the *International Classification of Diseases* (ICD). Section 2.3 introduces the datasets used within the thesis (MIMIC-III and MIMIC-IV), comments on common issues within these datasets, and lists datasets corresponding to similar tasks in non-medical domains. Section 2.4 describes the strongly-labelled task of *Named Entity Recognition and Linking* (NER+L) along with existing rule-based, neural, and hybrid methods. Finally, Section 2.5 outlines the weakly-labelled *Large-Scale Multi-Label Text Classification* (LMTC) task – the core task towards which this thesis contributes – in contrast with NER+L and provides a high-level description of relevant neural network approaches to medical document coding. The Chapter does not provide background on Data Augmentation and Large Language Models – in-depth background on these topics can be found in Chapters 4 and 5 respectively.

2.1 Medical Document Coding

Medical document coding is the task of assigning structured codes from a medical ontology – such as the *International Classification of Diseases* (ICD) – to clinical documents. The codes are assigned based on the mentions of their corresponding concepts (such as specific procedures or conditions) within the document indicating their presence and relevance to the case (as opposed to negative mentions indicating the absence of a condition, *e.g.*, "patient denies abdominal pain") on the document

level, rather than identifying the concepts' individual mentions. The task is not one of diagnosis, but effectively summarisation of unstructured/semi-structured text into a structured format of one or more codes. The advantages of the structured format are its brevity (which leads to easier data management, and comparison between documents), searchability (through a reduced search space in a standardised terminology), simplified cohort building, and, in cases of label-spaces common across different nations and health services, international comparison. These structured representations serve as an easier means to store, process, maintain, and communicate data on patients' conditions and hospital stays. Medical document coding is important not only for administrative, financial, and statistical purposes, but also for cohort building and patient screening (such as in the case of assignment to vulnerable groups during the COVID-19 pandemic).

The task of assigning codes from ontologies representing medical concepts is currently performed by specially-trained human professionals known as coders. At the early stages of the project members of clinical staff in National Health Service (NHS) Lothian were consulted regarding the task – what it entails, common issues, and potential avenues of improvement – as part of a review on automated clinical document coding (presented in Dong et al. (2022a), not considered part of the thesis). The task is laborious and time-consuming. Within NHS Scotland a clinical coder usually manages to code about 60 cases per day, taking roughly 7-8 minutes per case (Dong et al., 2022a). A department of 25-30 coders at this rate manages to code about 20,000 documents per month (Dong et al., 2022a). Despite this effort, similar to radiology, there is a risk of backlogs forming due to discharge summaries being generated faster than coded with some cases waiting to be coded several months (or even more than a year, as presented by Alonso et al. (2020)). This issue becomes more pressing considering the trend of aging population in many countries, especially in China, Japan, the United States, and countries in Europe, but to a lesser degree worldwide (Sardanelli, 2017). Furthermore, the task of manual coding is also error-prone. This may stem from incompleteness of patient records, data entry errors, subjectivity of choice when assigning codes, or insufficient expertise (Dong et al., 2022a). For these reasons, the involvement of machine learning in the process of clinical document coding has long been debated (Stanfill et al. (2010) report Dinwoodie and Howell (1973) as the earliest attempt at the task, with more consistent development starting in the mid-1990s).

2.2 Medical Terminologies and Ontologies

Classification systems have a substantial history in the field of medicine. A notable early example is William Farr's (1807 – 1883) "statistical nosology" – a system which defined diseases not only by category, but also by synonymy and "provincial terms" representing the local names of diseases. Farr's nosology is considered a predecessor of the International Classification of Diseases (ICD) (Eyler, 1979). Over the past two centuries several new vocabularies were introduced defining signs, symptoms, and other manifestations of disease.

Ontology is a philosophical discipline whose object of study is the nature of existence. It aims to understand how the world and things within it fall into various categories, and how related these categories are. In the context of informatics (more commonly referred to as computer science), the Semantic Web community defines an ontology as a formal explicit specification of a shared conceptualisation of a domain of interest (*e.g.*, healthcare, legislation, or industry) (Grimm, 2009).

Ontologies provide classifications of entities within their respective domain. An entity is said to make up a term of the ontology. Additionally, unlike a simple vocabulary, the ontology must provide a structure of semantic relations between its terms. Hence the primary purpose of an ontology within natural language processing is to provide a standardised structured vocabulary for a given field. The most common structure of ontologies in the field of biomedicine is a *directed acyclic graph* – a graph whose edges are directed (as opposed to lacking direction implying being trivially bidirectional) and form no cyclical substructures – for all paths within the graph, no path visits the same vertex more than once. Some ontologies, such as ICD-9, further restrict themselves to a tree structure.

Ontologies are designed such that vertices represent concepts while edges represent relations (*e.g.*, the "Is A","Causative agent", "Procedure Site" relations in SNOMED CT). A common relation type is that between an ancestor (or *parent*; more general) concept and their descendants (more specific) concepts – this type of relationship is sometimes referred to as an "Is A" relationship in the sense that a descendant concept is an instance of the ancestor concept (*e.g.*, "Diabetes mellitus (disorder)" is a "Disorder of glucose metabolism (disorder)" in SNOMED CT). Within the general context of a graph the ancestor-descendant relationship can be represented by an edge going from the ancestor vertex (or node, or, in the context of an ontology specifically, concept) to the descendant vertex with the corresponding directionality. A terminology is a special

case of an ontology where the relationships between concepts are limited to "Is A" type relations. As ICD-9 and ICD-10 are limited to "Is A" relations leading to them being structured as trees, these ontologies are in fact terminologies (however, in the thesis they are referred to by the more generic term "ontology"). Within the context of the tree structure such a representation leads to the nodes closer to the root of the ontology representing more general concepts, while the most specific concepts appearing at the tree's leaves. As a result, many ontologies in biology or medicine follow the *true-path rule*, by which labelling of terms occurs on the most specific level possible, but annotation is also implicitly assumed for all the ancestors of the labelled term – *e.g.*, a mention of a heart attack can be labelled as a myocardial infarction, but is also a case of the increasingly more broad terms of heart disease, cardiovascular disease, and disease in general. One of the focuses of this thesis is the utilisation of the ontological structure and it uses the following terms in relation to the ontological structure: a *family* of concepts refers to a set of concepts that share a common ancestor concept, often referred to as the family of the ancestor concept; concepts sharing their direct ancestor (meaning they are children of the same concept) are referred to as *sibling* concepts.

Ontologies differ in structure and granularity. These differences can be motivated by the purpose of the ontology. For instance, an ontology intended for description of disease by physicians (*e.g.*, SNOMED CT) necessitates great detail, while systems developed for global statistics or billing purposes (*e.g.*, ICD-9) do not require such granularity. As multiple ontologies may focus on a common set of concepts, mappings exist between ontology pairs allowing translation of a concept from a source ontology to a target ontology. Due to differences in structure and (especially) granularity, however, one-to-one mapping (where each concept in a source ontology has a unique counterpart in the target ontology and vice-versa) may not be possible, resulting in a single concept in one ontology potentially mapping to more than one in another.

The following subsections discuss three notable ontologies directly involved within the research of the thesis. The Unified Medical Language System is introduced as, beyond being an ontology, it also contains a component that allows mapping between ontologies which was used within the thesis. A further subsection is dedicated to other notable biomedical ontologies or similar taxonomies in non-medical domains that have been known to be used in related research.

2.2.1 International Classification of Diseases

The *International Classification of Diseases* (ICD) is the foundation for identification of trends and statistics of health globally, and an international standard for reporting medical conditions. It is a diagnostic standard for clinical and research purposes. Its contents are not limited to diseases, but health conditions in general (including disorders or injuries), and medical procedures. These terms (or codes) are structured in a comprehensive hierarchical fashion. Coding medical documents with the ICD allows for easy storage, retrieval and analysis of medical information (*e.g.*, in cohort building); sharing and comparison of healthcare statistics between institutions, geographic regions, or settings; or comparison within an institution across time.

National extensions to iterations of the ICD were developed to suit purposes of different countries. Notably, the *Clinical Modification* (CM) to the ninth iteration (ICD-9) was developed in the United States to support morbidity coding for reimbursement and other purposes. ICD-9-CM was later replaced by a clinical modification applied to the ICD-10 (ICD-10-CM). Further notable extensions are the *Procedure Coding System* (PCS) code sets (ICD-9-PCS and ICD-10-PCS) comprising codes for clinical procedures, such as different surgical procedures, radiological scans, or radiotherapy.

A *revision* is a major rework of the ontology. Revisions may include major changes, *e.g.*, adjustment to the general structure of the ontology, concept reference conventions (*e.g.*, the core ICD-9 codes are numeric-only with only two supplementary chapters being distinguished by a leading letter, while all ICD-10 codes are alphanumeric), or suffix patterns (*e.g.*, introduction of laterality into the code and its description within ICD-10). Over the course of its existence (more than a century) the ICD has undergone several revisions, most serving as the standard for roughly a decade with the latest two legacy iterations – ICD-9 and ICD-10 – being the standard for longer (ICD-9 was published in 1977, followed by ICD-10 in 1992 (Hirsch et al., 2016)). The latest (eleventh) revision – ICD-11 – was adopted by the Seventy-second World Health Assembly in May 2019 and came into use on the first of January 2022.

Minor amendments, such as an addition of a new code (*e.g.*, for COVID-19), removing/merging a code, or relocating an existing code within a pre-existing ontological hierarchy are published as *editions* (or versions) of a revision. These are typically released on an annual basis. Support of such editions may continue for an older standard – *e.g.*, the latest edition of ICD-9's Clinical Modification (ICD-9-CM) was

released in 2011[1], over a decade after the introduction of ICD-10.

With its progressive improvement based on clinical input, research, and epidemiology, the ICD ontology has become suitable for use in different settings, such as tracking the cause of death, recording rare diseases, or as part of the billing process.

The task of automated medical document coding is explored within the thesis in the space of the ICD – in particular the 9th and 10th revisions – and the ontology is utilised in the thesis' experiments. The methods developed as part of the thesis in order to evaluate medical document coding models and address concept sparsity within medical document coding were designed with the two aforementioned revisions of the ICD in mind, but can be translated into other label spaces.

2.2.1.1 Ninth Revision

The 9th revision of the International Classification of Diseases (ICD-9) was adopted in 1979 and replaced by the 10th revision in 1999. The Clinical Modification (ICD-9-CM) is an adaption developed by the U.S. National Centre for Health Statistics (NCHS) used for the assignment of diagnosis and procedure codes on medical documents within the United States. ICD-9-CM consists of ∼13,000 codes divided into 17 chapters with 2 supplementary classification chapters for *Factors Influencing Health Status And Contact With Health Services* ("V"-classification) and *External Causes Of Injury And Poisoning* ("E"-classification) (Cartwright, 2013).

Diagnosis codes from the core 17 chapters are fully numeric with a three-cipher basic structure (referred to within the thesis as *head*) describing the condition's category. Heads from "V"-classification are indicated by a starting "V" followed by two ciphers, while "E"-classification heads (*e.g.*, *E880.1: Accidental fall on or from sidewalk curb*) use a starting "E" followed by three ciphers. Head codes can have further expansion into more specific concepts indicated by a *suffix* consisting of up to two ciphers. The head code and suffix are clearly separated using a decimal point ("."). The code suffix describes the aetiology (a cause, or a set of causes of the disease), anatomical site, or manifestations of the condition. Note that some head codes have no further descendants and hence are themselves leaves without further expansions via suffices (*e.g.*, *319: Unspecified intellectual disabilities*). The anatomy of an example ICD-9 code is presented in Figure 2.1. Procedure codes are fully numeric, with the code consisting of a two-cipher head, a suffix of up to 2 ciphers. The head and the suffix (if present)

[1] https://ftp.cdc.gov/pub/Health_Statistics/NCHS/Publications/ICD9-CM/2011/

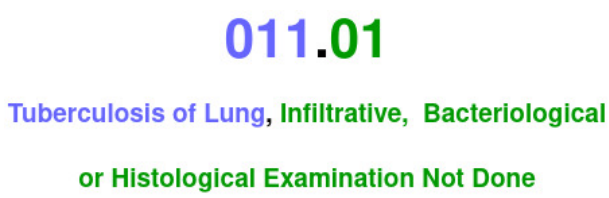

Figure 2.1: Structure of a sample ICD-9 code. The code consists of a head describing the disease (here in blue), and a suffix (here in green) providing supplementary information about the aetiology, or anatomy. These two parts of the code are clearly divided by a decimal point. More abstract (higher-level) codes can exist without a suffix. There are several codes with the head *011: Pulmonary Tuberculosis* descended from this concept and differentiated by the suffix.

are separated by a decimal point (“.”).

Suffix patterns exist within ICD-9 where the suffix “.9” usually signifies an *unspecified* aetiology, and “.8” usually signifies *other* aetiology – meaning the disease is specified within the text, but does not match any of the existing specific aetiology descriptions within the ontology – *e.g.*, the head code *162: Malignant neoplasm of trachea bronchus and lung* includes five descendant codes with a specific anatomy mentioned in the description (*trachea*; *main bronchus*; *upper lobe, bronchus or lung*; *middle lobe, bronchus or lung*; *lower lobe, bronchus or lung*); one code for any other anatomically relevant site (*other parts of bronchus or lung*); and one code for the option that the specific site was not provided (*bronchus and lung, unspecified*). Sometimes these special aetiologies deviate from the described patterns – *e.g.*, they are merged into a single one (*e.g.*, *413.9: Other and unspecified angina pectoris*) or they are presented in tandem while describing only one of the patterns’ meanings (*e.g.*, *482.89: Pneumonia due to other specified bacteria*, which is descended from code *482.8* with the same description, which has four more descendants further described via specified bacteria). Furthermore, the associated special keywords (“unspecified” and “other”) may be present in different parts of the description, while the suffix of the code describes only one – *e.g.*, the head code *215: Other benign neoplasm of connective and other soft tissue* contains the keyword “other” twice (without a reference to “.8” trivially, as it is a head), and has a descendant *215.8: Other benign neoplasm of connective and other soft tissue of other specified sites* adding a third “other” upon the dimension the suffix describes – the site of the neoplasm.

While ICD-9 is an outdated standard, its main advantage was the availability of

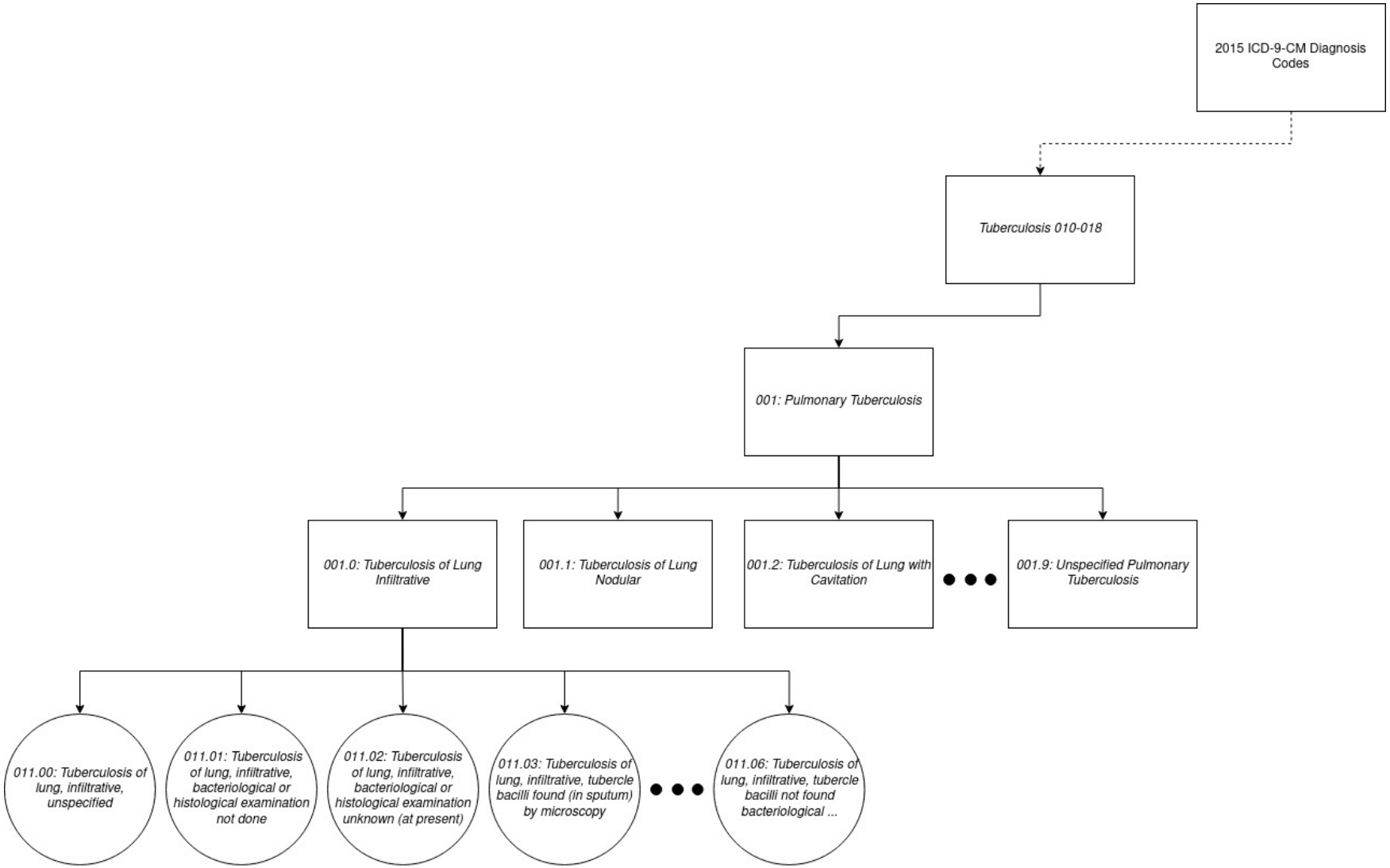

Figure 2.2: The tree structure of the ICD-9 presented on a subset of nodes relating to Tuberculosis. Non-leaf nodes are represented as rectangles, leaf-nodes are represented as circles.

datasets – large ICD-10 coded datasets, such as MIMIC-IV were released only relatively recently (note that ICD-10-CM replaced ICD-9-CM as the coding standard in October 2015 as reported by Hirsch et al. (2016)). Hence, a significant portion of prior work on neural ICD coding was conducted on ICD-9 data (MIMIC-III). This includes the research reported in Chapters 3 and 4. However, the core features of ICD-9 are retained by ICD-10 and many of the developed methods developed on ICD-9 (evaluation approaches reported in Chapter 3) have been translated to ICD-10 as part of the thesis' experiments on MIMIC-IV. Others (rule-based data augmentation and synthesis reported in Chapter 4) while not translated as part of the work on the thesis, can be adjusted to work with ICD-10.

2.2.1.2 Tenth Revision

The 10$^{\text{th}}$ revision of the International Classification of Diseases (ICD-10) was adopted in 1998 and was the global standard until the adoption of of ICD-11. The Clinical Modification of ICD-10 (ICD-10-CM) is far more detailed than ICD-9-CM with $\sim$70,000 codes (up from $\sim$13,000) – one contributing factor to this significant increase is the addition of greater specificity of laterality (specification of right or left side *e.g.*, *S72.431: Displaced fracture of medial condyle of right femur* and *S72.432: Displaced fracture of medial condyle of left femur*). These codes are divided into 21 chapters similar to the 17 chapters in ICD-9, albeit in a different order. The supplemental classification from ICD-9 ("E", and "V" codes in ICD-9) is incorporated into the main classification. Injuries are classified primarily by anatomical site, and secondarily by type. The codes are up to 7 alphanumeric characters long (not counting the decimal point) and start with an alphabetical character clearly indicating the chapter. The patterns of ".8" and ".9" suffices describing the "other" and "unspecified" aetiology respectively is retained from ICD-9. Similar to ICD-9, while ".8" and ".9" are commonly used to indicate these concepts, deviations from these patterns exist through different levels of the same keyword concept (*e.g.*, S52.30 Unspecified fracture of shaft of radius with the descendant S52.309 Unspecified fracture of shaft of unspecified radius), or a local pattern – *e.g.*, S52.32 Transverse fracture of shaft of radius has direct 6 descendant codes S52.32x (where x is in $\{1, 2, 3, 4, 5, 6\}$) and individual descendants indicate the laterality – left, right, unspecified; and displacement status – displaced/non-displaced (note that in none of these the cipher 9 appears).

Procedure codes in ICD-10 consist of 7 alphanumeric character and do not include a decimal point that would divide the code into a head and a suffix. Each character

contains up to 34 possible values (all alphanumeric except "I" and "O" to avoid confusion with "1" and "0"). Each value represents a specific option given its position. Each position has its own semantic role given its presence within a the section of the ICD-10-PCS (indicated by the first character). Thus, the semantic roles of characters 2-7 are common among codes belonging to the same section, but may differ from the semantic roles of characters 2-7 of codes from other sections. For instance, for section 0 *Medical and Surgical* the seven characters correspond to: Section, Body System, Root Operation, Body Part, Approach, Device, and a Qualifier respectively; whereas for section C *Nuclear Medicine* they stand for: Section, Body System, Root Type, Body Part, Radionuclide, and 2 instances of Qualifier.

Within the context of the thesis, ICD-10 has been used in Chapter 5 within MIMIC-IV. Evaluation approaches described in Chapter 3 were also translated into ICD-10 so as to facilitate evaluation within Chapter 5.

2.2.1.3 Eleventh Revision

Beyond the typical changes between revisions (restructuring or code syntax conventions), ICD-11 introduces three major adjustments: the foundation component, post-coordination, and a digital-friendly design (Fung et al., 2020).

The foundation component is built on an underlying knowledge base intended to be updated continuously, rather than in annual editions. The foundation component allows multi-parenting, where a concept can have multiple direct ancestors, neither of which needs to be an ancestor of the other. A particular code hierarchy or "linearisation" can be derived from the foundation component for a given purpose, such as a country-specific modification, or a specialty subset. Such hierarchies are referred to as linearised derivatives of the foundational component. These linearised derivatives follow a single-direct-parent hierarchy akin to ICD-9 and ICD-10, and, once adopted by a particular healthcare system are intended to be updated periodically and versioned.

Post-coordination allows addition of further context through code sequences delimited with special symbols. Multiple core (or "stem") concepts can be combined using the "/" character – *e.g.*, urinary tract infection due to extended spectrum beta-lactamase producing Escherichia coli being coded as GC08.0/MG50.27, where the "/" character is used to combine the core concepts *GC08.0 Urinary tract infection, site not specified, due to Escherichia coli* and *MG50.27 Extended-spectrum beta-lactamase producing Escherichia coli*. Extension codes can be applied to a leading stem code in order to provide further specificity. This is indicated by inserting an "&" character be-

	ICD-9-CM	ICD-10-CM/PCS	ICD-11 (foundation)
# of Diagnosis Codes	14,025	69,823	
# of Procedure Codes	3,824	71,924	
# of All Codes	17,849	141,747	14,622

Table 2.1: A comparison of the number of leaf-level concepts of the legacy standards ICD-9-CM, ICD-10-CM/ICD-10-PCS, and the current standard ICD-11.

tween the leading code and the extension code – *e.g.*, more precise anatomy in the case of tuberculosis of prostate can be coded as 1B12.5&XA63E5 (*1B12.5: Tuberculosis of the genitourinary system*; *XA63E5: Prostate gland*) (Fung et al., 2020).

Finally, while ICD-9 and ICD-10 exist as digitised resources, these were created post-hoc, while ICD-11's design involved digital utility, including ontology browsers, and coding tools.

In terms of size, ICD-11 has 14,622 leaf codes, a sizable increase compared to ICD-10's 10,607. While this is significantly fewer than the size of ICD-10-CM (69,823 leaf codes), due to ICD-11's increased expressivity, if an ICD-11-CM (or a similar national-level version) is released, it will likely be defined via a subset of the possible leaf-code combinations enabled through the postcoordination feature, rather than as an extension of the standard itself.

ICD-11 was not used within the thesis, due to lack of available ICD-11-coded data at the time of method development and experimenting. However, similar to the translation of the methods from ICD-9 to ICD-10 conducted as part of the thesis, the methods presented within the thesis should be applicable to linearised derivatives of the ICD-11.

2.2.2 Unified Medical Language System

The *Unified Medical Language System* (UMLS) (Humphreys and Lindberg, 1989) is a project of medical terminology originally released in 1990. The core components of the UMLS are the *Metathesaurus* containing various medical vocabularies and mappings between them, a *Semantic Network* representing the connections between the terms, and the *SPECIALIST lexicon* – a syntactic lexicon of biomedical and general English designed to support the UMLS SPECIALIST Natural Language Processing system which incorporates the other components of the UMLS, including the Metathesaurus, the Semantic Network, and the named entity recognition tool MetaMap (Aron-

son and Lang, 2010) designed for recognising UMLS concepts in free text.

While the ultimate beneficiaries of the UMLS are health professionals, it was designed specifically with system developers in mind (Amos et al., 2020). The development strategy assumed that relevant information would continue to be distributed across disparate databases (*e.g.*, bibliographic databases, patient record systems, or knowledge bases). This can be contrasted with ICD or Medical Subject Headings (MeSH) (Lipscomb, 2000), which were primarily designed to promote consistent worldwide reporting, and to reflect medical concepts appearing in literature respectively. Search was considered to be a major use case for the UMLS even at the time of its initial release. The system has been incorporated into several information retrieval tools, such as MetaMap (Aronson and Lang, 2010), SemEHR (Wu et al., 2018), or Apache cTAKES (Savova et al., 2010). Furthermore, the Metathesaurus component provides connection with other ontologies, enabling mapping of concepts in one medical ontology to another through the concept's *Concept Unique Identifier* (CUI) – one of the UMLS's most utilised features (Amos et al., 2020). Examples of such mappings are the SNOMED CT to ICD-9, and SNOMED CT to ICD-10 maps curated by the UMLS.

Within the context of the thesis the UMLS was used as part of pipelines involving information retrieval tools SemEHR (Wu et al., 2018) and MedCAT (Kraljevic et al., 2019) – the retrieved data are linked to CUIs. These CUIs were further translated into ICD-9 through mappings. Furthermore, the links between the UMLS and other ontologies have been used for collecting various (synonymous) surface-form representations for a given concept as part of data augmentation/synthesis.

2.2.3 Other Notable Ontologies

This subsection briefly introduces notable otntologies and non-biomedical hierarchical label spaces which are not directly used within the thesis, but have features similar to the utilised ontologies. These similarities can be exploited to transfer the methods developed throughout the thesis in future work.

2.2.3.1 SNOMED CT

The *Systematic Nomenclature of Medicine and Clinical Terms* (SNOMED CT)[2] is an international terminological standard, and also one of the terminological backbones of the Observational Medical Outcomes Partnership. SNOMED CT started as the

[2] https://www.snomed.org/

Structured Nomenclature of Pathology (SNOP) in 1965. Predecessors of SNOMED CT used a system of self-standing axes to describe its concepts – these axes could be combined into composite codes (Bodenreider et al., 2018). SNOMED Reference Terminology (SNOMED RT) abandoned this system in favour of a description logic formalism called Ontylog based on the Knowledge Representation System Specification syntax and the K-REP system (Spackman et al., 1997). SNOMED CT as a successor of SNOMED RT has continued to use this description logic as its underlying representation.

Since its initial release in 2002, SNOMED CT has been updated twice a year. As of January 2018 SNOMED CT has 341,000 active concepts, 1,062,000 relationships, and 1,156,000 descriptions – capable of more expression than the ICD. The largest families of concepts in SNOMED CT are disorders, procedures, body structures, clinical findings other than disorder, but also organisms. The design philosophy of SNOMED CT is to keep concept expressions simple enough to be broadly usable by clinicians, while at the same time maintaining a faithful representation of the concepts' meaning.

2.2.3.2 Orphanet Rare Disease Ontology

Orphanet Rare Disease Ontology (ORDO) (Vasant et al., 2014) is an ontological representation of the data within the Orphanet information system (Weinreich et al., 2008). Orphanet is a disease-centric database of rare diseases and orphan drugs (developed specifically for treating rare diseases). The latest release of ORDO at the time of writing (version 4.4) includes over 15,000 classes. Beyond a classification of rare diseases, ORDO concepts also include information on phenomes, diseases, genes, genetic inheritance mode and prevalence. ORDO links with a number of other terminologies (*e.g.*, the UMLS, SNOMED CT), databases (*e.g.*, ensembl (Harrison et al., 2024)) and classifications (ICD-10). These links have been utilised in previous work in medical coding (Dong et al., 2021b).

ORDO is not directly used within the thesis, but is included in this background section due to its relevance to rare concepts in medical text.

2.2.3.3 Human Phenotype Ontology

The *Human Phenotype Ontology* (HPO) conceptually differs from the previous examples in that it is an ontology of phenotypes. As of September 2020, the HPO contained 15,247 terms. In 2014 when the HPO's coverage of phenotypes was compared to the

UMLS, it was found that UMLS resources covered only about 35% of terms in the HPO (Köhler et al., 2021) – this led to the incorporation of the HPO into the UMLS.

Similar to the ICD, the HPO organises its terms in a tree structure following the *true-path rule* and provides a brief description of each term. Unlike the ICD, it provides further associations within the ontology beyond the basic tree structure – associations to diseases, and genes. For instance, there is a single path in the HPO hierarchy to the term "Cachexia" – a severe form of weight loss and muscle wasting – yet this term is linked to further 73 disease terms and 60 gene terms. Some of the disease associations automatically carry a link to gene terms – *e.g.*, Cachexia is associated with Majeed Syndrome, which in turn is associated with the gene LPIN2.

The HPO also contains synonyms for terms, and references to respective terms in other terminologies or ontologies (MeSH, SNOMED CT, UMLS). Furthermore, HPO associations link to frequency and onset information of signs and symptoms of rare diseases through ORPHA codes from the Orphanet nomenclature of rare diseases.[3]

The HPO is not directly used within the thesis, but is included in this background section due to its significance as a biomedical ontology.

2.2.3.4 Structured Terminologies in Non-Biomedical Domains

Structured terminologies or ontologies can also be found in non-biomedical domains, such as legislation or marketing. The European Vocabulary (EuroVoc)[4] is a multilingual thesaurus of legal terminology used by various organisations within the European Union, including the European Parliament and national parliaments in Europe. Recent developments involving models previously developed for ICD coding have been presented on data labelled with EuroVoc and a hierarchical terminology of products from Amazon (Chalkidis et al., 2019b,a, 2020). These label spaces are not used within the thesis, but are referred to as examples of similar label spaces some of our methods can be transferred to.

2.3 Datasets

A major issue within the medical domain is access to data. As medical data is sensitive, acquiring of a medical dataset is usually preceded by ethics training. Barring

[3] https://www.orpha.net/consor/cgi-bin/index.php?lng=EN

[4] https://op.europa.eu/en/web/eu-vocabularies

datasets associated with various shared task challenges (*e.g.*, ones provided by n2c2[5]), and the relatively recently released MIMIC-III (Johnson et al., 2016) and MIMIC-IV (Johnson et al., 2023) datasets, labelled medical text is largely unavailable to academic researchers outside the healthcare system or local trusted research environments (such as NHS Safe Havens[6] in the UK). A major contributor to this scarcity of freely available data is the necessity for extensive anonymisation of personal identifiable data – an often laborious process (Dorr et al., 2006) – in order to preserve the patients' privacy. This leads to researchers developing their methods on datasets with restricted access (*e.g.*, coming from hospitals associated with the authors' research institution), leading to issues in reproducibility of their work (such as in the case of Zhang et al. (2020b) or Rios and Kavuluru (2018b)). These restrictions to data access result in majority of the research in certain tasks focusing solely on one dataset (*e.g.*, MIMIC-III) without evaluating on an alternative dataset from a different data source (institution). This lack of variety also means transferrability across languages is questionable – even between English-speaking countries. Apart from differences in the variants of the English language, the medical vocabulary may also differ, for example, in drug names – *e.g.*, the drug commonly known as Paracetamol in the UK is primarily referred to as Acetaminophen in the United States. Furthermore, organisational differences between national healthcare systems affect the data. A notable example of this is the purpose of coding – while in the UK coding is primarily intended for internal use, ICD codes within the US are also used for billing purposes.

2.3.1 MIMIC-III

Medical Information Mart for Intensive Care III (MIMIC-III) (Johnson et al., 2016) is a multimodal dataset relating to patients admitted to the intensive care units of the Beth Israel Deaconess hospital in Boston MA. This includes data for adult patients from 2001 to 2012, and neonates from 2001 and 2008. The dataset includes a variety of data, including waveforms, imaging reports, nursing notes, discharge summaries, and the diagnostic and procedural codes assigned to the patients. For the purposes of this thesis the focus will be on the free-text discharge summaries and their respective codes coming from the ICD-9-CM and ICD-9-PCS standards. Patients are assigned ICD-9 codes in order of importance for their admission. One patient can have multiple codes assigned to them. These codes do not perfectly correspond to the content of the

[5] *National NLP Clinical Challenges* https://n2c2.dbmi.hms.harvard.edu/

[6] https://www.nhsresearchscotland.org.uk/research-in-scotland/data/safe-havens

discharge summaries. Searle et al. (2020) argue that MIMIC-III is significantly under-coded for specific conditions, and sometimes incorrect codes are assigned, *e.g.*, in the case of the patient's smoking status (Falis et al., 2019).

MIMIC-III contains data from 53,423 unique hospital admissions for adult patients (aged 16 years or higher). The discharge summary part of the dataset consists of 59,652 discharge summaries relating to 52,726 hospital admission. Personal data of the patients, but also locations, hospital personnel, and dates are anonymised, while some useful information is retained – *e.g.*, the delta used for anonymisiation of time is the same for a single patient across their hospital admissions, hence while the absolute time frame of their hospital visit is de-identified, the time delta within each visit and among the patient's visits as a whole is preserved and hence it can be calculated how much time had passed between hospital admissions.

The dataset's availability is conditioned upon signing a data sharing agreement which implements further security measures including prohibiting the person granted access from attempting to re-identify the data, share the data with persons who have not been granted access, or processing the data via an online API. As showing an example from the dataset would be in conflict with the data sharing agreement, an example discharge summary was produced based on a freely available discharge summary from the *Transcribed Medical Transcription Sample Reports and Examples*[7] repository. Furthermore, the document was rewritten to loosely follow the style and format of MIMIC-III, and a sample coding was produced for it (the coding was not performed by a trained coder and is meant for illustrative purposes only). This example discharge summary is presented in Figure 2.3.

2.3.1.1 Mullenbach *et al.*'s Pre-processing

Given the significance of the contribution of Mullenbach et al. (2018) with their CAML model trained on MIMIC-III, and, crucially, their data processing and training scripts being available on GitHub[8], their pre-processing and dataset split have often been re-used by subsequent work. The pre-processing includes lowercasing, and removal of purely numeric strings and punctuation. Furthermore, discharge summaries relating to the same hospital admission are concatenated, resulting in a single document per hospital admission. Finally, the length of processed input text into the model is lim-

[7] https://www.mtsamples.com/site/pages/sample.asp?Type=98-General%20Medicine&Sample=2617-Cellulitis%20-%20Discharge%20Summary

[8] https://github.com/jamesmullenbach/caml-mimic

NAME: [Known patient lastname**], [**Known patient firstname**]**

[Unit Number 621**]**

Admission Date: [2345-08-15**]**

Discharge Date: [2345-08-30**]**

Date of Birth: [2292-07-19**]**

Sex: M

SOCIAL HISTORY:

Patient is a smoker.

HISTORY OF PRESENT ILLNESS:

The patient is a 52-year-old male who has had a very complex course secondary to a right lower extremity complex open wound. He has had prolonged hospitalizations because of this problem. He was recently discharged when he was noted to develop as an outpatient swollen, red tender leg. Examination in the emergency room revealed significant concern for significant cellulitis. Decision was made to admit him to the hospital.

HOSPITAL COURSE:

The patient was admitted on [**2345-08-15**] and was started on IV antibiotics elevation, was also counseled to minimizing the cigarette smoking. The patient had edema of his bilateral lower extremities. The hospital consult was also obtained to address edema issue question was related to his liver hepatitis C. Hospital consult was obtained. This included an ultrasound of his abdomen, which showed just mild cirrhosis. His leg swelling was thought to be secondary to chronic venostasis and with likely some contribution from his liver as well. The patient eventually grew MRSA in a moderate amount. He was treated with IV vancomycin. Local wound care and elevation. The patient had slow progress. He was started on compression, and by [**2345-08-23**] his leg got much improved, minimal redness and swelling was down with compression. The patient was thought safe to discharge home.

DISCHARGE DIAGNOSIS: Complex open wound right lower extremity complicated by a methicillin-resistant staphylococcus aureus cellulitis.

ADDITIONAL DISCHARGE DIAGNOSES:

1. Chronic pain.
2. Tobacco use.
3. History of hepatitis C.

DISCHARGE INSTRUCTIONS: The patient was discharged on doxycycline 100 mg p.o. b.i.d. x10 days. He was also given prescription for Percocet and OxyContin, picked up at my office. He is instructed to do daily wound care and also wrap his leg with an Ace wrap. Followup was arranged in a couple of weeks.

DISCHARGE CONDITION: Stable

Figure 2.3: An example discharge summary . Sample labelling: *891.1: Open wound of knee, leg [except thigh], and ankle, complicated*; *682.6: Cellulitis and abscess of leg, except foot*; *782.3: Edema*; *338.4: Chronic pain syndrome*; *305.1: Tobacco use and dependence*; *V12.09: Personal history of other infectious and parasitic diseases*.

ited to 4,000 tokens. For inputs exceeding the limit, the first 4,000 tokens are used. Mullenbach's dataset split was used in ICD-9 coding experiments in Chapters 3 and 4.

2.3.2 MIMIC-IV

The ICD-coded discharge summaries of the fourth Medical Information Mart for Intensive Care (Johnson et al., 2023) were released in early 2023. The dataset contains documents coded either with ICD-9-derived ontologies (ICD-9-CM, and ICD-9-PCS), or with the more recent ICD-10 standards (ICD-10-CM, ICD-10-PCS). The data was collected from the same institution as MIMIC-III over the time period between 2008 and 2019 and underwent similar procedures of anonymisation. MIMIC-IV is larger than MIMIC-III, with 209,326 discharge summaries coded with ICD-9, and 122,279 coded with ICD-10. Similar to MIMIC-III, access to MIMIC-IV is conditioned upon signing a data-sharing agreement. A comparison between the sizes of the MIMIC-III and MIMIC-IV datasets and the respective label spaces can be seen in Table 2.2.

2.3.2.1 Edin *et al.*'s Pre-processing

Edin et al. (2023) criticise the widely-utilised dataset split and pre-processing of Mullenbach et al. (2018) in tandem with macro-averaged evaluation. The authors adjusted the split proposed in Mullenbach et al. (2018) such that the validation and test sets do not include labels absent from the training set, and produced similar splits for the ICD-9 and ICD-10 subsets of MIMIC-IV. While the macro-averaged evaluation concern is valid, this adjustment results in the removal of many labels, and, more crucially, removes the rare labels from the task. As the thesis focuses on data sparsity, the complete removal of these labels makes the split unsuitable for the research presented in it. Hence, this split was not used. Macro-averaged evaluation will be further explained in Chapter 3.

2.3.2.2 Nguyen *et al.*'s Pre-processing

Nguyen et al. (2023) propose a dataset split for the ICD-10 subset of MIMIC-IV more akin to the MIMIC-III split proposed by Mullenbach et al. (2018). No labels were removed as part of the processing. This is the dataset split used for experiments in Chapter 5.

2.3.3 Label Distribution

A general issue in datasets involving ontologies is the label distribution. Firstly, the label space tends to be quite large, although some studies reduce the problem to a number of most frequent labels. The label distribution tends to be skewed, with the label frequency distribution having a big head (showed for MIMIC-III and MIMIC-IV in Table 2.3) and a long tail (shown in Table 2.4). Within MIMIC-III roughly one fifth of the labels appears only once. Almost half of the labels do not appear more than 5 times within the dataset. Codes appearing up to 50 times constitute 81.43% of the codeset, while only representing 7.3% of the total number of labels (the long tail). The ten most populous labels, on the other hand, represent only 0.11% of the label space while covering 13.85% of the total number of labels (the big head). Figure 2.4 visualises the scale of the big-head long-tail issue within MIMIC-III. The phenomenon is even more pronounced within MIMIC-IV with higher proportions of the codeset being covered by less populous labels, while the top 10 most populous labels constitute 31.53% of the total number of labels.

Infrequent codes lead to the *few-shot* (FS) scenario in machine learning – training models using limited amounts of training samples. Dataset splitting with infrequent labels also may lead to the *zero-shot* (ZS) scenario – instances where a model encounters previously unseen labels (present in a validation or test set while absent from training data). What is more, a portion of the label space may not be present in the dataset at all. Mullenbach et al. (2018) note that there are 8,921 unique codes coming from ICD-9-CM and ICD-9-PCS in MIMIC-III, whereas the entirety of the combined codeset exceeds 13,000. Addressing the concept sparsity embodied by the few-shot and zero-shot scenarios is the core focus of the thesis.

2.3.4 Other Dataset Issues

There is relatively few large expertly-labelled datasets within the space of medical document coding. Furthermore, of the existing datasets many are either available with restricted access (*e.g.*, MIMIC-III and MIMIC-IV), or unavailable to the general public – some studies use their local hospital data which they are unable to share (*e.g.*, (Rios and Kavuluru, 2019)). There is also an issue in labelling standards – MIMIC-III (a popular dataset before the release of MIMIC-IV in 2023) may have been significantly undercoded (Searle et al., 2020). Furthermore, most of the datasets available are in English (notable exceptions are datasets in Spanish (Miranda-Escalada et al., 2020)

	MIMIC-III (ICD-9)	MIMIC-IV (ICD-10)
Number of Discharge Summaries	59,652	122,310
Number of Hospital Admissions	52,726	122,310
Number of Subjects (Patients)	41,127	65,683
Number of All ICD Codes	848,692	1,974,338
Number of Unique ICD Codes	8,930	26,096
Median Number of Codes Per Document	14	15
Mean Number of Codes Per Document	16.1	16.4

Table 2.2: A comparison of coded discharge summaries within MIMIC-III and the ICD-10-coded part of MIMIC-IV. The size of the ICD-10-coded subset of MIMIC-IV is more than double the size of MIMIC-III, containing more than twice the number of codes. The size of MIMIC-IV's ICD-10 codeset is almost three times that of MIMIC-III's ICD-9 codeset. Note that on the document level code statistics are fairly similar between the two datasets – MIMIC-III's number of codes per document median is 14, mean 16.1, with a standard deviation of 8.47, while MIMIC-IV's statistics are 15, 16.14, and 9.05 respectively.

	ICD-9 Code	Pop in MIMIC-III	ICD-10 Code	Pop in MIMIC-IV
1	401.9	20,053	E78.5	44,043
2	38.93	14,444	I10	43,573
3	428.0	12,842	Z87.891	36,296
4	427.31	12,594	K21.9	30,802
5	414.01	12,179	F32.9	23,232
6	96.04	9,932	I25.10	22,606
7	96.6	9,161	N17.9	19,706
8	584.9	8,907	F41.9	19,156
9	250.00	8,784	Z79.01	15,319
10	96.71	8,619	Z79.4	15,276

Table 2.3: A comparison of the ten most frequent codes within MIMIC-III and the ICD-10-coded subset of MIMIC-IV. "Pop" stands for population. The differences in scale between the datasets stem from the size difference reported in Table 2.2. There is some commonality between the codes – *e.g.*, ICD9's *401.9: Essential hypertension, unspecified* corresponds to ICD-10's *I10: Essential (Primary) Hypertension*.

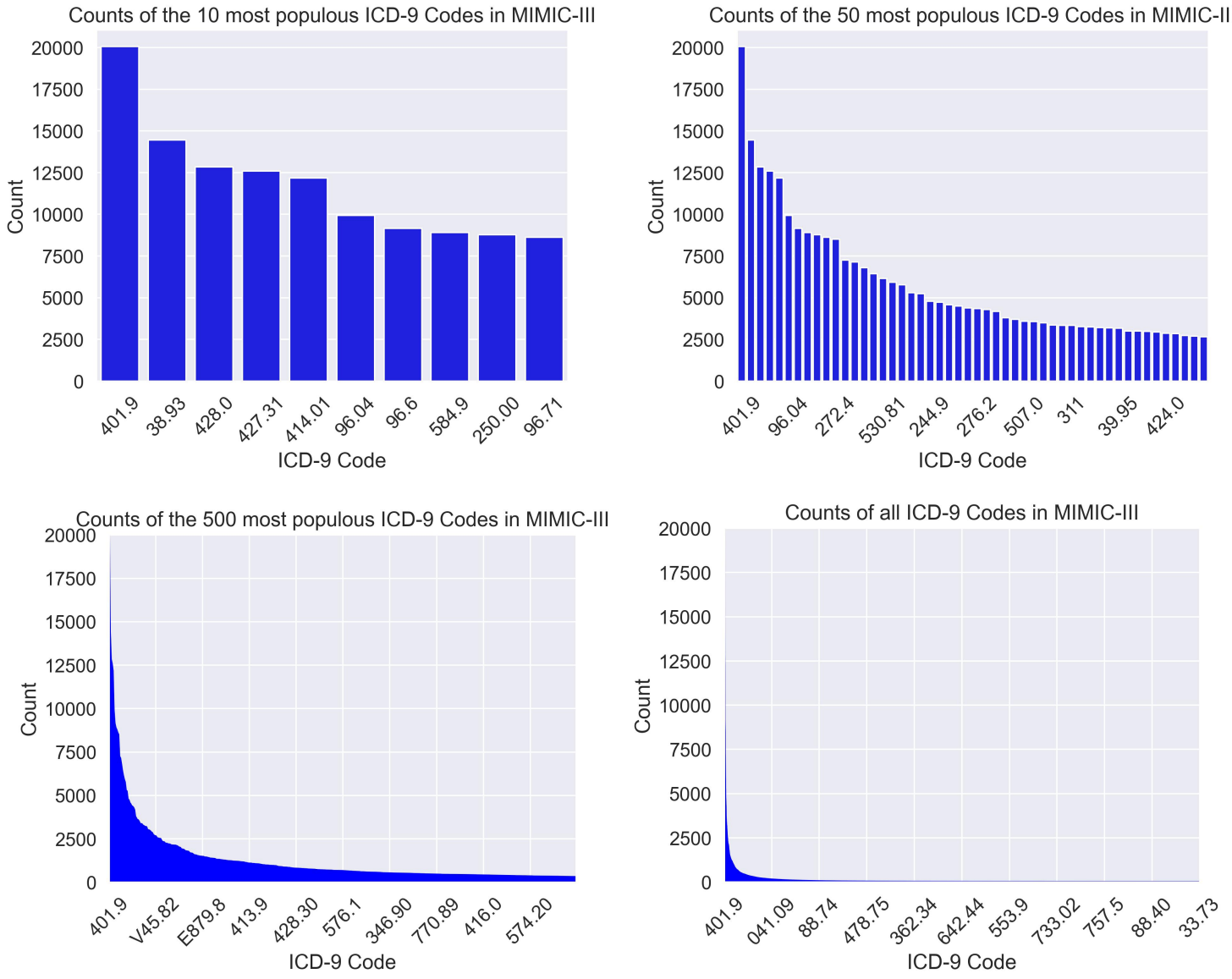

Figure 2.4: The Big-Head Long-Tail distribution of labels within MIMIC-III for the 10, 100, and 500 most populous labels, and the full label set.

	MIMIC-III (ICD-9)				MIMIC-IV (ICD-10)			
Pop	Unique	%	Total	%	Unique	%	Total	%
1	1,914	21.43	1,914	0.23	8,046	30.83	8,046	0.41
≤5	4,351	48.72	9,387	1.11	15,686	60.11	30,840	1.56
≤10	5,402	60.49	17,522	2.06	18,483	70.83	52,221	2.64
≤50	7,272	81.43	61,991	7.30	22,954	87.96	155,879	7.90
top 10	10	0.11	117,515	13.85	10	0.04	622,430	31.53

Table 2.4: The Big-Head Long-Tail distribution of labels within the MIMIC-III and MIMIC-IV (ICD-10 subset) datasets of discharge summaries. "Pop" stands for population.

and Chinese (Dong et al., 2022a)), similar to the general state of matters within the field of NLP. Even within the scope of English, variety arises not only due to local linguistic differences, but also the length and style of discharge summaries (Dong et al., 2022a) or coding guidelines of different healthcare systems (*e.g.*, the added factor of billing purposes within the US).

2.4 Named Entity Recognition and Linking

Named Entity Recognition and Linking (NER+L) is the combination of two tasks – identifying words or phrases of interest, also referred to as named entities – (NER); and classifying them by linking them to concepts in a standardised vocabulary/ontology (L). The task is performed on the word or phrase level, with the output being a list of mentions indicated by indices and the associated classes. For instance, in the sentence "The patient suffers from T1D." the goal would be to retrieve the mention of "T1D" on the index range of 25:28 (assuming 0-indexing) and link it to the concept *250.01: Diabetes mellitus without mention of complication, type I [juvenile type], not stated as uncontrolled* in ICD-9 (or a different controlled vocabulary, such as the UMLS). There can be multiple mentions of the same concept presented in a variety of surface forms (*e.g.*, "Type 1 Diabetes", "Type 1 Diabetic", or "T1D") all of which should be retrieved (their uniqueness presented via their respective index ranges) and associated with the concept within the vocabulary.

The thesis does not contribute directly to the NER+L task. It, however, utilises pre-existing NER+L methods to identify mentions of concepts within training document relating to the expert-assigned codes as part of data augmentation and synthesis

methods in Chapter 4.

2.4.1 Rule-Based Methods

Medical NLP has a strong tradition in rule-based approaches. Algorithms based on rules devised in collaboration with medical professionals having been developed for tasks involving Named Entity Recognition of medical concepts, text segmentation, or negation detection– *e.g.*, the original MetaMap (Aronson, 2001), Edie-R (Grivas et al., 2020), NegEx (Chapman et al., 2001). Approaches that detect medical concepts usually incorporate a medical ontology, most notably the UMLS. The most successful system combining hand-crafted rules and classical machine learning techniques is the clinical Text Analysis and Knowledge Extraction System (Apache cTAKES) (Savova et al., 2010). Apache cTAKES builds on pre-existing open-source technologies incorporating a sentence boundary detector, tokeniser, normaliser, Part-of-Speech tagger, a shallow parser and a Named Entity Recognition annotation, including a status and negation annotator. The system uses a dictionary that is a subset of the UMLS including SNOMED CT and other ontologies. The system itself is designed for a different task (information extraction), but it has been considered for the ICD coding task. There has been at least one attempt of using the output of cTAKES during the training of a neural ICD coding model (Wiegreffe et al., 2019), although this has not led to improved performance.

2.4.2 Neural and Hybrid Methods

2.4.2.1 SemEHR

SemEHR (Wu et al., 2018) is a general-purpose biomedical information extraction system for Electronic Health Records (EHRs). It builds upon previous work within the CogStack project at King's College London to build a system identifying contextualised mentions (considering negation, temporality, and the experiencer) of a vast array of biomedical concepts coming from ontologies and standardised vocabularies, such as SNOMED CT or ICD-10. The mentions are also associated with semantic type annotations and their clinical contexts based on the documents or sections they were found in (mostly via rule-based systems).

The backbone of the pipeline of the extraction system is Bio-YODIE – an NLP pipeline for identifying mentions of clinical concepts within clinical notes and associ-

ating them with standardised concepts within the UMLS. Bio-YODIE forms a common static system, which can be further adapted through a continuous learning subsystem. This comprises a rule-based subcomponent based on the user's rejection of unwanted results; and a machine learning engine which considers the user's feedback (acceptance or rejection of identified mentions within an input corpus) and uses it as further training data. Furthermore, SemEHR includes a consuming subsystem supporting downstream tasks (*e.g.*, patient characterisation or trial recruitment) using the predicted mentions with the possibility of inclusion of external knowledge resources (such as ontologies).

SemEHR was utilised in the thesis as one of the NER+L systems involved in rule-based data augmentation and synthesis in Chapter 4.

2.4.2.2 MedCAT

MedCAT (Kraljevic et al., 2019) is a more recent extraction system also working within CogStack. A major improvement over SemEHR comes from context-sensitive representations – different instances of the same surface form, if they appear sufficiently often within different contexts, can be recognised as referring to different concepts. One of the chief use cases are abbreviations (*e.g.*, "HR" can mean "hour" or "heart rate"), which are common within medical records in the interest of brevity and differ in meaning depending on the subdomain (*e.g.*, "LFT" can stand for "liver function test" or "lung function test"). Similar to SemEHR, MedCAT can link detected concepts to the UMLS. The reliance in this system is more on neural models in the NER+L, relying on training the model on the user's own data. However, a ready-to-use pre-trained model along with its associated concept database and vocabulary based on MIMIC-III can be requested from the authors.

The system contains a concept database (CDB), which can be seen as a controlled vocabulary of relevant entities. It can be initialised with various ontologies. A core assumption within MedCAT is that each relevant concept, though it may be presenting in a variety of ways, will have one surface form that is unique (not requiring disambiguation). Once the concept is identified and its embedding is learned it can be aligned with other surface forms that appear close in the embedding space and/or in similar contexts (by means of co-occurrence matrices).

MedCAT was utilised in the thesis as one of the NER+L systems involved in rule-based Data Augmentation and Synthesis in Chapter 4.

2.5 Large-Scale Multi-Label Text Classification

The main focus of this thesis is on the task of document classification with structured label spaces coming from medical ontologies – such as automated ICD coding in MIMIC-III. The family of such document classification tasks can be found within the literature under different names. While Mullenbach et al. (2018) refer to it merely as Multi-label classification, Rios and Kavuluru (2018b), and Chalkidis et al. (2019b) opt for a more specific name – *Large-Scale Multi-Label Text Classification* (LMTC). This task name was chosen to differentiate from the task of Extreme Multi-label Text Classification (XMTC) whose size of the label space is in the order of millions -– such as the label set of categories in Wikipedia. An LMTC task is a document classification task with a label space typically in the order of thousands of classes. This task is applicable to various domains, *e.g.*, classification of diseases and procedures in medical text, legal concepts in legislation, and categorisation of products.

Similar to NER+L, LMTC utilises labels from a standardised vocabulary/ontology/label space to unstructured/semi-structured text. While individual predictions and gold standard labels on the word-level are unique and associated with a range within the text (*strongly-labelled scenario*) in the case of NER+L, in LMTC the comparison between the predictions and gold standard are on the document level as sets (*weakly-labelled scenario*).

The large label spaces of LMTC tasks often display skewed label distributions with many labels absenting from the training set, or the dataset as a whole. Furthermore, the label space of an LMTC task is organised in a label hierarchy (*e.g.*, a medical ontology). These hierarchies may possess different labelling guidelines depending on the domain, *e.g.*, labelling only the leaf node, or labelling the leaf node and its ancestors (similar to the true-path rule). Due to the organised nature of the label space, there may be explicit relations between labels, *e.g.*, incompatibility of a pair of labels. These label sets may receive regular updates that include reassigning a label's position within the hierarchy, or even the introduction of new labels. Without an update of the data, the introduction of a new label constitutes the *zero-shot* scenario.

In the following subsections we shall briefly introduce how evaluation is commonly performed in LMTC and describe relevant LMTC models which were either directly used or were frequently referred to within the thesis.

2.5.1 Evaluation

The performance of LMTC models is commonly measured using *precision*, *recall*, and F_1-score. For a given input document the set of predicted labels is compared against the set of gold standard labels. Labels common between the two sets are considered true positives. Labels appearing only in the prediction set are false positives, while ones appearing only in the gold standard are false negatives. These metrics can be produced for each label within the label-space. However, the size of the label-space makes per-label reporting inconvenient and the most common results are micro- and macro-averaged F_1-scores. The common approaches to evaluation in LMTC treat the label-space as a set of individual independent labels ignoring the rich information present within the ontology, notably the ontological structure. A more detailed background on evaluation including the mathematical expressions for individual metrics is presented in Chapter 3, which focuses on evaluation within LMTC and describes our contributions to evaluation approaches involving the ontological structure.

2.5.2 Convolutional Attention for Multi-Label Classification

Probably the most notable early attempt at the LMTC task of ICD-9 coding in MIMIC-III is the Convolutional Attention for Multi-Label classification (CAML) (Mullenbach et al., 2018). The architecture consists of a convolutional neural network (CNN) (O'Shea and Nash, 2015) encoder, which encodes the input text on a phrase-level, a label-specific attention mechanism over phrase encodings, a combination of the attention with phrase encodings resulting in a document representation and, finally, a binary classification. The label-specific attention mechanism produces an attention distribution over all the encoded phrases with respect to each individual label in the label space. Hence, for a label space L of size $|L|$, $|L|$ individual representations are created representing the relevant parts of the text for each respective label. Each representation is evaluated by a binary classifier corresponding to its label. Therefore, the output layer of the model are $|L|$ binary predictions. While coming from the same input text, the binary predictions are independent of each other. Furthermore, the structure of the ontology is not utilised, with the output layer being flat.

Mullenbach's approach has since been improved upon. Sadoughi et al. (2018) addressed the CNN encoder's uniform filter size by introducing a "multi-view" CNN – the input text is read through several convolutional filters of different sizes and max-pooled prior to the per-label attention mechanism. Falis et al. (2019) make the true-

path assumption and leverage the code hierarchy performing multi-task learning on the different levels of the label space. While this approach leads to improved performance through leveraging the attention information from higher levels, the output layers of their model are flat (just as in Mullenbach et al. (2018)). A diagram of the CAML model is presented in Figure 2.5.

Mullenbach et al. (2018) also use the computed attention distribution for each class for the purpose of interpreting the predictions. Their approach is to retrieve windows of text centered on tokens with high attention scores for a given class. Through the inspection of these windows of text, the reader can evaluate the model's prediction qualitatively.

While some of the methods introduced by Mullenbach *et al.* are present in newer models – most notably the pipeline of encoder, per-label attention, and output -– the approach has several fundamental shortcomings. Firstly, it is not suited for addressing data sparsity. In particular, its language model includes a word2vec model (Mikolov et al., 2013a) trained on word-level units with a limited vocabulary (every word appearing at least 5 times in MIMIC-III). All *Out of Vocabulary* (OOV) tokens are mapped to a special unknown "**UNK**" token. Rare diseases or procedures may be associated with rare tokens, *e.g.*, surnames – Aarskog-Scott syndrome. Hence poor handling of rare, or previously unseen tokens may affect performance for rare classes.

Secondly, as part of the pre-processing, all tokens that do not include alphabetical characters are removed. This includes, but is not limited to, numbers. Some ICD-9 codes directly reference numbers within their description *e.g.*, *96.71: Continuous Invasive Mechanical Ventilation For Less Than 96 Consecutive Hours*. Others may not refer to numbers within the description, but depend on them *e.g.*, assignment of the descendant codes of the *401: Essential hypertension* head (malignant, benign, unspecified) within MIMIC-III does not necessarily rely on the explicit statement of "benign hypertension" or "malignant hypertension", with the direct reference to hypertension potentially missing from the text. The decision may be affected by the patient's vitals, in particular their systolic and diastolic pressure, whose values are removed during Mullenbach *et al.*'s pre-processing. While numeric data is not our primary focus, it is important to keep its relevance in mind as it might affect or even negate our efforts for some labels.

Of the methods within CAML, probably the most contributing to its success was its per-label attention mechanism. After producing a common encoding of the input document – in the case of CAML by first translating words into static word em-

beddings (Mikolov et al., 2013a) and then combining them through convolutions – a label-specific representation of these embeddings is derived for each of the labels within the considered label space. An attention mechanism in the form of a softmax function is applied over each of the label-specific representations, resulting in a distribution of attention over the convolved embeddings. The result of the label-specific attention is multiplied with the common encoding resulting in label-specific document representation, which is then used within the binary prediction of the respective label in the decoder. The label-specific attention corresponds to the importance of each word/phrase within the input for the respective label's prediction. This technique has since been an integral part of further research, be it in its original or improved form (Sadoughi et al., 2018; Falis et al., 2019; Chalkidis et al., 2020; Dong et al., 2021c). The original publication claims that these attention weights (possible to visualise as a heatmap over the text) can be used as a form of explainability of the model. This explainability claim has since been contested (Wiegreffe and Pinter, 2019).

2.5.3 Hierarchical Label-Wise Attention Network

Hierarchical Label-wise Attention Network (HLAN) (Dong et al., 2021a) introduced further refinement of the attention mechanism. Common word-level representations are created for each sentence within the input by combining the word embeddings within the sentence via a Bi-directional Gated Recurrent Unit (Bi-GRU) module. Hence, rather than creating a label-specific document embedding from the common representation as in Mullenbach et al. (2018), label-specific sentence embeddings are first created. A further Bi-GRU + attention module is then applied to the sentence embeddings in order to build the final document-level representation. Furthermore, HLAN initialises label embeddings in order to represent the correlation between the different labels.

2.5.4 The Label Attention Model

The Label Attention Model (LAAT) (Vu et al., 2020a) compared to CAML introduced an updated label-wise attention module. The common representation goes through one more hidden layer with a *tanh* activation before the creation of label-specific representations. Vu et al. (2020a) also include a hierarchical joint model JointLAAT, which uses two attention mechanisms to compute attention on two levels of the hierarchy – the head code, and the full leaf-level code – with the prediction of the higher level being

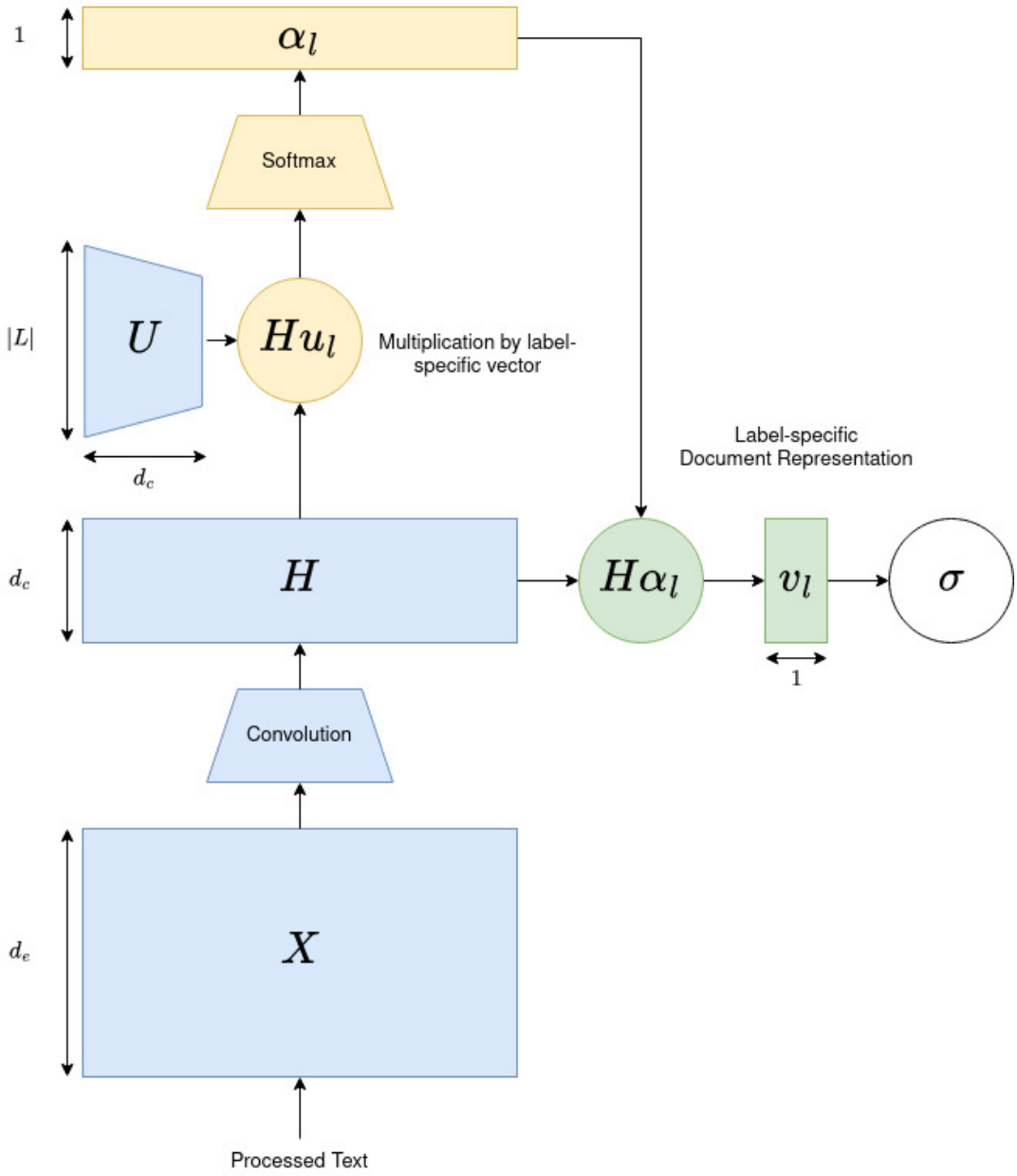

Figure 2.5: The CAML architecture from Mullenbach et al. (2018)). Blue components indicate layers common for all labels, yellow components are label-specific ($|L|$ such layers exist in parallel where $|L|$ is the number of layers), green components come from the combination of yellow (label specific) and blue (common) components (hence there is also $|L|$ parallel green components). Component X represents word embeddings, H convolved word embeddings, H_l convolved word embeddings multiplied by a label specific vector (u_l). Matrix U is a trainable component containing the individual vectors u_l for each of the labels in the label space L. Vector v_l is the dot product between the common convolution component H and the label-specific α_l. As these two components share a dimension that is dependent on the size of the input text, the size of the resulting vector will be independent of the size of the input text. To produce a version of this network for all labels L, the label-specific u_l vectors are combined into a matrix U.

propagated into the final leaf-level prediction (an idea similar to Falis et al. (2019)).

2.5.5 Multi-Filter Residual Convolutional Neural Network

The Multi-Filter Residual Convolutional Neural Network (MultiResCNN or MRCNN) (Li and Yu, 2020a) utilises a multi-filter convolutional layer to capture text patterns with different lengths (similar to ones proposed by (Sadoughi et al., 2018)) and a residual convolutional layer to enlarge the receptive field.

2.5.6 Pretrained Language Model-Based ICD-Coding

Huang et al. (2022a) present a framework for ICD coding using pre-trained language models (PLM-ICD), such as the *Bidirectional Encoder Representations from Transformers* (BERT) model (Devlin et al., 2018), as the encoder subarchitecture. To better match the language within medical records, domain-specific pre-trianed language models were suggested as the most suitable (*e.g.*, BioBERT (Lee et al., 2020), PubMedBERT (Gu et al., 2021), and RoBERTa-PM (Lewis et al., 2020a)). Unlike in the case of CNN- and RNN-based encoders in tandem with word embeddings, pre-trained transformers have a limit on the size of the input document. To manage this issue the authors propose segment pooling, which first splits the input document into consecutive segments short enough to satisfy the input length constraint and encode the segments into segment representations through the pre-trained language model. The hidden representation of the document is a concatenation of the segment representations.

2.5.7 Graph Convolutional Neural Network Methods

Rios and Kavuluru (2018b) noted the skewness of the label distribution and investigated the performance of CAML on classes within the few-shot and zero-shot scenario. To address the hierarchical structure of the label space, they modify CAML's output layer to be a 2-layer graph convolutional neural network considering the vector representation of the parents and children of each node. These vector representations come from the embeddings of the textual description of each node within the ontology in order to provide background information for the few-shot and zero-shot scenarios. For each description, the words are converted into individual word embeddings and averaged to produce a single embedding. It should be noted that this representation is

similar to Bag of Words in that the information about the word order is lost (*e.g.*, "The dog chased the cat." results in the same embedding as "The cat chased the dog."). This issue could be addressed by changing the encoder to either a recurrent architecture or an attention-based model, such as BERT (Devlin et al., 2018).

Despite the fact that code descriptions are relatively short (Rios and Kavuluru (2018b) report an average description length of 7 words) and tend to differ very little among codes who share a parent (sibling codes), this approach resulted in an improvement in the few-shot and zero-shot scenario. While the GCNN layers were also reported to help in the few-shot and zero-shot scenario, Chalkidis et al. (2020) contested their conclusions, showing in their experiments that their models with GCNN components benefited mostly from additional model capacity, rather than hierarchy awareness and hence opted for exploiting the hierarchy through the use of graph encoding Node2Vec (Grover and Leskovec, 2016). The conclusion they draw is that the label encodings coming from the textual descriptions are far more important than the hierarchical information. While both of the discussed publications provide results using GCNNs, they do not seem to specify how likely the models are to mispredict descendant codes for parent codes – be it through prediction of a single incorrect descendant, or overprediction of multiple descendants.

2.6 Summary

This chapter introduced the task of automatic medical document coding. First, the medical coding task was described as performed by specially-trained human coders. Issues within the task were highlighted, such as heavy use of human resources, and being error-prone in order to motivate the need for (at least partial) automation. Clinical ontologies were presented with a particular focus on the International Classification of Diseases (ICD) – the family of ontologies used as the label-space of the medical document coding datasets this thesis utilises. The MIMIC-III and MIMIC-IV datasets were introduced along with pre-processing pipelines used alongside them within automated ICD coding studies. Comments were made on the common issues within ICD coding datasets, particularly data sparsity. *Named Entity Recognition and Linking* (NER+L) and *Large-Scale Multi-Label Text Classification* (LMTC) tasks were described and contrasted. As ICD coding has been commonly cast as an LMTC task, further detail was provided on some of the existing LMTC models applied to ICD coding. The common evaluation approaches used in prior work were touched upon and will receive

more attention in Chapter 3, where the thesis contributes evaluation metrics involving ontological structure. Background on data augmentation and Large Language Models was not included in this Chapter. Instead it is discussed in Chapters 4 and 5 alongside the thesis' contributions to rule-based data augmentation and synthetic data generation within the space of ICD coding respectively.

Chapter 3

Evaluation Metrics

3.1 Introduction

Evaluation within the space of *Large-Scale Multi-Label Text Classification* (LMTC) approaches to ICD coding is usually presented using micro- and macro-averaged metrics treating individual predicted labels as independent from one another. While convenient for an overall presentation of model performance, this approach does not capture the finer detail useful for error analysis and understanding of the flaws of the evaluated model. Crucially, labels are not fully independent of each other within the hierarchical label space. This chapter introduces the existing common approaches to evaluation within LMTC, argues for a more prominent involvement of the hierarchical structure of the label space within the evaluation and presents two methods developed for the thesis which utilise the structure – Count-Preserving Hierarchical Evaluation (published in Falis et al. (2021)); and Weak Hierarchical Confusion Matrices (published in Falis et al. (2022)).

In the field of machine learning we utilise a range of evaluation metrics to inform us of a model's performance, thus allowing comparison between models given a common evaluation set. The metric of choice for a task should depend on the features of the task itself – what is the desired outcome, and how can we measure our success in attaining it? Are we trying to group the most similar unlabeled datapoints together in clusters to discover subpopulations? Then our metric should favour models grouping similar datapoints together, while keeping different ones apart. Are we concerned with maximising the number of correct predictions – *e.g.*, if trying to identify need for a costly intervention? Then it is suitable to optimise the precision of the model, representing the proportion of predictions that were made correctly. Are we interested in

capturing every single instance, while accepting potential false alarms (*e.g.*, predicting a technical issue in a nuclear power plant)? Then our focus should be on recall, corresponding to the percentage of relevant events that were correctly identified.

Furthermore, our goals may not be clear-cut. We may be interested in more than one metric, or even a combination of them. For instance, high precision and recall are both desirable (though different settings may give priority to one) but can individually be maximised while ignoring the other. A model making a single prediction which happens to be correct will report perfect precision, but its recall may be low if prediction-worthy events are frequent. On the other hand, a model making every possible prediction for every datapoint will achieve perfect recall, but will likely have low precision due to predictions on irrelevant datapoints. The F_1-score is the harmonic mean between the precision and recall of a model on a set of datapoints assigning both metrics equal weight.

Why do we need evaluation? In its most basic form, it allows us to compare models and methods to determine which one is more suitable for a given task based on performance on a set of data. We recognise training, validation, and generalisation error. During the procedure of training, the model is presented with training data (commonly referred to as *training set*) and performs a training task on them – *e.g.*, classification, clustering, or regression. The prediction is then evaluated on the gold standard associated with the data for supervised learning tasks (classification, or regression); or on the cohesion of the predicted structure in unsupervised learning (*e.g.*, clustering). The errors are quantified through a loss (or error) function, based on which training steps are performed adjusting the trainable parameters of the model. The aim of this procedure is the lowering of training error in the next training iteration through modelling the training data more accurately. Working only with the training data, however, risks the model learning to fit the training data perfectly (including the irrelevant details commonly referred to as *noise*), while not leaving enough leeway for data unseen during training – the *overfitting* problem (Jurafsky, 2000). To account for this, training procedures utilise a *validation set* – a set of datapoints the model is not directly trained on, but is evaluated on during training so as to compare the trends in training error and validation error. If a model's performance improves on the training set while deteriorating on the validation set, the model is considered to be overfitting to the training data, and hence less likely to generalise well and more likely to underperform on previously unseen data. Validation performance is often used for selecting the best iteration of a model during a training procedure, which is subsequently tested for its generalisation.

Generalisation error (Bishop, 1995) represents the model's performance on all previously unseen data. As it is impossible to evaluate the model on all of the existing (current or future) data, the true generalisation error cannot be calculated. Instead, the generalisation error is approximated through evaluating a model on a held-out *test set*. Similar to the validation set, the test set datapoints are not used for adjusting the trainable parameters of the model during training. Unlike the validation set, the test set is not intended as an iterative check for a model to select the parameter setting that generalises the best to a held-out set, but rather be used only once as a final evaluation of the model with the parameters selected through the training-validation procedure as an approximation of how the model will generalise to new data. This approximated generalisation error is then used to compare the performance of different approaches to a given task and determine the best performing approach – the *State of the Art* (SotA).

Lack of generalisation appearing in evaluation on data unseen in training may relate to already seen concepts deviating in the presentation of their own surface form (*e.g.*, black swans to ancient Romans) or in different contexts (*e.g.*, a bright-red double-decker bus during daytime to a model trained only on images taken during nighttime); or concepts that have never before been observed (*e.g.*, a new strain of a coronavirus undetected by older at-home testing kits). In the context of document coding, issues in generalisation may arise from the variety of surface forms (synonyms or abbreviations) in which a concept may appear, (lack of) adaptation to updated coding guidelines, or introduction of new codes.

Medical document coding is a task encompassing a wide range of concepts (in the order of tens of thousands) whose frequency in the world – and hence the data collected from real-world scenarios – varies drastically (as outlined in Chapter 2). Human coders are equipped with an up-to-date coding standard with further reference guidelines[1] that provide a taxonomy of the different concepts with notes on inclusion and exclusion criteria on individual codes (see example in Figure 3.1). Regardless of the frequency of a concept (and by proxy the likelihood that the coder encountered it in a clinical document before) its presence within the coding guidelines makes it valid for a coder to look up and consider for assigning. Human coders assign codes according to the latest adopted standard within their institution. For instance, MIMIC-IV contains data spanning more than a decade (between the years 2008 and 2019) and uses codes that were valid at the point of the coding of each discharge summary (Johnson et al., 2023) including editions of ICD-9 and ICD-10.

[1]Example reference books by the National Health Service: `https://classbrowser.nhs.uk/#`

Organ failure must be coded in addition when documented with sepsis: ***see DCS.IX.10: Heart failure (I50), DCS.X.7: Respiratory failure, not elsewhere classified (J96) and DCS.XVIII.10: Multiple organ failure (R68.8)***.

Septic shock

Whenever septic shock is documented in the medical record by the responsible consultant, code **R57.2 Septic shock** must be assigned in any secondary position following the code that classifies sepsis.

Severe sepsis

The following codes and sequencing must be used for a diagnosis of severe sepsis:

A41.- **Other sepsis** (or the specific type of sepsis recorded in the medical record)
R65.1 **Systemic inflammatory response syndrome of infectious origin with organ failure**
U82.- **Resistance to betalactam antibiotics**, **U83.- Resistance to other antibiotics** or **U84.- Resistance to other antimicrobial drugs** (use only if the severe sepsis is resistant to antibiotics or antimicrobial drugs).

Neutropenic sepsis

The following codes and sequence must be used for a documented diagnosis of neutropenic sepsis:

A41.- **Other sepsis** (or the specific type of sepsis recorded in the medical record)
R65.1 **Systemic inflammatory response syndrome of infectious origin with organ failure** (use only if the sepsis is documented as severe)
U82.- **Resistance to betalactam antibiotics**, **U83.- Resistance to other antibiotics** or **U84.- Resistance to other antimicrobial drugs** (use only if the sepsis is resistant to antibiotics or antimicrobial drugs)
D70.X **Agranulocytosis**

If the responsible consultant has documented that the neutropenic sepsis was due to a drug, then an adverse effects code from Chapter XX must be assigned after **D70.X**, ***see DCS.XX.7: Drugs, medicaments and biological substances causing adverse effects in therapeutic use (Y40-Y59).***

See also:

- ***DGCS.6: Infections***
- ***DCS.I.4: Bacterial, viral and other infectious agents (B95-B98)***
- ***DCS.XVI.5: Group B streptococcus (GBS) bacterial infections in babies***
- ***DChS.XVIII.1: Signs, symptoms and abnormal laboratory findings***
- ***DCS.XIX.7: Postprocedural complications and disorders***

Figure 3.1: An example of coding guidelines taken from National Clinical Coding Standards ICD-10 5th Edition (Version 8.1, 2022 NHS Digital)

The models we wish to build to tackle the task of medical document coding should display these features:

- the concepts predicted by the model match the gold standard (**precision**);
- concepts present in the gold standard are predicted (**recall**);
- the model should be able to predict a concept regardless of how frequent it is in real-world settings (and corresponding data);
- and the model should be robust to changes in concepts seen in the data (surface-form variability) and concepts not seen in the data at all.

While previous work used evaluation metrics that partially captured these features (precision, recall, F_1-score), the rich ontological structure was not considered in the context of their evaluation. Individual predictions of codes per document are treated as independent from each other, while there are non-explicit rules present within the taxonomy. For instance, certain codes should not co-occur in the same document due to being mutually exclusive, *e.g.*, the ICD-9 codes *401.0: Malignant essential hypertension* and *401.1: Benign essential hypertension* should not be assigned to the same patient for the same admission – the patient either presented with one, the other, or neither – but not both.

In the early stages of the project clinical coding staff was consulted in order to better understand the task as part of the review presented in Dong et al. (2022a). The human approach to the task involves tracing a concept through the taxonomy and considering alternative concepts close within the taxonomy as differences on the leaf level may stem from minute details – *e.g.*, there are 40 different leaf-level descendants of *250: Diabetes mellitus* in ICD-9, differentiating on the type of diabetes, whether it is stated as uncontrolled, and whether it presents with different complications (*e.g.*, ketoacidosis). Note that, the cases of ICD-9's *401: Essential hypertension* and *250: Diabetes mellitus* differ in that, while the leaf descendants of the former are mutually exclusive, a patient with diabetes may have multiple complications warranting multiple descendants belonging to subfamilies corresponding to complications (*e.g.*, *250.13: Diabetes with ketoacidosis, type I [juvenile type], uncontrolled* appearing alongside *250.43: Diabetes with renal manifestations, type I [juvenile type], uncontrolled*). As such, it is reasonable to pay special attention to errors in the context of subtrees within the ontology – which the thesis will refer to as *code families*.

This chapter introduces two evaluation methods that incorporate ontological structure. These methods were designed and explored on the task of ICD coding, but can be altered to work with other label spaces coming from tree-structured taxonomies. Section 3.2 provides a brief background on *Large-Scale Multi-Label Text Classification* (LMTC) in comparison and contrast with *Named Entity Recognition and Linking* (NER+L). Section 3.3 first presents the baseline *flat* evaluation (Section 3.3.1) – precision, recall, F_1-score on the micro and macro level; and contrasts it with *hierarchical* methods (Section 3.3.2). This section introduces the novel methods developed as part of the thesis – *Count-Preserving Hierarchical Evaluation* (CoPHE) Count-Preserving Hierarchical Evaluation (Section 3.3.2.2 alongside hierarchical evaluation in prior art in Section 3.3.2.1), and Weak Hierarchical Confusion Matrices (WHCM) (Section 3.3.3.2 alongside the baseline confusion matrix method in Section 3.3.3.1). Section 3.4 presents the results on experiments demonstrating evaluation with CoPHE (Section 3.4.1) and WHCM (Section 3.4.2) on MIMIC-III. Sections 3.5 and 3.6 discuss the implications of the chapter within the task of LMTC and summarise our conclusions.

This work has been published as part of Falis et al. (2021) and Falis et al. (2022) with myself as the lead and first author with supervision and guidance from my co-authors. Doctor Hang Dong, beyond continuous discussion of the ideas provided access to the HLAN model (Dong et al., 2021a) and its results on the MIMIC-III dataset. The text of the publication has undergone minimal changes with regard to methods, results, and conclusions based on feedback from the supervisory team. An expanded introduction and background sections were written for the thesis.

3.2 Background

Automated medical document coding, *e.g.*, ICD coding, is an example of a *Large-Scale Multi-Label Text Classification* (LMTC) task. Given an input document in an unstructured or semi-structured format (*e.g.*, with standard subsections/headers), the task is to assign one or more labels to the document, usually from a large pool of labels often coming from a structured label space, such as an ontology. The goal is to indicate the presence of concepts within the document as a set of labels, rather than indicating each individual positive mention of each concept – *e.g.*, a discharge summary with labels indicating that the patient suffers from “Type 1 Diabetes without complications”, and an “acute myocardial infarction”.

Named Entity Recognition and Linking (NER+L) is the combination of two tasks

– identifying words of interest – or named entities – (NER) and classifying them according to a standardised vocabulary/ontology (L). The task is performed on the word or phrase level, with the output being a list of mentions indicated by indices and the associated classes. For instance, in the sentence "The patient suffers from T1D." the entity mention of "T1D" on the index range of 25:28 (assuming 0-indexing) should be retrieved and linked to the concept *250.01: Diabetes mellitus without mention of complication, type I [juvenile type], not stated as uncontrolled* in ICD-9 (or a different controlled vocabulary, such as the UMLS). There can be multiple mentions of the same concept presented in a variety of surface forms (*e.g.*, "Type 1 Diabetes", "Type 1 Diabetic", or "T1D"). All mentions should be retrieved (their uniqueness presented via their respective index ranges) and associated with the concept within the vocabulary. Similar to LMTC, in NER+L the aim is to assign labels from a standardised vocabulary/ontology/label space to unstructured/semi-structured text. While individual predictions and gold standard labels on the word-level are unique and associated with a range within the text (*strongly-labelled scenario*) in the case NER+L, in LMTC the comparison between the predictions and gold standard are on the document level as sets (*weakly-labelled scenario*). An analogy in image processing can be made, where given a picture containing three dogs and two cats with the label space being a set of species of animals, the equivalent of LMTC is expected to report that the picture contains at least one dog and at least one cat, while the equivalent of NER+L is expected to segment the image to indicate the individual dogs and the individual cats as separate entities and assign them the corresponding species' label.

While hierarchical evaluation incorporating the structure of the label space is not commonly used in LMTC, some prior art exists. Kosmopoulos et al. (2015) propose an evaluation metric propagating the logical OR of binary values of child nodes to their ancestors (further explained in Section 3.3.2.1). With regard to error analysis, extensions to confusion matrices for hierarchical weakly-labelled scenarios were proposed over the course of the development of our methods (Görtler et al., 2021; Heydarian et al., 2022). Görtler et al. (2021) propose a method of analysis in a hierarchical multi-output setting, approaching high-dimensional confusion as a distribution. Heydarian et al. (2022) extend the standard confusion matrix for multi-label classification in a non-hierarchical setting.

3.3 Methods

In a strongly-labelled scenario, such as NER+L, each gold standard label and prediction are associated with a particular identifying section of the input data – *e.g.*, a span indicated by indices in text (such as presented in Figure 3.2), or a patch of pixels in an image. Hence it is possible to create a correspondence between predictions with gold standard labels through overlaps in these identifying sections – be it a perfect match, partial overlap with at least one gold standard label, or no overlap at all. By having predictions associated with gold standard labels, error analysis can be performed to explore the mispredictions which tend to arise when the model deals with spans associated with a given gold standard label.

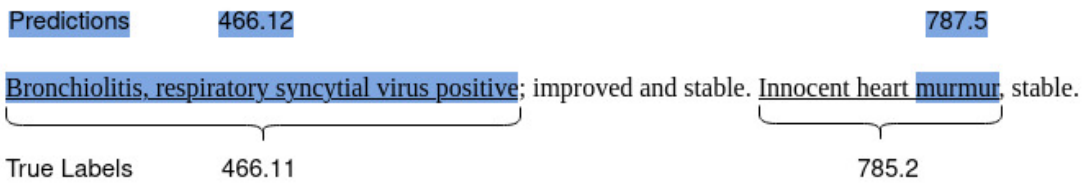

Figure 3.2: An example of a strongly-labelled scenario – *Named Entity Recognition and Linking* (NER+L).

Correspondence between predictions and gold standard labels within the weakly-labelled scenario is possible if the task is restricted to a single prediction and a single gold standard label per data point. The LMTC scenario, involving multiple gold standard labels and multiple predictions per datapoint, does not allow such analyses in a straightforward manner. Two sets of labels are presented – the prediction and the gold standard (as presented in Figure 3.3); upon which subsets can be produced using binary set operations – the intersection between the sets represents the correct predictions (true positives) and the differences between them – predictions absent from the gold standard (false positives), and gold standard labels absent from the prediction (false negatives) – represent errors. These measurements then allow calculation of the precision, recall, and F_1-score of the model's predictions. While this is a valid way to quantify model performance (be it on a label-by-label basis, or averaged across the labels), it is lacking in options for error analysis.

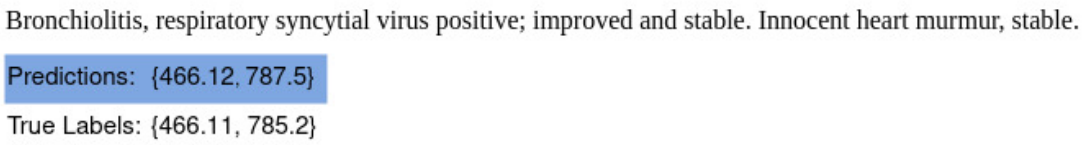

Figure 3.3: An example of a weakly-labelled scenario (Classification).

3.3.1 Baseline

Previous work in neural ICD coding and LMTC in general treats the output layer of the model as $|L|$ individual units, where L represents the label space (CAML by Mullenbach et al. (2018), often used as a baseline in ICD coding, has an output layer of $|L|$ sigmoid predictions without any interaction between them within the layer). Models are evaluated using precision (*Eq.* 3.4), recall (*Eq.* 3.5) and F_1-score (*Eq.* 3.6). The metrics are calculated using the number of true positives (*Eq.* 3.1), false positives (*Eq.* 3.2), and false negatives (*Eq.* 3.3) based on the prediction (Z_d) and gold standard (Y_d) sets for each document (d) in the set of evaluation documents (D).

$$TP_d = |Z_d \cap Y_d| \tag{3.1}$$

$$FP_d = |Z_d - Y_d| \tag{3.2}$$

$$FN_d = |Y_d - Z_d| \tag{3.3}$$

$$P = \frac{\sum_{d=1}^{D} TP_d}{\sum_{d=1}^{D} TP_d + FP_d} \tag{3.4}$$

$$R = \frac{\sum_{d=1}^{D} TP_d}{\sum_{d=1}^{D} TP_d + FN_d} \tag{3.5}$$

$$F_1 = \frac{2 \cdot P \cdot R}{P + R} \tag{3.6}$$

In their basic and most common form, these metrics are used for comparing the predictions gold standard. As the predictions are real-valued, while the gold standard is binary, predictions are binarised through thresholding. An alternative means of evaluation is to, rather than considering the entire prediction vector, focus on the k predictions with the highest sigmoid outputs. Mullenbach et al. (2018) motivate the use of precision at k ($P@k$) by the potential use case of the automated coding system predicting the k most likely codes to be assigned and subsequently having them reviewed by a human coder. Rios and Kavuluru (2018b) argue for the use of $P@k$ and recall at k ($R@k$) due to being independent of a specific threshold. In their evaluation focusing on the subset of the label set depending on label population – frequent ($f > 5$ where

f is the population of the label in the training set), few-shot ($5 \geq f > 0$) and zero-shot ($f = 0$) – they highlight the usefulness of $R@k$ over $P@k$ for few-shot and zero-shot scenarios as $P@k$ tends to decrease to 0 rapidly with the increase of k to values higher than the number of subset-specific-labels assigned to each instance (by virtue of being infrequent, labels from the few-shot and zero-shot subset are less likely to appear leading to high values of k being detrimental in $P@k$). Chalkidis et al. (2019a) further use the R-Precision@K ($RP@k$) with the top k' predictions evaluated, where k' is the minimum between k and the number of gold standard labels assigned to the document. Hence, if k is lower or equal to the number of gold standard labels, $RP@K$ is equal to $P@K$. If k exceeds the number of gold standard labels, k' is set to the number of gold standard labels and hence the metric does not suffer the drop associated with high k in $P@k$ noted by Rios and Kavuluru (2018b).

The value of k may vary depending on the parameters of the dataset, especially the average or median number of labels per input document – *e.g.*, Mullenbach et al. (2018) use $k = 15$ for their MIMIC-III experiments with the full label set, as 15 roughly corresponds to the mean number of labels per document in this dataset. Investigating subsets of the label set, such as by Rios and Kavuluru (2018b) can further reduce the number (Rios and Kavuluru (2018b) use $R@5$ and $R@10$). Chalkidis et al. (2020) use $k = 15$ for MIMIC-III (full codeset), but a smaller $k = 5$ for the other two evaluated LMTC datasets (EURLEX57K (Chalkidis et al., 2019b), and AMAZON13k (McAuley and Leskovec, 2013)).

Micro-averaging (*Eq.* 3.7) (where M stands for metric – and can be Precision or Recall) treats each individual prediction with equal weight, skewing the overall result in favour of high-population classes (*e.g.*, *401.9: Essential hypertension, unspecified* in MIMIC-III as presented in Section 2.3). Macro-averaging (*Eq.* 3.8), on the other hand, first computes the performance for each unique label within the label space, and averages across the label space thereby giving the results from each label equal weight regardless of their population – this consequently means that poor performance on the long tail becomes more visible within the result. Macro-averaging provides an understanding of how well the model can be expected to perform for a label in the labelspace selected uniformly at random (rather than based on its frequency in data), removing the skew in micro-averaging introduced by high-population labels.

Our primary evaluation metrics common with the majority of previous work are

micro-F_1 (*Eq.* 3.9) and macro-F_1 (*Eq.* 3.10) scores.

$$M_{micro} = M\left(\sum_{l=1}^{|L|} TP_l, \sum_{l=1}^{|L|} FP_l, \sum_{l=1}^{|L|} FN_l\right) \tag{3.7}$$

$$M_{macro} = \frac{1}{|L|}\sum_{l=1}^{|L|} M(TP_l, FP_l, FN_l) \tag{3.8}$$

$$F_{1(micro)} = \frac{2 \cdot P_{micro} \cdot R_{micro}}{P_{micro} + R_{micro}} \tag{3.9}$$

$$F_{1(macro)} = \frac{2 \cdot P_{macro} \cdot R_{macro}}{P_{macro} + R_{macro}} \tag{3.10}$$

To illustrate the difference between micro and macro averaging we present an example: If we collect data from general practitioners during the influenza season, many of the patients will present with influenza, but some will present with less common diseases – let's say we have 100 patients, out of which 96 present with the same common disease (influenza), while the remaining 4 present with different less common diseases. Suppose that we built a model on a dataset with similar distributions of diseases, and evaluate it on this dataset of 100 patients. Furthermore, suppose that the model – given its exposure to many influenza patients, while having little-to-no exposure to the less common diseases – performs perfectly in predicting influenza, while failing to predict any of the other diseases. In a micro-averaged scenario, this illustrative example would result in a high P, R, and F_1 of 0.96, which could be interpreted as a well-performing model, in that if we are condition-agnostic, most of the time, the model predicted the condition correctly. For macro-averaging, in our illustrative example, the performance for influenza would be a perfect 1, while for the other four conditions it would be 0 resulting in an average of $\frac{1+0+0+0+0}{5} = \frac{1}{5} = 0.2$.

This basic approach to evaluation summarises the model's performance, and it is correct to try and maximise it – not only for claiming State of the Art, but also the general goal of building an accurate model and getting as many correct predictions as possible. This, however, only fulfills one objective of evaluation, leaving very little information for error analysis – we only recognise False Positives and False Negatives without any option of further analysis. Treating individual label predictions as independent entities then means that all errors of the same type are indistinguishable.

Assume a patient presenting with a myocardial infarction and type 1 diabetes, and two models – A and B – employed to code the patient's discharge summary. Model A

predicts that the patient is to be coded with heart failure and type 1 diabetes. Model *B* predicts that the patient is to be coded with alcohol dependency and type 1 diabetes. Both models produced one correct prediction and one incorrect prediction and hence the baseline evaluation approach considers the model performance to be equivalent – in each case we have one true positive (type 1 diabetes), one false positive prediction (heart failure and alcohol dependency respectively), and one false negative (the missed heart attack). However, we can argue that, in fact, the models' errors are different – Model *A*, while not predicting myocardial infarction correctly, predicted a different cardiovascular condition, while Model *B*'s prediction did not involve any cardiovascular conditions.

This example inspires two streams of research for the following two research questions – can we somehow represent our knowledge of the conditions such that we can award partial credit to situations where a non-True-Positive label in either the prediction or gold standard set is close to a label in the other set? And can we track what labels tend to be mispredicted as other similar labels?

3.3.2 Hierarchical Evaluation

Firstly, we tackled the question of the partial credit by incorporating the structure of the label space in the computation of P, R, and F_1 in order to quantify the closeness of a prediction to the gold standard labels. In a structured label space, the labels exist within a connected graph (or multiple graphs, such as Conditions and Procedures in the ICD), and some labels are inherently closer to each other. Furthermore, in tree-structured ontologies, such as ICD-9 and ICD-10, we recognise the so-called *true-path rule* which states that if a node in the hierarchy is assigned to be true, all of its ancestors are also considered to be true. For instance, a patient suffering from ICD-9 *425.0: Endomyocardial fibrosis*, is more generally considered to be suffering from the more abstract concepts represented by the label's ancestors within ICD-9: *425: Cardiomyopathy*; *390-459: Diseases Of The Circulatory System*; and the root of ICD-9-CM *Diagnosis Codes* (rather than procedure codes).

3.3.2.1 Set-Based Hierarchical Evaluation

Previous work within the scope of weakly-labelled scenarios propose *set-based* measures (Kosmopoulos et al., 2015), incorporating the true-path rule. The original prediction set and gold standard set are augmented by adding all of the ancestor labels

into the respective sets. In mathematical terms:

Let Z' represent a set of N' ICD codes individually denoted as z'_1, z'_2, ... $z'_{N'}$. Let $AN_j(z'_i)$ denote a function that returns a set of all ancestors of the code z'_i up to the depth j within the ontology. The augmented set $Z'_{(aug)}$ consists of the union of the original codes in Z' and their ancestors *Eq.* 3.11.

$$Z_{(aug)} = Z \cup \{AN_j(z_i)|1,...,N'\} \tag{3.11}$$

In set-based hierarchical evaluation we calculate the Precision, Recall and F_1-score with their standard formulae based on re-defined TP (*Eq.* 3.12), FP (*Eq.* 3.13), and FN (*Eq.* 3.14).

$$TP_{(set)d} = |Z_{(aug)d} \cap Y_{(aug)d}| \tag{3.12}$$

$$FP_{(set)d} = |Z_{(aug)d} - Y_{(aug)d}| \tag{3.13}$$

$$FN_{(set)d} = |Y_{(aug)d} - Z_{(aug)d}| \tag{3.14}$$

As the result of these set augmentation procedures are sets – same as in the baseline metrics (Equations 3.1, 3.2, and 3.3) – standard baseline evaluation metrics can be directly applied to these augmented sets, treating ancestor-level binary data the same as prediction-level binary data. The inclusion of ancestor data within the evaluation means that a misprediction on a low-level within the hierarchy can still result in a match on a higher level awarding partial credit. Thus, in our example patient, myocardial infarction and Model *A*'s predicted heart failure will match on the parent concept of cardiovascular disease (and subsequently all further ancestors), while myocardial infarction and the alcoholism predicted by Model *B* will only match on the root, resulting in a higher reported performance for Model *A*.

3.3.2.2 Count-Preserving Hierarchical Evaluation (CoPHE)

In *Count-Preserving Hierarchical Evaluation* (CoPHE) the ancestor labels are associated with the number of descendant codes present in the respective sets. This is handled by re-defining TP, FP, and FN per document (d) per code family (c) (*Eq.* 3.15, 3.16, 3.17). Note the difference between the CoPHE equations compared to those of set-based hierarchical evaluation (3.12, 3.13), 3.14)) – while set-based evaluation equations return the sizes of results of binary operations on sets (*i.e.*, a binary operation is performed on the prediction and gold standard sets and the size of the resulting set is returned), in CoPHE the size of the respective augmented sets are calculated first resulting in natural numbers upon which further arithmetic operations are performed.

The per-document value of TP, FP, and FN is the sum of the respective per-code-family metrics drawn from the leaf-level members of the family (example for TP in *Eq.* 3.18]). The F_1-scores for CoPHE are then calculated based on the precision and recall calculated from the CoPHE versions of TP, FP, and FN. Preserving counts allows tracking whether the correct number of leaf-level descendants was predicted, or whether there is under-/over-predictions. The concept of under-/over-prediction within the thesis corresponds to situations where a model respectively tends to predict fewer or more descendants of a node within the label space than expected according to the gold standard. An example of over-prediction is a model predicting three different instances of hypertension *401.0: Malignant essential hypertension*, *401.1: Benign essential hypertension*, *401.9: Unspecified essential hypertension* (all leaf codes descended from *401: Essential hypertension*; mutually exclusive) when only *401.0: Malignant essential hypertension* is expected. If a model is meant to predict multiple codes within a family – *e.g.*, in the case of diabetes where multiple codes from a single family may appear due to the possibility of multiple complications (*e.g.*, neurological, ophthalmic, renal) – but predicts fewer – *e.g.*, only capturing that the patient is diabetic, but failing to predict the complications (or potentially predicting them as codes belonging to a different family of codes, *e.g.*, relating to the anatomy affected by the diabetic complication) resulting in fewer descendants of a particular code being predicted than expected in the gold standard – this constitutes and under-prediction.

$$TP_{(CoPHE)c,d} = min(|Z_{(aug)c,d}|, |Y_{(aug)c,d}|) \tag{3.15}$$

$$FP_{(CoPHE)c,d} = max(|Z_{(aug)c,d}| - |Y_{(aug)c,d}|, 0) \tag{3.16}$$

$$FN_{(CoPHE)c,d} = max(|Y_{(aug)c,d}| - |Z_{(aug)c,d}|, 0) \tag{3.17}$$

$$TP_{(CoPHE)d} = \sum_{c=1}^{|C|} TP_{(CoPHE)c,d} \tag{3.18}$$

Figure 3.4 shows a family of codes in a coding example indicating the gold standard and predicted labels via line and border style. The vector representation of this example is presented in Figure 3.5. The evaluation statistics for the performance overall and on different levels of the ontology in this scenario are presented in Table 3.1.

While parallels can be drawn between the design of the set-based hierarchical evaluation and CoPHE, perhaps the connection between the true positive derivations – Equations 3.12 and 3.15 for set-based hierarchical evaluation and CoPHE respectively

– is the least obvious. A high-level node in set-based evaluation is assigned a positive binary value if there is at least one descended node predicted and at least one descended node expected in the gold standard (note that these do not need to be the same descended node). Since the ancestor augmentation procedure (Equation 3.11) propagates the binary values of descendants up, if descendants of an ancestor node appear both in the prediction and gold standard, the ancestor node will appear in both the augmented prediction and augmented gold standard and hence will be captured by the intersection operation in Equation 3.12 as a true positive. Note that Equation 3.12 works with augmented sets and returns the overall number of true positives for the entire label set. In the case of the CoPHE approach to calculating True Positives (Equation 3.15), this is defined on the level of a node, and the overall count for the label set is further calculated with Equation 3.18. The calculation of True Positive values on a node-by-node basis (Equation 3.15) calculates the number of descended nodes predicted and expected in the gold standard by applying the ancestor augmentation to the respective sets and subsequently returning the sizes of these augmented sets. This represents how many descendants of a given node were predicted versus expected. The minimum operation between the sizes of these two sets (natural numbers) is analogous to the intersection in its set-based counterpart (Equation 3.12) – which on a node-level is a binary value. If the sizes of the augmented sets are equal (as many descended codes predicted as expected) the value is trivially equal to the sizes of the sets. If the augmented sets differ in size, the number of true positives is the overlap – the size of the smaller set, while the difference between the sizes is represented as false positives if there are more predictions than gold standard labels, or false negatives if more labels were expected in the gold standard than predicted by the model.

While the definitions of CoPHE draw upon and contrast with the set-based hierarchical evaluation metric, in practice these metrics are not directly comparable – they capture different behaviour of the model. It is the fact that the behaviours being captured are related – presence of descendant leaves of a given family in set-based versus the count of descendant leaves of a given family in CoPHE – that allows to see the effects of over- or under-prediction when viewing these metrics side-by-side. Hence, while the motivation for use of any of the two hierarchical metrics is the ability to assign partial credit to predictions which would be considered purely false in a leaf-only evaluation, using CoPHE alongside (rather than instead of) set-based hierarchical evaluation further provides data on the extent to which a model tends to predict more or fewer codes from a given family than expected in the gold standard.

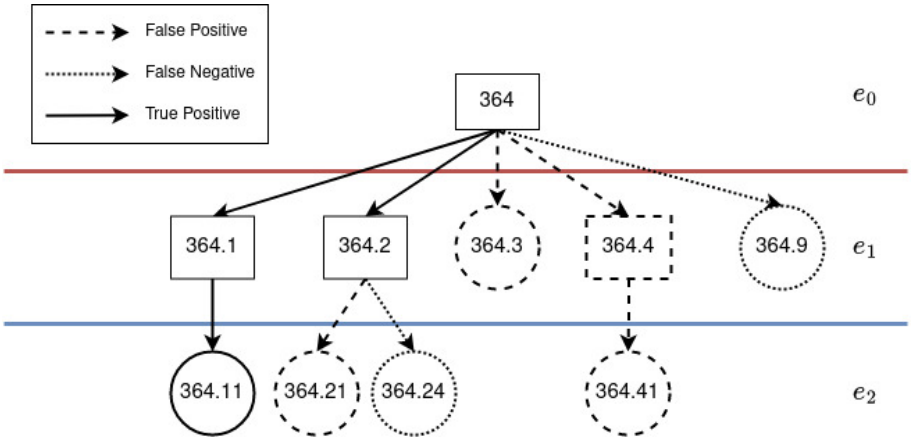

Figure 3.4: An example of hierarchical evaluation. Circular nodes represent leaf nodes (for non-hierarchical evaluation), borders of nodes represent set-based hierarchical evaluation, edges represent count-preserving hierarchical evaluation. Solid lines represent TP, dashed-lines represent FP, dotted lines represent FN. Levels of depth in the ontology (e_0, e_1, and e_2) are indicated with horizontal lines.

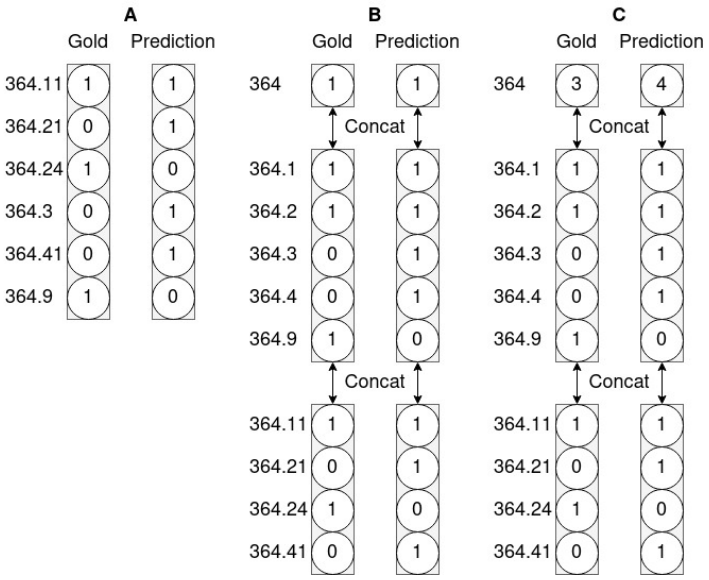

Figure 3.5: A comparison between three styles of evaluation: **(A)** Evaluation performed only on leaf nodes (no use of hierarchical relations); **(B)** Set-Based Hierarchical Evaluation (Kosmopoulos et al., 2015). Four descendants of the node *364* are predicted, and three appear in the gold standard (corresponding to the situation presented in Figure 3.4). This is reflected in the middle-level vector, but the information is lost in the top level; **(C)** Count-Preserving Hierarchical Evaluation. The numeric information of predictions and gold labels are preserved on higher levels.

	TP	FP	FN	P	R	F_1
Leaf-Only	1	3	2	0.25	0.333	0.284
Set-Based						
e_2	1	2	1	0.333	0.5	0.4
e_1	2	2	1	0.5	0.667	0.571
e_0	1	0	0	1	1	1
Overall	4	4	2	0.5	0.667	0.571
CoPHE						
e_2	1	2	1	0.333	0.5	0.4
e_1	2	2	1	0.5	0.667	0.571
e_0	3	1	0	0.75	1	0.86
Overall	6	5	2	0.545	0.75	0.631

Table 3.1: Evaluation of the three representations presented in Figure 3.5. In the case of Set-Based evaluation and CoPHE we present both the evaluation at each level of the ontology, and overall evaluation across levels.

Figures 3.6, and 3.7 present different scenarios and the behaviour of the analysed metrics within them. Notably, Figure 3.6 describes a situation where set-based evaluation yields a higher P (and consequently F_1 as the presented example constitutes perfect recall) than CoPHE, due to the occurrence of overprediction. In contrast, Figure 3.7 shows a situation where the count-preserving feature of CoPHE is visible only in the highest level with every other level being identical to set-based evaluation. In this situation the number of FPs and FNs remains the same, while the number of TPs increases due to label reconciliation in the highest level, resulting in higher P, R, and F_1 for CoPHE.

Both set-based evaluation and CoPHE aim to reconcile mispredicted labels at higher-levels of the hierarchy. This reconciliation may not happen in the first few levels, especially if the ontology comes with much detail. Consider ICD-10-CM disease codes with 4-character compared to 2-character aetiologies in ICD-9-CM that can be mapped to each other (describing close enough concepts) – assuming the same level of detail is presented on the head-code level in both ontologies, and a situation where a predicted and expected code match on the head-code but completely mismatch on the aetiology at the maximum length of their respective ontology, the relatively later reconciliation on the head code for ICD-10-CM (by two levels) will result in further contribution of false positives and false negatives before reaching the single true positive on the head-code level (the same single true positive that would be reached in the case of ICD-9-CM with fewer levels between maximum-length-aetiology leaf and head code). Thus, applying these methods does not necessarily lead to higher scores than in the

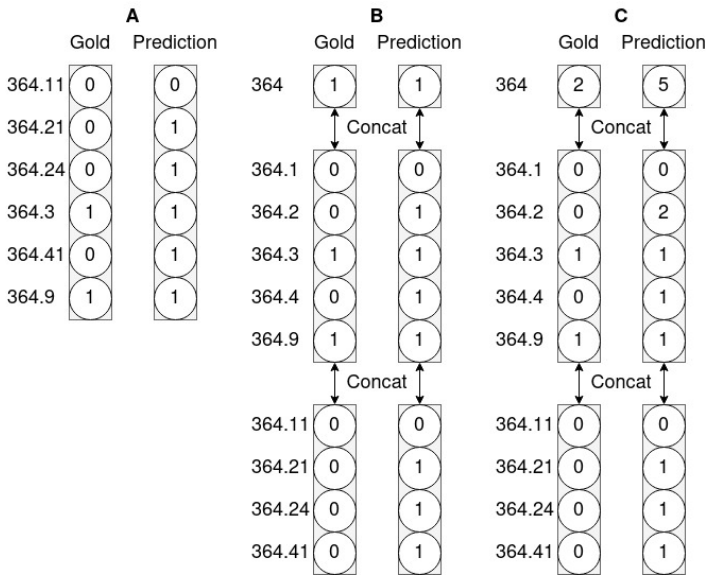

Figure 3.6: A comparison between the three styles of evaluation demonstrating overprediction within a code family 364 (two descendants expected and five predicted). The vector representations correspond to **(A)** Leaf-only evaluation (no use of hierarchical relations); **(B)** Set-Based Hierarchical Evaluation; and **(C)** Count-Preserving Hierarchical Evaluation. The resulting set-based P and F_1 are higher than for CoPHE (0.375 vs 0.273 and 0.7 vs 0.273), due to CoPHE penalising mismatches in counts on higher levels (over-/underprediction).

case of the baseline (leaf-only) evaluation. Set-based evaluation and CoPHE can be applied either on a particular level of hierarchy, or aggregated over multiple levels (as seen in Table 3.1). Hence, the level of the hierarchy on which the augmentation of the prediction and gold standard vectors is performed (or a cut-off level if aggregating over multiple levels) is a parameter of these metrics and affects the final result. Choosing a low cut-off (*e.g.*, before reaching a head code) may result in mismatched codes only propagating their FPs and FNs without ever being reconciled thereby lowering the precision and recall in hierarchical measures. An example of such a situation is presented in Figure 3.8 – should the top level (*364*) not have been considered, only false positives and false negatives would have increased in population through augmenting the prediction and gold standard vectors (with true positives unchanged). For this reason we recommend investigating the codeset used and determining the granularity at which CoPHE should be applied in order to maximise its usefulness.

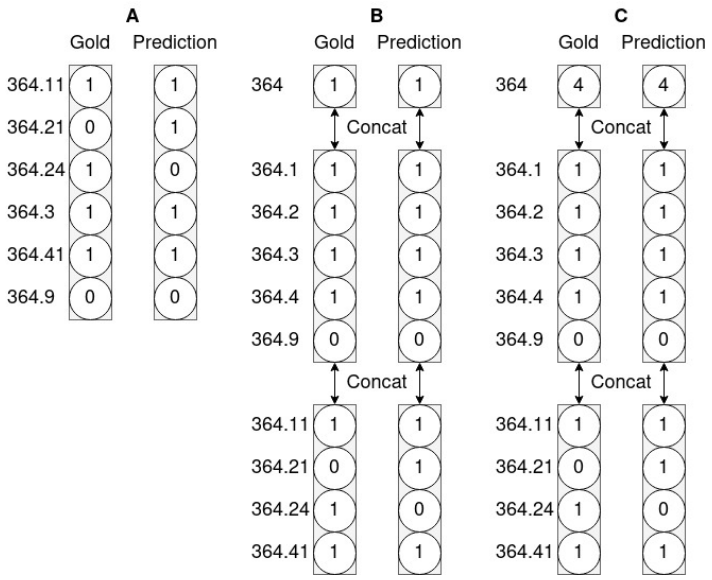

Figure 3.7: A comparison between the three styles of evaluation in a situation where applying the hierarchical measures results in the number of false positives and false negatives matching in the hierarchy-enhanced vector representations while having differing numbers of true positive due to the differences in approaches to representing hierarchy. Vector representations correspond to **(A)** Leaf-only evaluation (no use of hierarchical relations); **(B)** Set-Based Hierarchical Evaluation; and **(C)** Count-Preserving Hierarchical Evaluation. The number of FP and FN is the same between in **(B)** and **(C)**, with a higher number of TP for **(C)**. This results in lower P, R, and F_1-scores for set-based compared to CoPHE (0.875 vs 0.909).

3.3.3 Confusion Matrices

3.3.3.1 Baseline

Confusion matrices (Tan et al., 2019) are a useful evaluation analysis tool for scenarios where correspondence can be drawn between the prediction and gold standard. Such correspondence can arise either due to linking predictions to gold standard labels via strong labelling (*e.g.*, NER+L); or constraints, such as every input possessing a single gold standard label and expecting a single output prediction making the correspondence trivial (*e.g.*, single-label classification).

The rows and columns of a confusion matrix correspond to the possible labels that appear in the gold standard and predictions. Some scenarios enforce pairing between the gold standard and predictions such that each gold standard label will be paired with at least one prediction and vice versa – *e.g.*, single-label classification. Others allow

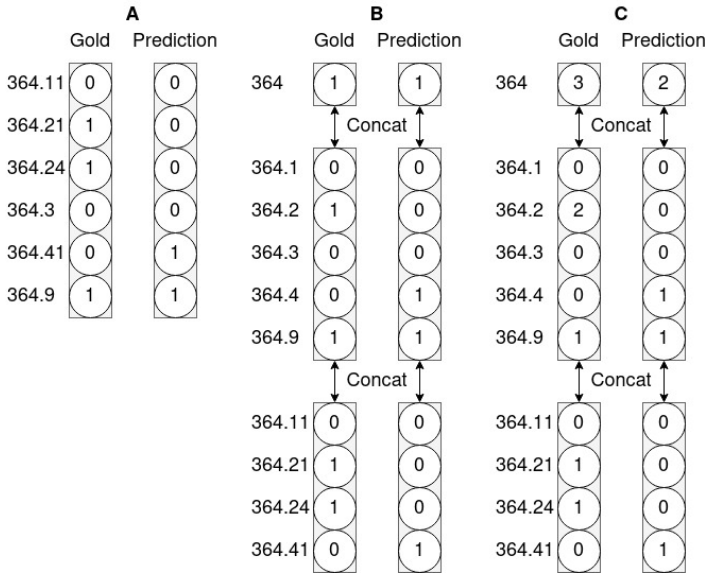

Figure 3.8: A comparison between the three styles of evaluation in a situation where the true positive matches arising from reconciling false positives with false negatives in code families appear on higher levels of the ontology. Vector representations correspond to **(A)** Leaf-only evaluation (no use of hierarchical relations); **(B)** Set-Based Hierarchical Evaluation; and **(C)** Count-Preserving Hierarchical Evaluation. While FPs and FNs are propagated from lower levels, they are reconciled into TP only in the head code *364*. F_1 for the hierarchical method is at least as high as that of the leaf-only evaluation. However, if the cutoff level was below the head code (*i.e.*, statistics for *364* were not considered), the FPs and FNs appearing from the lowest level with double aetiology would not have been reconciled, and the results for hierarchical evaluation metrics would be lower than those of leaf-only evaluation.

situations where a predicted label may not have a gold standard counterpart (or vice versa) – *e.g.*, in NER+L if a particular named entity is failed to be identified, and thus not assigned a class, or a spurious entity being identified. These can be then expressed by a default *"Outside"* label, indicating a false positive (the model predicts an entity with a class within the task label space for an irrelevant input – overprediction), or false negative (the model fails to identify and assign a class to a relevant input – underprediction). If the presentation of the order of labels is the same for the rows and the columns, the resulting matrix contains true positive counts on the diagonal, with misclassifications appearing off the diagonal. An example of a baseline confusion matrix can be seen in Figure 3.9. Compared to evaluation approaches providing P, R, and F_1 score statistics, be it on individual classes or micro-/macro-averaged across the

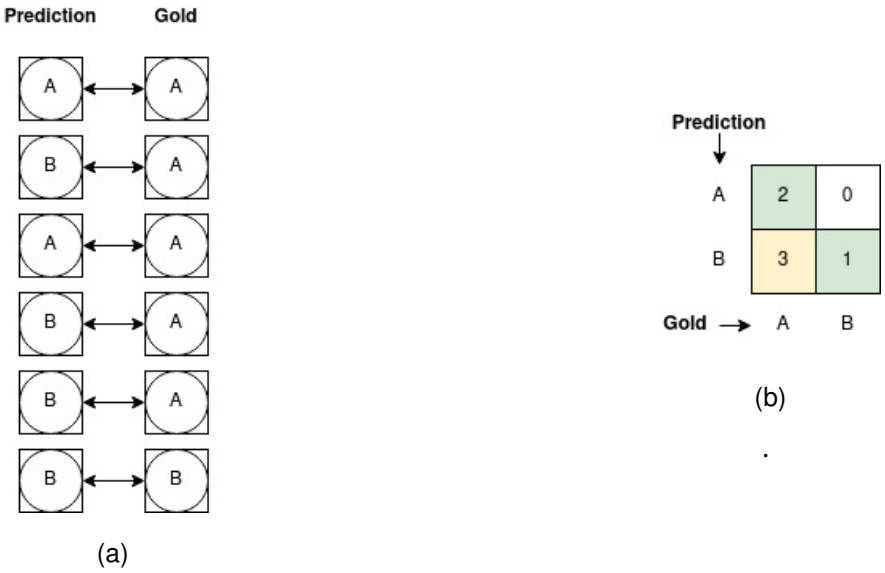

Figure 3.9: (a) An example of a scenario suitable for analysis via a confusion matrix: A single-label binary classification task evaluated on 6 datapoints. For each datapoint there is a single gold standard label and a single prediction allowing trivial correspondence. (b) A confusion matrix created from the data presented in Subfigure 3.9a. Diagonal entries (True Positives) presented as green, confusion (non-diagonal non-zero) presented as yellow. As the setting did not involve false positives or false negatives necessitating an *Outside* label, the confusion matrix does not include it. We can see that, while all the gold standard instances of B are correctly classified, the model confuses between A and B in gold standard instances of A.

labelspace, the confusion matrix provides data on performance on classes as pairs. A high misclassification between two classes indicates that, with respect to the model parameters and the data (be it training or evaluation), instances of the classes are similar and difficult to distinguish. The output of such model analysis can be utilised in further model design, or distributed as supplementary information in order to inform better use of a deployed model.

3.3.3.2 Weak Hierarchical Confusion Matrix

Within the LMTC task, such as ICD-9 coding, both predictions and gold standard labels are presented as sets. The correspondence between the labels in these sets does not arise through an identifier (as it would in strongly-labelled scenarios), nor is it trivial (as in single-label classification). Hence, if there is a mismatch between the prediction and gold standard labels, the correspondence for the misclassifications is not

trivial – one or multiple false positives may correspond to one or more false negatives (or even none), and vice versa. From the viewpoint of the correspondence between the input text and the predicted versus expected output, the features of the text that would indicate a particular false negative label in the gold standard may have contributed to the prediction of one or more different codes (or none at all). This presents an issue for error analysis – what errors occur with what labels?

Consider an example datapoint with gold standard labels A.1, A.2, A.3, and B.1, while a model predicts A.1, A.3, and A.4 (presented in Figure 3.10). First we will construct the co-occurrence matrix between the prediction Z and gold standard Y by marking all possible pairings between individual predicted and gold standard labels (a total of $|Z| \times |Y|$ pairs). We then apply three assumptions:

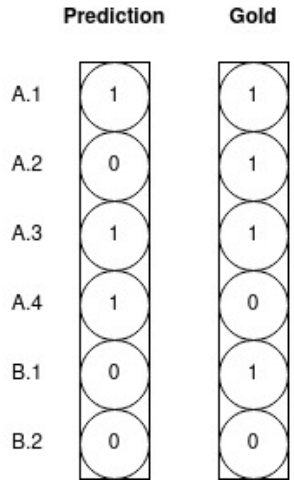

Figure 3.10: An example of a comparison of a prediction and gold standard set for an LMTC setting with labels belonging to two families – A.1, A.2, A.3, and A.4 to family A; and B.1, and B.2 to B.

1. **1-to-1 True Positive Correspondence**: If a label is present both in the prediction and gold-standard for a document, this is a True Positive (TP), and not considered for confusion (green cells in Figure 3.11).

2. **Within-Family Confusion**: non-TP codes in the prediction are matched with non-TP codes in the gold standard within their respective code families. Cross-family pairings (represented as black cells in Figure 3.11) are ignored.

3. **The Out-Of-Family Scenario**: If in withing-family confusion matching a code from prediction/gold cannot be matched (there is no code from its family left to

match in the other set), the code is associated with a special *Out of Family* (OOF) code (see the red cell in Figure 3.11).

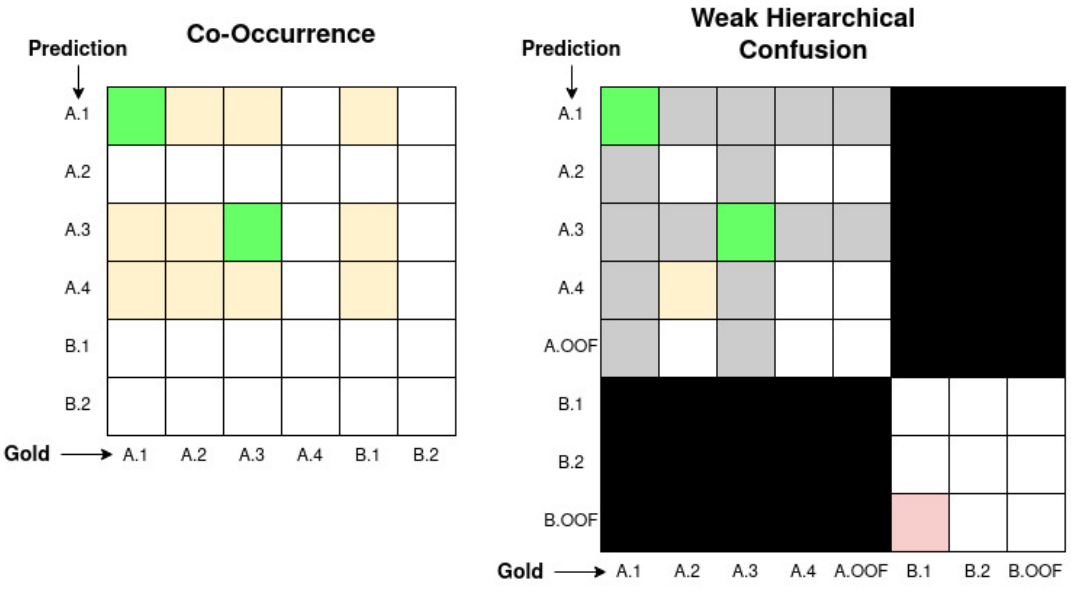

Figure 3.11: Left: A simple co-occurrence matrix between the prediction and gold standard labels for two label families for a single document. Labels A.1, A.3, and A.4 are predicted, while codes A.1, A.2, A.3, and B.1 are in the gold standard. Green cells indicate a match between the prediction and gold standard, yellow cells indicate a mismatch. Right: A weak hierarchical confusion matrix constructed from the co-occurrence matrix with the use of the three assumptions – Gray cells were eliminated via 1-to-1 correspondence, black cells were eliminated via within-family-confusion, red cells indicate the OOF scenario. The resulting confusion matrix indicates A.1 and A.3 being correctly predicted (green), B.1 being a false negative – an OOF (red), and the predicted code A.4 being confused with expected code A.2 (yellow).

Note that these are indeed assumptions and may not reflect true model behaviour. There may not be 1-to-1 correspondence between true positive labels – in fact the very scenario of over-/under-prediction we explore through CoPHE constitutes situations where there is a difference between the number of expected and predicted labels from a family. Assume a situation where the gold standard includes only the code *401.0: Malignant essential hypertension*, while the prediction set includes both *401.0: Malignant essential hypertension* and the highly populous code *401.9: Essential hypertension, unspecified*. In this case it is likely that the mispredicted code *401.9* is linked with the gold standard's *401.0* (the model having predicted multiple different instances of hypertension). Given the 1-to-1 correspondence assumption for true positives, however,

401.0 is already matched and not considered for within-family confusion and *401.9* will be matched with an OOF. Hence, the Out-Of-Family scenario does not constitute only situations where codes from unexpected families are predicted, but also overprediction/underprediction within families, assuming no mismatches can be drawn post removal of true positives. Secondly, there may be relevant confusion patterns between codes that are related through a more high-level node in the ontology – such as in a situation where concepts are semantically similar while being far apart in the ontology (*e.g.*, due to ordering of splitting rules within the taxonomy). For instance, some codes may reference a dependence on a different concept which may be elsewhere in the hierarchy (*e.g.*, *796.2: Elevated blood pressure reading without diagnosis of hypertension* shares no common ancestors with *401.9: Unspecified essential hypertension* beyond the root). The scope of a family can be adjusted within the WHCM to be on higher levels of the ontology (chapter or the root of conditions) to include such connections, but larger family scopes will lead to capturing less detail by reducing the impact of the second assumption on the co-occurrence matrix.

It is also important to note that Assumption 2 results in a many-to-many correspondence. Hence a singular non-TP label in one set can be associated with none (OOF), one, or more than one in the other. This means that the individual counts in within-family confusion for an input document do not correspond to the exact number of false positives or the exact number of false negatives, but rather the product between the number of false positives and false negatives within the family. Figure 3.12 shows an example of this many-to-many correspondence.

These weak hierarchical confusion matrices can be further aggregated to represent the proportion of true positives, within-family errors, and out-of-family errors within a given family. Similar to the Precision, Recall, and F_1-score, these can then be aggregated through micro- or macro-averaging. Given a weak hierarchical confusion matrix for a particular family of codes, it can be investigated what proportion of predictions of a code are made correctly (Assumption 1), result in a misprediction within the code family (Assumption 2), or constitute the OOF scenario (Assumption 3). These can be calculate on the level of individual codes and then macro-averaged. Further interesting statistics can be drawn – *e.g.*, what label is most likely to match with the predicted label, and whether this corresponds to a True Positive (a binary per-label metric we refer to as *Match*). These statistics computed on the prediction inform the user of the system how to interpret and how much to trust predictions of certain codes alongside the standard metrics, such as the system's precision. Conversely, statistics can be

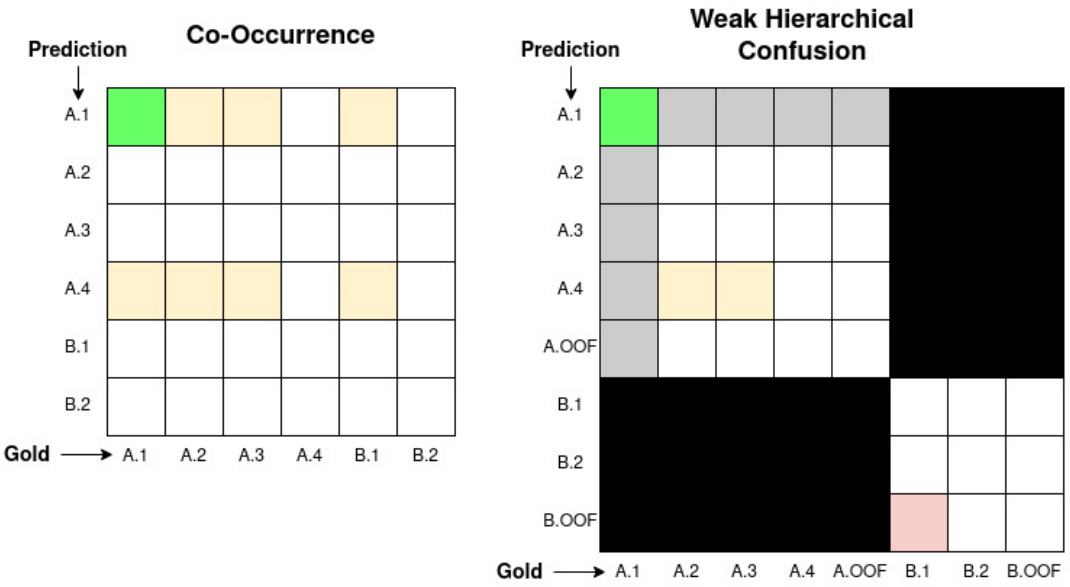

Figure 3.12: An example of the many-to-many correspondence of within-family confusion. Codes A.1 and A.4 are predicted, while codes A.1, A.2, A.3, and B.1 are in the gold standard. After applying Assumption 1 (True-positive 1-to-1 correspondence) we and Assumption 2 (within family confusion) we arrive at the predicted code A.4 being confused with the gold standard codes A.2 or A.3.

drawn predicated on labels within the gold standard, expanding the user's knowledge on the model's recall. Comparing models' distribution of errors between OOF versus within-family confusion should follow the intuition that errors in OOF are worse than within-family errors, as the latter means that labels from the right family were assigned, whereas the former point to over/underprediction.

3.4 Results

3.4.1 Count-Preserving Hierarchical Evaluation

In order to investigate the *Count-Preserving Hierarchical Evaluation* metric on a task and contrast it with baseline leaf-only evaluation and set-based evaluation of Kosmopoulos et al. (2015), experiments were conducted on the ICD-9 coding task in MIMIC-III. The CAML (Mullenbach et al., 2018), BIGRU-LWAN, (Chalkidis et al., 2019a), and HLAN (Dong et al., 2021a) models were trained and evaluated on the dataset setup (pre-processing and split) of Mullenbach et al. (2018) using the *top50* codeset (comprising the 50 most frequent labels).

	Standard Flat			Set-Based			CoPHE		
Model	P	R	F_1	P	R	F_1	P	R	F_1
CAML	59.3	61.4	60.1	62.6	65.5	64.2	61.1	65.4	63.1
BGLWAN	68.4	57.6	62.5	71.8	61.2	66.0	70.5	61.4	65.1
HLAN	73.9	57.4	64.2	77.0	60.3	68.5	76.1	59.1	66.5

Table 3.2: A comparison between flat evaluation and hierarchical evaluation – Set-Based and CoPHE – on three models from prior art using the *top50* codeset. The results are micro-averaged across labels (and ancestors for hierarchical measures). Levels of hierarchy up to and including the lowest chapter level (e_2, e_1, e_0, c) are considered. Hierarchical measures report higher F_1 scores than the original flat measures. Furthermore, Set-Based evaluation F_1 scores are higher than those of CoPHE. Note that the hierarchical metrics are not directly comparable. but when used in tandem they can be utilised in analysis of tendency to over-/under-predict.

CAML was chosen as the first notable LMTC model developed on MIMIC-III, commonly used as a baseline in the field. BIGRU-LWAN (BGLWAN) was an alternative to CAML swapping the CNN encoder with a bidirectional GRU and utilising the zero-shot-motivated label description embeddings proposed by Rios and Kavuluru (2018b). This model's performance was originally presented against different LMTC models (including CAML) on three datasets (including MIMIC-III). Finally, HLAN was included due to its inclusion of label correlation encoding pre-trained from the training labelsets in order to capture the complex semantic relations between the labels (which may differ from the relations present within the ontology). The results for the common leaf-only (Standard Flat), hierarchical evaluation proposed by Kosmopoulos et al. (2015) (Set-Based) and CoPHE (Falis et al., 2021) on the test set for the *top50* codeset averaged across 10 training runs are presented in Table 3.2. Note that these metrics are not directly comparable – rather they are meant to be used in tandem. By observing the set-based evaluation values and comparing them to CoPHE one can learn about the model's tendency to over-/under-predict. This does not invalidate standard leaf-only evaluation measures (which track exact-match performance), nor does it mean one or the other hierarchical evaluation metric is superior – the analysis of their results side-by-side can enhance the understanding of a model's errors.

Firstly, the scores for both hierarchical metrics are higher than the flat leaf-only evaluation approach. This shows that the metrics are more lenient than the baseline,

which does not involve the hierarchy within the evaluation. It should be noted that this does not mean hierarchical metrics should replace the baseline metrics (as they capture different information), only that they exhibit behaviours expected given their design.

The scores for CoPHE are consistently lower than for the set-based counterparts, both for P and R (and consequently for F_1). This implies that all of the presented models suffer both from instances of overprediction and underprediction visible thanks to the count-preserving nature of CoPHE (which is absent in the set-based hierarchical evalation). This may stem from the fact that, while these models differ in their encoders, attention mechanisms, and semantic similarity of labels, they have a fairly similar approach to generating outputs – a layer of $|L|$ sigmoid units corresponding to individual labels. These results serve to illustrate the use of the metric. Results in Chapters 4 and 5 are further enhanced with CoPHE.

3.4.2 Weak Hierarchical Confusion Matrices

The development of Weak Hierarchical Confusion Matrices was conducted alongside the work on rule-based data augmentation (Chapter 4), where it was involved in the comparison of models trained on datasets enriched with the developed augmentation techniques against one trained on the baseline MIMIC-III dataset. Within the context of this chapter, we present the results on a baseline-dataset trained CAML considering the entire codeset (Table 3.3) along with a WHCM for a family of codes (*427: Cardiac dysrhythmias*[2]) (Figure 3.13) with the corresponding analysis of error types given different gold standard codes (Table 3.4). Similar statistics can be drawn for this matrix for performance given a predicted code by running the same statistics on the transposed confusion matrix – *i.e.*, rather than considering how likely a prediction is correct or mispredicted within or outside of the family given an observed gold standard label, the same statistics applied to a transposed confusion matrix provide the likelihood of correctness of the prediction or the type of misprediction given an observed predicted code.

The overall results on CAML using the *full* codeset (Table 3.3) provide an expanded understanding on the macro-level in the setting predicated on the gold standard codes – especially the WHCM-derived proportions of predictions which are correct

[2]This code family was chosen only as an illustrative example, as it includes a number of unique codes sufficiently large to illustrate confusion between multiple labels, yet sufficiently small to be easy to digest for the reader. Furthermore the family illustrates the common case of a code family having one code that tends to be assigned often (and tends to be correctly predicted) while other codes having smaller populations (some considerably so).

(Mac-Cor, corresponding to Assumption 1 in WHCM construction), mispredictions (confusions) within the same family (Mac-Conf, Assumption 2), and out-of-family errors (Mac-OOF, Assumption 3). On average 90% of the entries within the individual WHCMs for all the code families used within MIMIC-III are OOFs. This is concerning, as it indicates underprediction, which may stem from the model providing the incorrect (lower) number of predictions from a family, or simply from failing to identify the concepts to be relating to the code family (which would result in within-family confusion rather than OOF).

This is further supported by the example results for the code family *427*. Figure 3.13 shows that, while the model experienced some over-prediction, it suffered far more from under-prediction (low-recall). A high number OOF predictions – both absolute (170) and relative to the number of instances of the true label (80.19%) – occur for the gold standard label *427.89: Other specified cardiac dysrhythmias* – an example of a problematic "umbrella" code (ICD9Data.com[3] lists 92 approximate synonymous phrases for the code).

Table 3.4 presents statistics for the confusion matrix presented in Figure 3.13 predicated on gold standard codes. There are 14 codes of this family present within MIMIC-III, with 12 appearing in the test set. Six of them have been correctly predicted at least once during the evaluation on the test set. Two of them (*427.31*, *427.32*), are more likely to be predicted correctly than being confused within the family, or underpredicted (OOF). Four (*427.89*, *427.5*, *427.41*, *427.1*) are predicted correctly at least once, but mostly suffer from underprediction (OOF). Six (*427.0*, *427.69*, *427.9*, *427.81*, *427.61*, *427.2*) are never predicted correctly. This matrix visualisation within the scope of a family was presented here to facilitate family-level analysis – something that can be applied to other families of interest. However, results in Chapters 4 and 5 further enhanced with WHCM-based evaluation will focus on the scope of the codeset – similar to Table 3.3).

[3] http://www.icd9data.com/2014/Volume1/390-459/420-429/427/427.89.htm

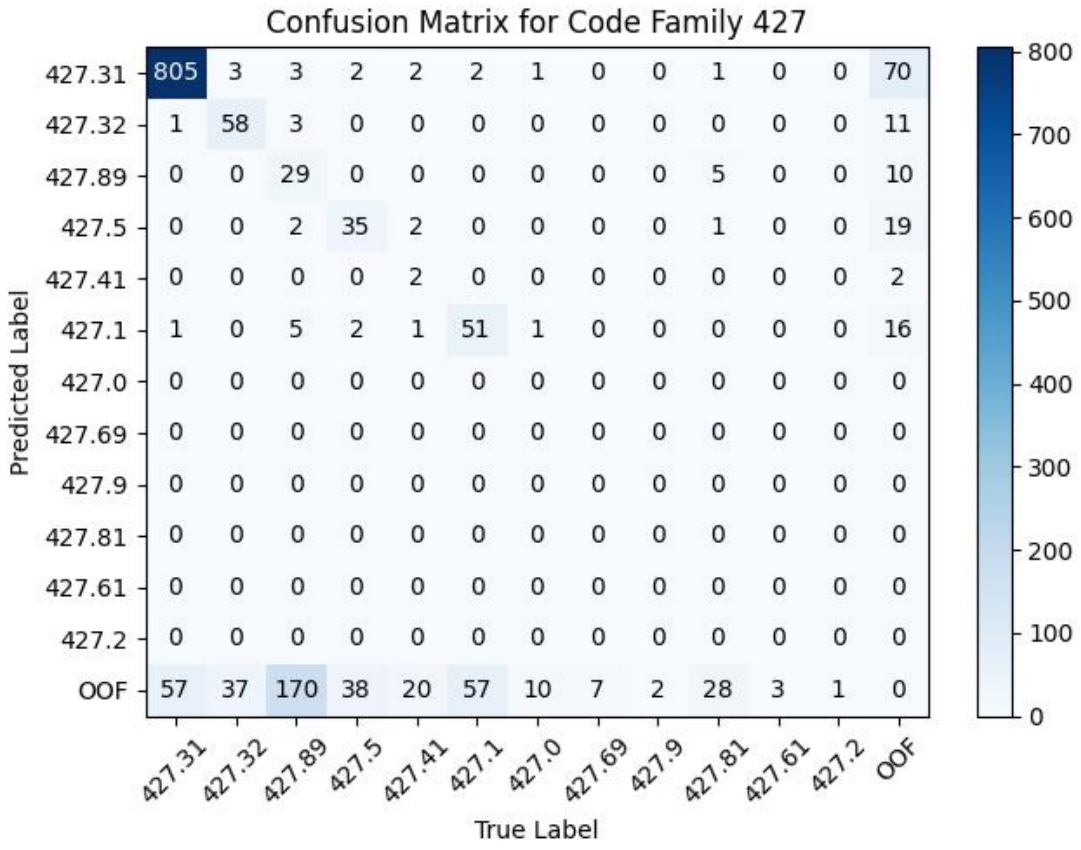

Figure 3.13: An example of a WHCM for the code family *427: Cardiac dysrhythmias*. Codes that are predicted, are mostly predicted correctly (high-precision). Codes *427.31: Atrial fibrillation* and *427.1: Paroxysmal ventricular tachycardia* get their predictions liked to a variety of their siblings. Code *427.81: Sinoatrial node dysfunction* is never predicted correctly (or at all), but was confused with three of its siblings.

Mic-F1	Mic-F1$_H$	Mac-Cor	Mac-Conf	Mac-OOF	Match
0.441	0.487	0.043	0.055	0.902	0.044

Table 3.3: Test-set performance of a CAML model trained on MIMIC-III with a focus on WHCM-derived macro-averaged metrics – Mac-Cor corresponds to the proporiton of predictions which are correct (corresponding to Assumption 1 in WHCM construction), Mac-Conf to proportions of mispredictions (confusions) within the same family (Assumption 2), and Mac-OOF to the proportion of out-of-family errors (Assumption 3). Match on the level of an individual label is a binary metric corresponding to whether the most common prediction given that label is the label itself or a different label – either within-family error or OOF. The value for Match in this table is also macro-averaged. Mic-F1 and Mic-F1$_H$ correspond to standard micro-averaged leaf-only F_1-score and micro-averaged CoPHE F_1 respectively.

3.5 Discussion

This chapter proposed the use of hierarchical evaluation measures in the LMTC task involving hierarchical label spaces and applied two novel evaluation approaches (CoPHE, and WHCM) to the task of automated ICD-9 coding of discharge summaries. Unlike the approaches in prior art of LMTC generally, and ICD coding specifically, which treat all mispredictions equally, CoPHE adjusts the penalty for misprediction based on the positioning of the prediction and the gold standard within the hierarchical label space through propagating the counts of leaf-level codes present in the prediction and gold standard sets to their ancestor nodes within the hierarchy. This allows partial credit in mispredictions. When compared against the performance of the similarly-motivated set-based hierarchical evaluation proposed by Kosmopoulos et al. (2015), the propagation of counts (natural numbers) within CoPHE enables tracking over- and under-prediction. The experiments conducted on some of the standard prior art models on the MIMIC-III dataset show a tendency of lower CoPHE scores compared to (binary) set-based hierarchical evaluation, suggesting the presence of over- and under-prediction errors. We speculate that this may be due to the design of the output layer of the models (which these models have in common) not incorporating the hierarchical nature of the label space or its implicit constraints. Structure of the label space has previously been incorporated into model architectures either in the design of model layers mimicking different levels of the ontology (*e.g.*, Falis et al. (2019)) or through modelling the ontology through graph-convolutional layers (*e.g.*, Rios and Kavuluru

Expected Code	Identity #	Identity %	Preferred Code	Preferred #	Preferred%	OOF %	In-Family-Confusion %	Match
427.0	0	0	OOF	10	83.3	83.3	16.7	FALSE
427.1	51	46.4	OOF	57	51.8	51.8	1.8	FALSE
427.2	0	0	OOF	1	100	100	0	FALSE
427.31	805	93.2	427.31	805	93.2	6.6	0.2	TRUE
427.32	58	59.2	427.32	58	59.2	37.8	3	TRUE
427.41	2	7.4	OOF	20	74.1	74.1	18.5	FALSE
427.5	35	45.5	OOF	38	49.4	49.4	5.1	FALSE
427.61	0	0	OOF	3	100	100	0	FALSE
427.69	0	0	OOF	7	100	100	0	FALSE
427.81	0	0	OOF	28	80	80	20	FALSE
427.89	29	13.7	OOF	170	80.2	80.2	6.1	FALSE
427.9	0	0	OOF	2	100	100	0	FALSE

Table 3.4: An example of the output of the WHCM analysis tool for the code family *427: Cardiac dysrhythmias* (corresponding to the Figure 3.13).

(2018b)). The issue of over and under-prediction could also be further resolved either through the model architecture (*e.g.*, through modelling families which allow a maximum of one code with a two-step process of first predicting the presence of a family and then, if the family is predicted to be present, applying a softmax on the family's leaf codes) or in post-processing (*e.g.*, further filtering the existing sigmoid predictions to allow only a certain number of positives per family – the ones with the highest sigmoid output – where the number of codes allowed per family could be determined either by expert knowledge or derived from training data). The proposed evaluation metrics could then be used to confirm whether such approaches to representing the code hierarchy and enforcing limits on predictions per family are beneficial to model performance.

WHCM results point to a major issue with most false negatives coming from underprediction with regard to families, rather than within-family confusion. The labels which seem to suffer the most from this within the sample family of *427* are either lower-resource, or umbrella labels (*e.g.*, the case of *427.89: Other specified cardiac dysrhythmias*). In the case of the umbrella label, interesting behaviour is observed, where the errors come predominantly from the OOF scenario and there is relatively little within-family confusion present compared to other non-umbrella labels with populations of at least 100 within the test set (*427.31*, *427.1*). The label performing the best is the one with the highest population within the family (*427.31: Atrial fibrillation*). This analysis was made possible through the visualisation of the confusion matrix.

These findings suggest that the OOF issue can be approached through providing more training examples. As real ones are difficult to acquire, it is reasonable to consider approaching the issue with data augmentation methods – both to provide more examples of relevant terms for a code in a variety of clinical scenarios, and also to cover a range of surface forms in which a code may be presenting (different concepts fitting under an umbrella, or synonyms for non-umbrella codes).

3.6 Conclusion and Future Work

This chapter focused on addressing shortcomings in existing standard evaluation metrics in LMTC by integrating the ontological structure into the evaluation and analysis of LMTC models within the task of ICD coding. The developed methods when applied to the task of assigning ICD codes to MIMIC-III discharge summaries point to a major issue of under-/over-prediction in prior art models. The difference between per-

formance on the big head and long tail of the label space has been previously observed and previous work proposed approaches to mitigate the issue. Future work should consider where the issues highlighted within this chapter may be coming from and how they may be addressed. Given the similarity of the output layers within the examined models, one avenue of model-side research to address the issues may be to integrate the rules of the ontology into the output layers, *e.g.*, via neuro-symbolic learning (Ahmed et al., 2022). The dominant long tail, on the other hand, suggests approaching the problem from the data side, *e.g.*, through data augmentation.

Chapter 4

Rule-Based Data Augmentation

4.1 Introduction

Large-Scale Multi-Label Text Classification (LMTC) tasks, such as automated ICD coding of discharge summaries, suffer from a big-head long-tail distribution of classes (as shown in Section 2.3). This phenomenon naturally arises due to some labels being more frequent than others – patients are far more likely to suffer from *401.9: Essential hypertension, unspecified* or *250.00: Diabetes mellitus without mention of complication, type II or unspecified type, not stated as uncontrolled* than *357.0: Acute infective polyneuritis* (also known as *Guillain-Barré syndrome*). Rare diseases even have a dedicated ontology – *Orphanet* (Weinreich et al., 2008) (for a brief introduction of Orphanet, please refer to Chapter 2 Section 2.2.3). Distribution of concepts can further be affected by the source of the data. In the case of clinical *Natural Language Processing* (NLP), dasesets often come from a single institution whose location or specialisation will be reflected by the frequency of relevant conditions and procedures within the dataset. For instance, hospitals in Switzerland are unlikely to have cases of injuries caused by shark bites (*e.g.*, *W56.41XA: Bitten by shark, initial encounter* in ICD-10). Hence, depending on the data source, some labels will have very low frequency within the training data – *few-shot* (FS), or be completely absent – *zero-shot* (ZS) scenario. Furthermore, adding new labels into a code standard (such as the ICD-9) with a new edition through splitting/fusing/altering existing concepts, or introducing new concepts also creates a ZS scenario – the new label not having been assigned to existing training data and hence being absent from it. Medical coding methods need to perform well regardless of the frequency of concepts and hence need to be able to adapt to these low-resource scenarios.

This chapter proposes a novel data synthesis method for ICD coding applied to ICD-9 data in MIMIC-III. This method is applicable to medical LMTC tasks using medical terminologies. It combines the output of *Named Entity Recognition and Linking* (NER+L) engines with access to medical vocabularies in order to replace mentions of underspecified codes (*e.g.*, *365.9: Unspecified glaucoma*) with more specific (and often less frequent) alternatives from the same family of codes (*e.g.*, *365.41: Glaucoma associated with chamber angle anomalies*). This represents novelty compared to utilising synonym replacement (which is also included in the data augmentation method produced alongside the data synthesis) and replacement of specific concepts with less specific ones (*hyponyms*) in previous work. While data augmentation in machine learning is understood as any adjustment to an existing datapoint in order to create a new one, within this chapter the term is used specifically to describe such adjustments to an existing datapoint (the model input, *e.g.*, text) that result in the creation of a new datapoint without making changes to its associated (gold standard) labels. A situation where the adjustments of the existing datapoint also extend to altering the associated labels (forming a silver standard), while in the context of machine learning is also a form of data augmentation, is referred to within this chapter as data synthesis for easier distinguishing between techniques.

Rule-based NER+L methods, assuming machine learning is not used, are not directly affected by the frequency of relevant concepts and labels within a training dataset. Rather than training a model to tackle the task, handcrafted rules (often designed with the involvement of domain experts) are employed. There either is a suitable rule designed for a given situation, or not. If a new code is introduced into the label space or an existing code is modified to cover different situation or surface forms, the rules need to be adjusted to reflect this. Savova et al. (2010)'s Apache cTAKES is an example of a notable clinical information extraction system which utilises rule-based components alongside machine learning.

Machine learning methods, and most notably neural learning approaches, are data-driven. The populations of labels available during training and the variety of the surface forms of the corresponding concepts within the associated inputs affect a model's generalisability (Maharana et al., 2022). This effect is especially pronounced if the model is not designed with the FS/ZS scenario in mind. Previous work has tried to address data sparsity issues on the model side by setting non-trainable parameters within networks as representations of ICD-9 codes enriched with knowledge from the ontologies (Rios and Kavuluru, 2018b). While the FS/ZS performance improved, the overall

performance deteriorated.

An alternative to making model adjustments is to avoid the FS/ZS scenarios by supplying more data. However, data acquisition is problematic. Not only are there relatively few corpora of clinical text available, but these are not necessarily labelled using the same label space, or may come from the same/similar institution making them unlikely to enrich the base dataset with previously missing rare labels. For instance, MIMIC-III and MIMIC-IV come from the same institution – Beth Israel Deaconess Medical Center in Boston MA. However, while MIMIC-III is labelled with ICD-9, MIMIC-IV has subsets labelled with either ICD-9 or ICD-10. While mapping exists between the ontologies, it is not one-to-one (*e.g.*, laterality – indicating if the condition is occuring on the left side, right side or both – accounted for in ICD-10 codes is not present in ICD-9), meaning parts of some concepts would be lost in translation of the labels.

If real data cannot be acquired, it can be enriched through *Data Augmentation* (DA) or synthesis. DA in the context of machine learning comprises methods for increasing the amount and variety of training data. This can be achieved either through modifications to already available real data (in the context of image processing *e.g.*, image rotation or flipping as reported by Maharana et al. (2022)) or producing new data from a generative model in conjuction with an input – be it random noise in GANs (Antoniou et al., 2017) or inputs based on existing relevant examples (*e.g.*, with the use of a stable diffusion model, such as in Trabucco et al. (2023)).

The focus of this chapter is on rule-based data augmentation where, rather than completely new datapoints, a method amends an existing datapoint to create a new one. Section 4.2 describes the background on data augmentation and synthesis techniques in general-domain and clinical NLP. Section 4.3 presents the proposed rule-based data augmentation and synthesis pipelines and the experiment conducted for the analysis of the methods' effect on model performance. Section 4.4 describes the experimental results and Section 4.5 interprets the results as well as presents conclusions. Section 4.6 discusses the shortcomings of the method and suggests directions for future work. This work has been published as part of Falis et al. (2022) with myself as the lead and first author with supervision and guidance from my co-authors. Doctor Hang Dong contributed with continuous discussion of the ideas and provided previously attained outputs of the SemEHR system on the relevant MIMIC-III data. The text of the publication has undergone minimal changes with regard to methods and conclusions based on feedback from the supervisory team. Further training runs of a pre-existing

experiment were conducted on three of the larger datasets in order to present mean performance across 5 runs (4.3). The results for these larger datasets were reported in Falis et al. (2022) on a single run each due to time constraints. All of the other (smaller) experiments were originally reported across 5 runs and their results were hence not adjusted. An expanded introduction and background sections were written for the thesis.

4.2 Background

4.2.1 Data Augmentation in Natural Language Processing

Li et al. (2022) provide a review of data augmentation methods in NLP categorising the approaches based on the diversity of the resulting augmented data. They note that *paraphrasing* methods provide relatively little improvement to data diversity compared to *noising*-based methods, for instance random insertion or removal of tokens (*e.g.*, "suffering from a heart attack" becoming "suffering from a attack" with the token "heart" being randomly removed), which involve more changes. The most diverse data is created with *sampling*-based methods, assuming that a sufficiently faithful representation of the training data is learned to generate novel examples.

Paraphrasing can be done on different levels – token, phrase, or sentence. Token-level (and some phrase-level) can be achieved with the use of a thesaurus, such as WordNet[1]. *Easy Data Augmentation Techniques* (EDA) is a widely used paraphrasing data augmentation technique replacing words with their synonyms based on WordNet. The *n* candidate tokens to be replaced are chosen randomly from non-stopwords within the input text. These are then replaced with random synonyms. A similar technique has been applied in extreme multi-label classification by Zhang et al. (2020a). Paraphrasing can also be achieved through semantic embeddings, where, rather than a curated thesaurus, a pre-trained word-embedding model, such as GloVe (Pennington et al., 2014), Word2Vec (Mikolov et al., 2013a) or FastText (Bojanowski et al., 2017) is utilised instead. Word embedding models are designed based on the distributional hypothesis of semantics – words that appear in similar contexts have similar semantic meanings (Almeida and Xexéo, 2019). These models have been shown to successfully encode semantic relationships of words within the embedding space Mikolov et al. (2013b). These could be synonyms, but also antonyms, hypernyms or hyponyms.

[1] https://wordnet.princeton.edu/

Furthermore, language models can be utilised for generating segments for suggesting possible replacement words and phrases, and generating sentences. Within the context of Li et al. (2022) these correspond to pre-trained language models (PLMs), such as BERT (Devlin et al., 2018). However, a *Large Language Model* (LLM) can be also utilised in such a capacity and will be further explored in Chapter 5. Semantic-embedding and language-model approaches themselves, however, rely on the availability of data and will reflect the training data issues/biases – *e.g.*, a language model or semantic embeddings based on medical texts from the United States will use American English spelling (*e.g.*, anemia instead of anaemia) or local preferred or branded versions of drug names (*e.g.*, Acetaminophen or Tylenol instead of Paracetamol).

Enhancing the vocabulary through a thesaurus or an ontology may positively affect the semantic embedding models as well. During evaluation, the embedding model may receive inputs with tokens not seen during pre-training – the *Out-of-Vocabulary* (OOV) scenario. Conceptually, OOV is similar to the zero-shot scenario – in both cases no relevant examples are observed during training while they can be encountered in evaluation. The difference between these two concepts lies in the model context – the OOV scenario is the extreme case of data sparsity in the unsupervised training of a word embedding model (on the level of input strings), while the zero-shot scenario presents the extreme case of data sparsity in a supervised task of classification, where classes unseen during training can appear within evaluation (Xian et al., 2017). Word embedding models commonly used in LMTC encoders tend to map OOV tokens to a special "unknown" token losing the uniqueness of the token. Such a token may be a keyword – such as in the case of rare diseases named after a person – *e.g.*, Munchausen's Syndrome (*301.51: Chronic factitious illness with physical symptoms* in ICD-9). Relevant terms – words and phrases associated with a given code differentiating it from similar codes (*e.g.*, its siblings within the code hierarchy) – unseen in the original data may be introduced through augmentation into the static vocabulary mitigating the issue.

Noising methods can be compared to salt-and-pepper (Maharana et al., 2022) (modifying an input image to include noise of randomly distributed white and black dots) in image processing. These are not informed through external resources like paraphrasing methods, but rather involve the manipulation of random (individual or groups of) tokens. These manipulations include token/phrase swapping, deletion, insertion, or substitution.

Sampling-based methods are task-specific and requiring task-specific information

(*e.g.*, labels) to model the data distribution for the task in order to produce new data for to it (Li et al., 2022). This can be achieved through rules and heuristics (*e.g.*, swapping the subject and object of a sentence), similar to paraphrasing methods. Machine learning models can be used for augmentation through sampling – *e.g.*, through back-translation (Sennrich et al., 2015a), where, rather than making adjustments to an existing input sentence, a model is first used to translate it into a different language and then back to the original rewriting the sentence in a generated way. Alternatively, unidirectional translation can be utilised for producing data in a target (lower-resource) language based on (higher-resource) source-language data (Li et al., 2022). Finally, samples can be created with the aid of pre-trained language models, *e.g.*, by fine-tuning on the training data in order to capture its distribution and generating new samples (Anaby-Tavor et al., 2020).

4.2.2 Data Augmentation in Clinical Natural Language Processing

Despite a large amount of clinical data being produced on a daily basis within hospital systems, clinical text available for building machine learning and deep learning models is scarce. This is due to the lack of availability of the data to the public, the privacy concerns of sharing medical-record data, the costs associated with the necessary de-identification of these documents in order to preserve the patients' privacy, and the high costs of expert annotation. In the absence of sufficient data, researchers turned to fine-tuning pre-trained models, such as BioBERT (Lee et al., 2020), which (due to their exposure to large amounts of data during pre-training) can be fine-tuned with significantly less data than a model trained from a random initial state. Alternatively, researchers have employed data augmentation techniques to produce further training data within the clinical domain.

Synonym replacement with WordNet has been previously employed by Ollagnier and Williams (2020) in medical document classification. Their method randomly replaces a set number of non-stopwords per document with their synonyms. The relatively unrestricted choice of words, however, means the synonym replacement may not be applied to concepts of high interest – medical vocabulary. However, as presented in Section 2.2, concepts and relations within the clinical and biomedical domains are captured within a variety of knowledge bases or ontologies. These lend themselves to be utilised similarly to WordNet in DA through paraphrasing.

Kumar et al. (2019) identify *Unified Medical Language System* (UMLS) concepts

within the data of a biomedical question answering task using MetaMap (Aronson, 2001) and produce new datapoints by replacing the identified string with a phrase of the format "<UMLS Canonical name>, a <UMLS Concept Type>" (*e.g.*, "T1D" would be augmented to "TYPE 1 DIABETES MELLITUS 1, a Disease or Syndrome"). UMLS-based synonym replacement guided by the output of MetaMap has also previously been used for DA in NER+L and sentence classification by Kang et al. (2021) alongside noise-based augmentation through random insertion, random swap, and random deletion. While MetaMap is closely tied to the UMLS project, other more recent NER+L methods for retrieving UMLS concepts exist – *e.g.*, SemEHR (Wu et al., 2018), and MedCAT (Kraljevic et al., 2019) (introduced in Section 2.4.2).

Schrempf et al. (2021) developed a document synthesis method through the use of templates in brain radiology reports. These templates are used for augmenting concepts of interest, or replacing them with synonyms based on the UMLS. This eliminates the need for a NER+L system within the method, as the relevant spans are already identified through the fields within the template. Furthermore, transformation of generic UMLS concepts into more specific ones is utilised – *e.g.*, the concept *Tumour* yielded potential replacement surface forms of *intracranial glioma* and *brain meningioma*. Within the context of Schrempf et al. (2021)'s study such replacement with a hyponym is utilised for introducing more variety to a generic label (*i.e.*, "brain meningioma" was used as a surface form for the generic "Tumour" class, rather than its own more specific class).

The common theme among these approaches is the two-step pipeline of first selecting candidate spans in training data for replacement (be it through identification via NER+L, or producing them prescriptively via document templates) and adjusting the words or phrases within these relevant spans through replacement with similar ones based on an ontology, such as the UMLS. None of them explore utilising such synonym or hyponym replacement for synthesising data specifically for zero-shot labels through specifying the unspecified, which was pursued in within the thesis. The following section presents a method for data augmentation through specifying unspecified concepts with the UMLS within context of large-scale document classification task of ICD-coding.

4.3 Method

This thesis employs UMLS-based paraphrasing DA for the LMTC task of ICD coding in the instance of ICD-9 coding in MIMIC-III. Similar to the methods of Kang et al. (2021) and Kumar et al. (2019), this method relies on identifying UMLS concepts and replacing them with surfaces forms coming from medical ontologies. Rather than MetaMap, more recent biomedical *NER+L* methods (SemEHR and MedCAT) are utilised for candidate selection. An existing Python package (PyMedTermino2[2]) is utilised for mappings between a variety of biomedical ontologies (SNOMED CT, UMLS, ICD) allowing the enhancment of the set of relevant surface forms, and manipulating only the discovered UMLS concepts relevant to the gold standard.

Further, the method employs a novel ontology-guided document synthesis, which converts relevant concepts into semantically adjacent concepts based on ICD-9, with the expected label set being adjusted accordingly forming a new silver standard. Similar to the approach of Schrempf et al. (2021), hyponymous surface forms are chosen to replace discovered surface forms relating to generic concepts appearing in the gold standard. The synthesis work within this chapter differs from the work of Schrempf et al. (2021) in that, rather than using hyponyms to represent high-level concepts, the surface forms related to a selected (source) code are replaced with surface forms associated with a target sibling code and the updated label is that of the target code. This increases the population of the more specific target label within the training data, rather than the already frequent generic label. The aim of this synthesis technique is to provide further training data specifically to FS and ZS labels.

The augmentation part of the method of Schrempf et al. (2021) is positioned between the synonym augmentation and synthesis method presented in this chapter. It utilises more specific related labels in synthesis, but uses the hyponyms in order to enrich the synonym set for the generic label, rather than produce data for more specific labels.

4.3.1 Augmentation Candidate Selection

The first step of the pipeline is the identification of relevant spans within the input text as candidates for augmentation. In order to determine these spans, first an NER+L system was applied on the training set. Two systems – SemEHR (Wu et al., 2018) and MedCAT (Kraljevic et al., 2019) – were explored to see if system choice (Med-

[2] https://owlready2.readthedocs.io/en/latest/pymedtermino2.html

CAT being more recent and with disambiguation capabilities) has major impact on performance of models trained on augmented data. The output of the NER+L engine produces entities present within the text along with the indices of their spans, and the UMLS *Concept Unique Identifier* (CUI) the system linked them to. These may not match with the gold standard document labels – this could stem from flaws within the NER+L system, or labelling error. Furthermore, the gold standard may not capture all mentioned concepts due local coding guidelines (*e.g.*, omission of certain codes). As the experiment was set up on a MIMIC-III[3] data split, the models would be evaluated on MIMIC-III data which would reflect the coding guidelines used during the development of the dataset. For this reason, despite knowledge of undercoding issues within the gold standard of MIMIC-III (Searle et al., 2020), the labels within the gold standard were considered superior to the output of the NER+L systems. In practice this meant that the outputs of the NER+L systems were checked against the document-level gold standard labels and NER+L outputs unsupported by the gold standard were not considered. The functioning of the augmentation candidate selection does not necessitate gold standard labels. This filer through available labels serves as a bridge between the human coding process at the source institution and the NER+L engines.

PyMedTermino[4] – a library of *Medical Terminologies for Python* enabling easy access to the most commonly recognised medical terminologies – was used to translate CUIs of the discovered concepts into ICD-9 in order to be compared with the gold standard. Unlike Searle et al. (2020) who sought to produce a silver standard by reconciling the output of NER+L methods with the gold standard, for the purposes of producing augmentation candidate within our method, a set intersection operation was applied between the NER+L outputs and the gold standard set. This resulted in a reduced set of NER+L outputs relating to the labels present within the gold standard for a given document. The candidate selection phase of the pipeline is presented in Figure 4.1.

4.3.2 Vocabulary Preparation

PyMedTermino was utilised as a knowledge base of synonymous surface forms for DA. PyMedTermino uses a locally stored version of the UMLS as its source knowledge base. For this research the 2021AA release of the UMLS[5] was employed. Four onto-

[3]MIMIC IV (Johnson et al., 2023) was not yet released at the time of the development of this method.
[4]`https://pythonhosted.org/PyMedTermino/`
[5]`https://www.nlm.nih.gov/pubs/techbull/mj21/mj21_umls_2021aa_release.html`

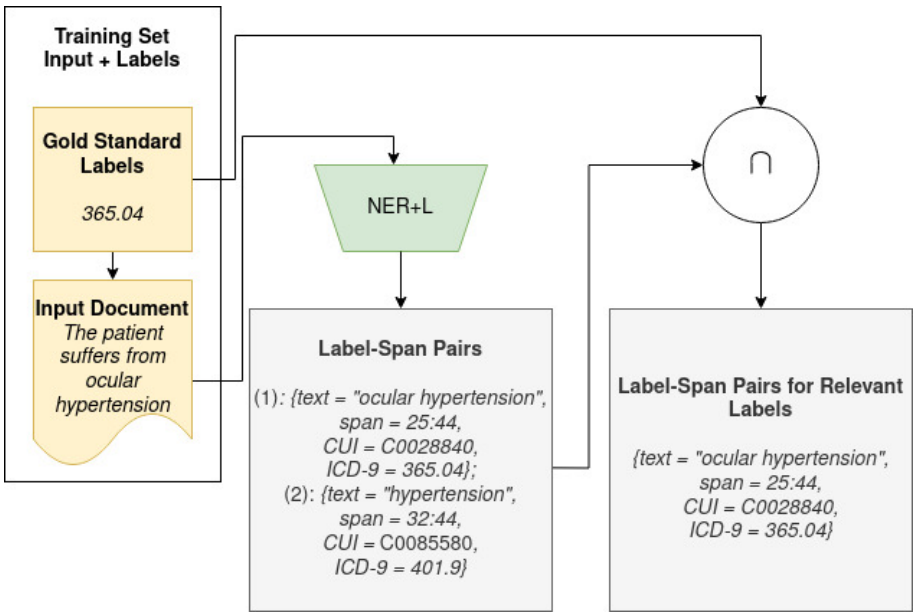

Figure 4.1: First phase of the augmentation/synthesis pipeline. Given a real data point from the training – free text in conjunction with gold standard labels – produces a list of span pairs returned by a Named Entity Recognition and Linking system and filters the ones not relating the gold standard labels.

logical databases were loaded through the library: 1. "CUI" (UMLS); 2. "ICD9CM" (ICD-9-CM); 3. "SNOMEDCT_US" (SNOMED CT); and 4. "ICD10" (ICD-10). While the SNOMED CT and ICD-10 label spaces were not directly used, links between the UMLS and the rest of the ontologies expanded the available synonyms for individual CUIs. It is important to note that the mapping between ICD-9 and the UMLS is not one-to-one – a simple example of this is the case of "other" umbrella codes within ICD-9 relating to several concepts. For each ICD-9 code present within MIMIC-III, a list of relevant surface forms was produced. Each list contains the textual labels of the relevant UMLS concepts, and its synonyms. The synonyms comprise the surface forms under the "synonyms" property of the concept within the PyMedTermino environment created.

As pointed out in Section 2.2, ICD-9 contains certain aetiological patterns – *e.g.*, ".8" being used for "other" or ".9" for unspecified. As "unspecified" is implied through omission of further specification in text, the token itself was considered obsolete within surface forms. Hence, synonyms whose lowercase version contained the token "unspecified" were removed from the synonym vocabulary. This was done to avoid producing unnatural phrasing (*e.g.*, it was desirable to represent the concept *365.9: Un-*

ICD-9 Code	Synonyms
707.9	Chronic ulcer
707.8	Chronic ulcer of other specified sites
707.25	Unstageable pressure ulcer|Nonstageable pressure ulcer
707.24	Pressure ulcer, stage IV|Pressure ulcer stage 4|Pressure injury stage 4
707.23	Pressure ulcer, stage III|Pressure ulcer stage 3|Pressure injury stage 3
707.22	Pressure ulcer, stage II|Pressure injury stage 2|Pressure ulcer stage 2
707.21	Pressure injury stage 1|Pressure ulcer stage 1|Pressure ulcer, stage I

Table 4.1: Example synonymous surface forms for ICD-9 codes of the 707 family. Rows correspond to different ICD-9 codes and surface forms that correspond with each code. Individual synonymous surface forms are separated by the "|" character (the choice of this delimiter was motivated by its absence in all concept descriptions and synonyms).

specified glaucoma as simply "glaucoma", rather than "unspecified glaucoma" or "glaucoma, unspecified"). Surface forms involving parentheses were also removed in favour of versions of the surface forms including all relevant tokens in non-parenthetical clauses. The resulting vocabulary was stored in a local file in order to free up the computational resources used by PyMedTermino, as the vocabulary corresponded to only a fraction of the loaded database. A sample from a generated synonym file is presented in Table 4.1. Note the different usage of negation prefixes (un- or non-stageable), Roman and Arabic numerals, the use of commas, and the absence of "unspecified" in the case of *707.9: Chronic ulcer of unspecified site*.

4.3.3 Synonymous Data Augmentation

The augmentation procedure uses the input document, the output of the NER+L filtered by the gold standard, and the vocabulary file. The spans within the input document identified by the NER+L engine are replaced with a random synonym, if available. The synonym replacement can be further controlled by a parameter, which sets the probability (between 0 and 1) of a relevant span being replaced with a synonym. This parameter is set by the user as a proxy to what proportion of the relevant spans per document is to be replaced. This enables mixing of the original surface forms with ontology-based ones within the document. The parameter was set to 1 within the experiment, meaning each of the retrieved mentions were replaced. The new augmented document is associated with the same gold standard labels and added to the training

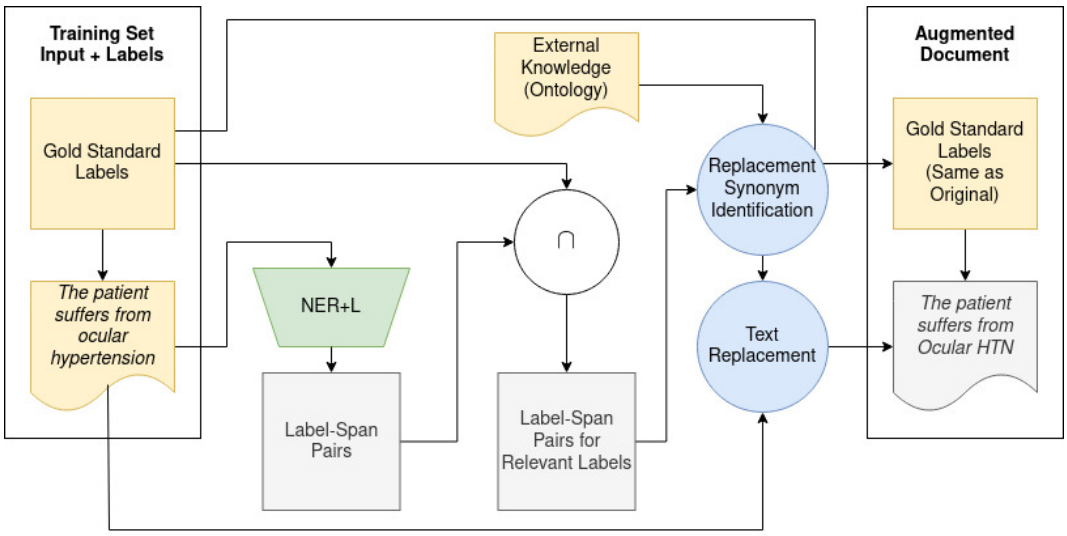

Figure 4.2: The full pipeline for data augmentation through synonym replacement.

set. The full pipeline for data augmentation through synonym replacement is presented in Figure 4.2.

4.3.4 Sibling Data Synthesis

In the synthesis procedure, further ontology-guided processes are introduced – identification of label spans relating to generic "unspecified" labels, and identification of labels within the same family which are more specific. Once an "unspecified" code with spans in the NER+L output is identified, a candidate replacement sibling is selected with low population siblings (constituting the FS or ZS scenario) being prioritised if available. This is performed on the labels on the document level, rather than for each mention. This means that for a given "unspecified" code, a single replacement sibling concept is chosen in order to replace the less specific version in multiple spans in the document (with possibly different synonyms of the replacement concept) – rather than each individual mention being replaced with a different random sibling concept. At least one mention of the "unspecified" concept within the document is then replaced with a surface form of this sibling concept following the same procedure of surface form replacement as in syonym-based data augmentation using the synonym vocabulary as described in Section 4.3.3.

A further difference between Sibling Data Synthesis and Data Augmentation is that synthesis includes an update to the labels associated with the document – the specific sibling label used in synthesis replaces the "unspecified" sibling in the set forming a

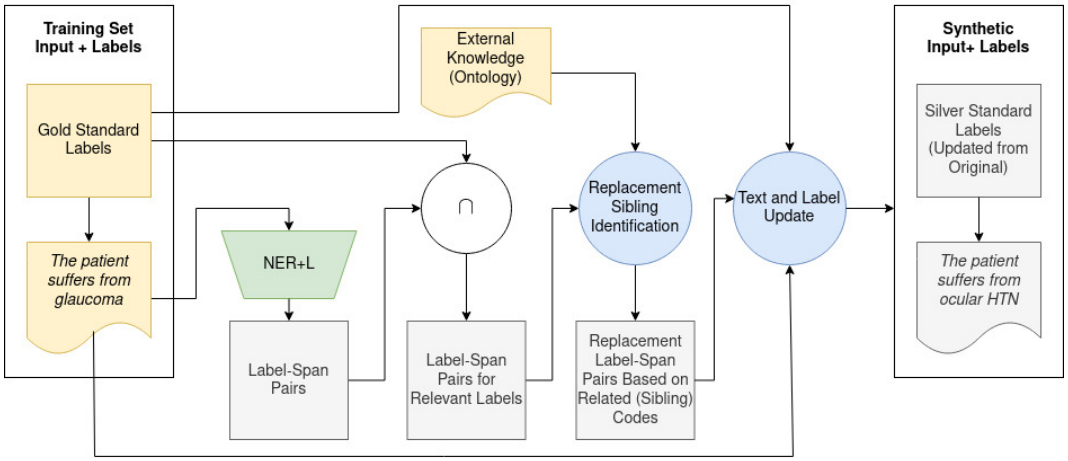

Figure 4.3: Rule-based document synthesis through replacing an "unspecified" concept with a more specific sibling.

silver standard. The new document associated with the silver standard is then added to the training set. The pipeline for the synthesis procedure is presented in Figure 4.3.

The focus on "unspecified" codes was motivated by the assumption, that an "unspecified" label means all its mentions within an input piece of text are non-specific, while a single specified mention warrants a more specific version of the code in the new silver standard. This mimics human behaviour where short-hands for concepts are acceptable, as long as the full surface form of a concept is presented within the text (*e.g.*, referring to "an intermittent angle-closure glaucoma" first and shortening it to "the glaucoma" later). This choice was further motivated by addressing potential imperfections in the NER+L predictions – replacing a specified code would require replacement of all its mentions, some of which may not be identified by the NER+L method leading to contamination of the data (*e.g.*, converting some – but not all – mentions of *401.0: Malignant essential hypertension* to *401.1: Benign essential hypertension* and having mentions of the patient suffering from both benign and malignant hypertension concurrently within the discharge summary – and hence creating an augmented document with contradictory information).

4.3.5 Datasets

The ICD-9-coded discharge summaries of MIMIC-III (Johnson et al., 2016) split according to Mullenbach et al. (2018) were used in order to create samples using the data augmentation and synthesis techniques described in Sections 4.3.3 and 4.3.4. The split was used to determine three population subsets for the labels: frequent (codes appear-

ing > 5 times in the training set), few-shot (present in the training set with population < 5), and zero-shot (absent from training set, but appearing in the validation or test set). These criteria for population subset membership are the same as those used by Rios and Kavuluru (2018b).

For data synthesis, the codes with "unspecified" or "not otherwise specified" in their description, and with "9" as the first or "0"/"1" as the second digit of the aetiology were selected. Of the total number of 8,692 unique labels within the training set 1,188 remained as viable "unspecified" codes (14.74% of the total code population within the training set).

Data synthesis and data augmentation were applied to the baseline dataset based on the outputs of the NER+L methods SemEHR and MedCAT. This resulted in the creation of four datasets – SemEHR-DA and MedCAT-DA for data augmentation (baseline enriched with data augmentation), and SemEHR-DS and MedCAT-DS for data synthesis (baseline enriched with data synthesis). A further pair of datasets merging the DA and DS were produced and named SemEHR-both and MedCAT-both. The inclusion of augmented and synthetic data resulted in the increase in population for a variety of labels making them less scarce and formally shifted away from FS and ZS. The resulting datasets, along with their sizes in the number of documents and the number of unique codes within the frequent, few-shot, and zero-shot sets are presented in Table 4.2. Furthermore a Baseline-like dataset of a similar size to the largest datasets – SemEHR-Both and MedCAT-Both – was created as a controlled experiment. This was done by concatenating two Baseline datasets (2xBaseline). Assuming a constant number of epochs, training on 2xBaseline corresponds to training on the Baseline for double the number of epochs.

4.3.6 Experiment

CAML models were trained based on the implementation of Chalkidis et al. (2019a) for 15 epochs on the created training sets. No few-shot/zero-shot model-side solution (such as the use of label embeddings as parameters) was applied. Each experiment used word embeddings of size 100 pre-trained on its respective training set according to the procedure proposed by Mullenbach et al. (2018). The development and test sets were the same across all experiments. The model weights with the best end-of-epoch validation F_1 score were evaluated on the test set. For each dataset 5 training runs were completed and the results were averaged. Note that Falis et al. (2022) reports results

Dataset	Size	Frequent	Few	Zero
Baseline	47,719	4,351	4,341	237
SemEHR-DA	66,559	4,818	3,874	237
MedCAT-DA	71,295	4,998	3,694	237
SemEHR-DS	74,851	5,167	3,538	224
MedCAT-DS	74,830	5,164	3,541	224
SemEHR-Both	93,690	5,446	3,259	224
MedCAT-Both	98,402	5,565	3,140	224

Table 4.2: Training set sizes (number of documents) and populations (number of unique codes) of the frequent ($5 > f$), few-shot ($5 \geq f \geq 1$), and zero-shot ($f = 0$) subsets based on the dataset's codeset (codes present in either the training, validation, and test sets) and the frequency in the training set (f).

for the larger three datasets (SemEHR-Both, MedCAT-Both, and 2xBaseline) only on a single training run due to time constraints. Additional experiment runs were conducted for these three datasets for the purposes of the thesis.

4.4 Results

The performance of the models trained on the baseline dataset and datasets enriched with the augmentation and synthesis techniques is presented in Table 4.3. The models trained on each dataset are compared using the following previously used metrics: Micro-F_1 for all codes, and R@10 (recall at k described in Section 3.3.1 with k set to 10 here) for few-shot and zero-shot codes. The codeset for few-shot and zero-shot codes is derived from the Baseline, and hence includes codes whose populations have increased in the DA, DS, and Both datasets. Furthermore, metrics introduced in Chapter 3 (CoPHE and WHCM) were employed to capture performance with respect to the ICD hierarchy. The micro-averaged results are enhanced according to CoPHE (Mic-F_{1H}). Furthermore, error-rates produced via WHCM macro-averaged across the codeset are presented – percentages of gold labels being predicted correctly (Mac-Cor); being confused with a code within the same family (Mac-Conf), and being confused as *Out of Family* (OOF) (Mac-OOF). Finally, the prediction most often matching with each gold standard code is tracked and compared against the label itself – whether a correct prediction is more likely than any kind of misprediction. For each code in the codeset this is reduced to a binary value (match or mismatch), and a macro average is

presented as the *Match* metric. The hierarchical evaluation approaches use the same representation of the ICD-9 tree originally developed for CoPHE.

It should further be noted that the baseline CAML results underperform with respect to the original results presented by Mullenbach et al. (2018). This is due to the use of fewer training epochs (Mullenbach et al. (2018) cease training after 10 epochs without significant improvement on $P@8$). All datasets enhanced with the proposed augmentation techniques result in improved standard and hierarchical Micro-F_1 compared to the baseline – at least 0.027 and 0.027 absolute (6.1% and 5.5% relative) for standard and hierarchical respectively (Mic-F1 and Mic-F1$_H$ in Table 4.3). Augmentation (DA) sets, while comparatively worse than Synthesis (DS) on R@10-Zero and standard and hierarchical Micro-F1, perform better on R@10-Few (corresponding to performance on the few-shot subset of the label space). The lower zero-shot performance (represented through R@10-Zero) in DA compared to DS can be explained by the fact that DA methods alter surface forms of labels already existing within the training set (contributing to codes within the frequent and few-shot subsets with already existing real text in the training data) while not introducing any previously unseen labels into it. On the other hand, DS methods introduce synthetic data for labels that were completely absent in the original training set leading to a greater potential for leading to a more pronounced relative impact on its target performance measure (R@10-zero for DS and R@10-few for DA respectively). Interestingly, SemEHR-DA performs on par with MedCAT-DA despite having a smaller training set.

The combination of DA and DS methods (Both) report the best F_1 results, with MedCAT-Both performing best in 5 out of the 8 reported metrics (including Mac-Cor, Mac-OOF and Match). On these metrics, both of these methods' results are at least as good as those of 2xBaseline, which is of a comparable size (each consisting of over 90,000 documents). The best R@10-Few performance was achieved by 2xBaseline, which corresponds to training the Baseline for twice as many epochs (while evaluating on the validation set the same number of times as the other models). While the improvement of DA and DS in R@10-Few and R@10-Zero implies our methods enhance these subsets, 2xBaseline dominating R@10-Few suggests that a better few-shot performance on the baseline can be achieved with more training epochs (as 2xBaseline effectively means double the exposure to the same training data as Baseline). Note that within Falis et al. (2022) where results for 2xBaseline, MedCAT-Both and SemEHR-Both were reported only on a single run per experiment, 2xBaseline also had the highest R@10-Zero score. Upon conducting 5 runs of the experiment for

each of these training sets with mean results reported, the average zero-shot performance for 2xBaseline is lower than both those of MedCAT-Both and SemEHR-Both and the smaller zero-shot focused MedCAT-DS and SemEHR-DS training sets. The best zero-shot performance was achieved on the synthesis-focused training sets, with SemEHR-DS reporting the highest performance. This shows that, while the additional exposure to the pre-existing original data – which seemed to have been sufficient to improve few-shot performance – including synthetic data for labels originally absent from the baseline training set leads to a consistent improvement in the zero-shot performance.

The difference between the standard and hierarchical (CoPHE) F_1 scores remained largely the same, which implies partial errors commonly seen in neural ICD-coding models (as described in Chapter 3 and reported in Falis et al. (2021)) were not addressed by these methods. This is further supported by the changes in Mac-OOF dominating compared to those of Mac-Conf. The lowest Mac-Conf was achieved by the original Baseline, but was coupled with a high OOF implying that this low confusion is mostly due to a higher proportion of codes not being predicted at all, rather than confused within the correct family of codes.

4.5 Discussion and Conclusion

It is important to further consider the augmentation strategy of specifying the unspecified proposed in the chapter to address the zero-shot scenario (in contrast, synonym-replacement is a fairly common means of data augmentation). While the replacement of concepts with their hypernyms (as done *e.g.*, in Schrempf et al. (2021)) is valid by the same logic as underpins the true-path rule (presence of a concept implies the presence of its more generic hypernymous ancestors), the same isn't necessarily true for the replacement of a concept with a hyponymous concept corresponding to more specific siblings of an "unspecified" concept. The resulting documents may not correspond to plausible clinical scenarios – the conditions or procedures corresponding to the more specific sibling codes may be unlikely to co-occur with the rest of the patients' information (conditions, procedures, family history, *etc.*) or may be in opposition with some of it (*e.g.*, test results). Furthermore, an "unspecified" label may in fact be an instance of undercoding by the human coder, where a more specific concept is in fact present in the input document, alongside unspecific mentions (leading to NER+L engines retrieving also unspecific mentions which would be matched with the

underspecified gold standard code). Such a scenario would also lead to the resulting document potentially containing conflicting information.

However, the aim of the augmentation strategy was to introduce keywords corresponding to these specific, yet rare concepts. This was achieved through the use of the unique descriptions associated with the codes. Even if the produced augmented document may not be valid from a clinical perspective, it exposes the model to keywords associated with the code (within the explored ontologies) during training, enabling the code's prediction during evaluation.

The data enrichment methods have improved on the baseline showing potential in approaching the few-shot/zero-shot scenario through data, rather than the model. This is especially true for the case of zero-shot performance on datasets built through replacing "unspecified" codes with more specific zero-shot siblings. While the data enrichment results are encouraging, further error analysis of LMTC models (future and already existing) is desirable. The WHCM results (Mac-Cor, Mac-Conf, Mac-OOf, and Match in Table 4.3) point to the fact that most false negatives are stemming from underprediction (Mac-OOF) of a family, rather than within-family confusion. Further error analysis should be conducted on false positives. The analysis from the WHCM tool can provide possible explanation of the errors of a model and may shed light on the design of more accurate models for LMTC.

4.6 Limitations

The approaches discussed in this chapter relied on the use of external NER+L tools, whose predictions are imperfect, and may not be available for all domains of interest. Other avenues of finding relevant entities, *e.g.*, the attention outputs of LMTC models, should be explored in future work.

While the use of the standard descriptions from the ontology introduces the concept's unique description into the text differentiating it from its "unspecified" sibling, this does not necessarily mimic real-world writing (*e.g.*, a real discharge summary may use an acronym or a different means of shortening in order to express a concept that corresponds to a code with a lengthy description in the ontology). The replacement of "unspecified" codes with their more specific siblings (and updating surface forms) also introduces issues of potentially creating unlikely or conflicting code combinations; and the possibility of replacing "unspecified" mentions of an undercoded concept wherein the document might in fact contain information pointing to a more specific concept,

but the coder – be it through guidelines or mistake – assigned a less specific code and the augmentation technique's result may be a document with conflicting mentions of specific concepts.

Furthermore, the methods presented within this chapter relied on making relatively small changes to pre-existing text. As such, these methods present relevant words/phrases, but only in the same contexts as the original surface forms they replace. Supporting information, such as test results, medication, relevant social/family history are not altered, which creates an issue of potentially presenting unrealistic or even conflicting information (*e.g.*, when an ICD code assignment depends on a test result). Each could technically be tackled with augmentation/synthesis methods involving specialised NER+L (*i.e.*, systems built for adjusting medication, and other supporting concepts). Such an extension would need a further linkage to knowledge bases and also a relation extraction component to identify which supporting information relates to a target disease/procedure concept.

An alternative to making changes to an existing narrative would be building a new narrative from the ground up. This is the subject of the next chapter.

Dataset	Mic-F1	Mic-F1$_H$	R@10-Few	R@10-Zero	Mac-Cor	Mac-Conf	Mac-OOF	Match
Baseline	0.441	0.487	0.034	0.035	0.043	**0.055**	0.902	0.044
2xBaseline	0.479	0.524	**0.09**	0.051	0.073	0.067	0.859	0.077
SemEHR-DA	0.469	0.514	0.055	0.034	0.062	0.062	0.876	0.063
MedCAT-DA	0.468	0.514	0.064	0.048	0.062	0.065	0.873	0.065
SemEHR-DS	0.471	0.518	0.051	**0.055**	0.067	0.065	0.869	0.069
MedCAT-DS	0.474	0.520	0.059	0.054	0.068	0.065	0.866	0.071
SemEHR-Both	**0.483**	0.529	0.064	0.052	0.077	0.068	0.855	0.082
MedCAT-Both	**0.483**	**0.53**	0.064	0.052	**0.079**	0.071	**0.85**	**0.083**

Table 4.3: Test-set performance (averaged across 5 runs) of CAML models trained on the original training set (Baseline) versus training sets with synonym augmentation (SemEHR-DA, MedCAT-DA), adjacent-code synthesis (SemEHR-DS, MedCAT-DS). Further experiments on larger datasets combining the data from augmentation and synthesis (SemEHR-Both and MedCAT-Both) are also compared against a duplicated baseline (2xBaseline with each input document of the original appearing twice to achieve a roughly similar size dataset). Best performance for each metric is marked bold. Zero and Few-shot codesets are based on the original Baseline training set. After each training epoch, the model is evaluated on the original baseline validation set. Results are reported on the original (baseline) test set.

Chapter 5

Synthetic Data Generation with Large Language Models

5.1 Introduction

The previous chapter focused on data augmentation with paraphrasing through string replacement. However, as presented in Section 4.2, there are alternative approaches to data augmentation in *Natural Language Processing* (NLP). This Chapter shall explore the use of a *Large Language Model* (LLM) in generating and coding clinical documents.

Data synthesis in ICD coding can also be approached through generating synthetic documents (unstructured data) based on structured inputs. Writing and coding discharge summaries requires extensive background knowledge – whether it be performed by human clinical experts or models. Recently, LLMs have displayed capability of using vast amounts of knowledge observed during training in generating cohesive text at a larger scale (Radford et al., 2019). It is, hence, of interest to investigate an LLM's capability of coding and generating discharge summaries. While using Application Programming Interfaces (APIs), such as the one for GPT-3.5, is problematic with real discharge summaries due to privacy issues, these models have the potential to aid in generating synthetic discharge summaries in order to supplement training local ICD coding models. An LLM capable of processing medical text can handle data sparsity by synthesising new data, either by augmenting existing documents or generating entirely new ones.

This chapter investigates the viability of GPT-3.5-generated medical documents for data augmentation in training local neural models and their credibility in clinical

settings. We investigate GPT-3.5 (within an ethical experimental setting for clinical note data) in the tasks of generating synthetic discharge summaries based on input lists of ICD-10 code descriptions assigned to real patients; its capability to code the generated data; and the ability to code real discharge summaries. The generated synthetic discharge summaries were also evaluated by clinical experts on the correctness, informativeness, authenticity and suitability within the clinical setting.

In automatic ICD Coding, discharge summaries serve as input, yielding codes from a specified version of the International Classification of Diseases (*e.g.*, ICD-10-CM)[1]. ICD coding faces distribution challenges mirroring other *Large-Scale Multi-Label Text Classification* (LMTC) tasks. Few common conditions (*e.g.*, hypertension), contrast with many underrepresented or absent in corpora, such as MIMIC-IV (Johnson et al., 2023). Moreover, limited real-world data availability, often restricted for privacy reasons, compounds these challenges. However, modern deep learning ICD coding approaches (*e.g.*, CAML (Mullenbach et al., 2018), HLAN (Dong et al., 2021a), RAC (Kim and Ganapathi, 2021)) are data-driven, and adversely affected by data sparsity unless explicitly designed to handle label under-representation. Techniques such as auxiliary information (Rios and Kavuluru, 2018b; Song et al., 2021; Ren et al., 2022; Wang et al., 2022), or data augmentation and synthesis (Falis et al., 2022; Kim et al., 2022b; Barros et al., 2022) attempt to mitigate these issues. ICD coding models with pre-trained encoders at best match the current state-of-the-art – usually involving domain-specific versions of BERT (Afkanpour et al., 2022). Large Language Models (LLMs), such as GPT-3 and its newer variants (Ouyang et al., 2022) (*e.g.*, GPT-3.5) or Large Language Model Meta AI (LLaMA). Touvron et al. (2023) have recently displayed state-of-the-art performance on several tasks with emerging capabilities (Zhao et al., 2023) and have become a new standard for advanced NLP tasks, especially ones relying on understanding natural language. These models retain and apply background knowledge observed during training. In the medical domain, notably Med-PaLM2 (Singhal et al., 2023a), was reported to perform on par with humans in multiple-choice medical school exams.

Recent studies have prompted discussions on GPT's utility in medicine, including applications in medical chatbots (Lee et al., 2023), or radiology (Lecler et al., 2023). Yeung et al. (2023) compared the ChatGPT with a clinical GPT model (Kraljevic et al., 2022) on generating patient vignettes. Previous research exists in generating data in low-resource settings in similar domains (*e.g.*, law, see Ghosh et al. (2023)).

[1] https://www.cdc.gov/nchs/icd/icd-10-cm.htm

This chapter is structured as follows. Section 5.2 describes a background on language modelling with a special focus on Large Language Models. Section 5.3 describes the selection of input lists of ICD codes used in augmentation, the prompt employed for generating the synthetic documents with GPT-3.5, the experiments used for investigating the model's capability to generate and code discharge summaries, and the evaluation approaches used in the experiments. Section 5.4 presents the results of the experiments their interpretation. Section 5.5 concludes the chapter with a summary of findings and a recommendation of directions for future work. Section 5.6 lists the limitations of this research.

This work has been published in Falis et al. (2024) with myself as the lead and first author with supervision and guidance from my co-authors. Aryo Pradipta Gema conducted evaluation of GPT-3.5's ICD-10 coding performance on MIMIC-IV data and summarised the qualitative assessment of the clinical evaluators. Both Aryo Pradipta Gema and Doctor Hang Dong participated in regular discussions of the project and helped develop the ideas presented within the publication. Doctor Luke Daines aided in the preparation of the text evaluation intended for clinicians, especially with the formatting. He also provided introductions to the other clinical staff involved in the evaluation. Doctors Luke Daines, Siddharth Basetti, Michael Holder, and Rose S Penfold utilised their experience in writing and coding discharge summaries in their evaluation of the synthetic discharge summaries on their suitability for the clinical setting. The generated discharge summaries are available as a dataset via PhysioNet[2].

5.2 Background

5.2.1 Language Modelling

Text generation, often formally referred to as *Natural Language Generation* (NLG), aims to produce plausible and readable natural language text (*e.g.*, a summary, answer, translation, or continuation of an excerpt) given input data (*e.g.*, a list of keywords, topics, or pre-existing text snippets to be built upon), as opposed to producing structured predictions (*e.g.*, a set of ICD codes relating to a document). As such, NLG is present in many classic examples of NLP tasks, such as machine translation (in encoder-decoder sequence-to-sequence models with attention mechanisms – as proposed by Bahdanau et al. (2014) – where text in the source language is encoded into a hidden state of

[2] https://physionet.org/content/generated-codes-low-resource/1.0.0/

the model and subsequently decoded into the target language) or text summarisation (abstractive – where a summary is generated, rather than extracted from existing text).

5.2.1.1 Neural Language Models

Neural networks proved to be relatively successful text generators (*e.g.*, in machine translation as shown by Bahdanau et al. (2014)) even before the advent of modern LLMs. A major issue in training generative neural models is the necessity of a large amount of training data. This is more pronounced with the increased number of trainable parameters – a complex model may learn to overfit the limited training data and generalise poorly.

5.2.1.2 Pre-trained Language Models

The need for extensive training datasets has partially been addressed with the advent of the *Pre-trained Language Model* (PLM), such as BERT (Devlin et al., 2018) or RoBERTa (Liu et al., 2019) based on the transformer architecture (Vaswani et al., 2017). These language models are pre-trained using generic language modelling tasks on large datasets. Through exposure to a large amount of training data, these large models encode the observed domain knowledge, which can then be utilised in the model's predictions. The pre-trained model weights can be used as an initial state for the model for fine-tuning for a downstream task – an instance of the transfer learning paradigm (Weiss et al., 2016). Commonly the parameters up to the output layer are set as non-trainable acting as an encoder model. The original output layer is replaced with a task-specific decoder layer/network, and the combined architecture is then fine-tuned on the comparatively lower-resource target task. While originally developed on general-domain datasets, such as Wikipedia articles, domain-specific PLMs, such as BioBERT (Lee et al., 2020), ClincalBERT (Alsentzer et al., 2019), or PubMedBERT (Gu et al., 2021) were introduced in order to enrich the model with domain-specific knowledge.

Unlike standard word embedding models, PLMs operate on tokenisers that allow subword units – with the Byte-Pair Encoding algorithm being utilised to iteratively build the vocabulary of tokens (Sennrich et al., 2015b). This algorithm builds the vocabulary from individual characters based on the pre-training data – common prefixes, suffixes or whole words/phrases are added into the vocabulary. Upon encountering an input, a tokeniser equipped with such a vocabulary can translate it into tokens based on

the highest available level of representation within the vocabulary – a full word can be matched (if very common) or be split into subwords. In the worst case the word can be processed as the individual sequence of character tokens, which means that, in theory, there are no unknown words, merely ones with better or worse representation. The embeddings within BERT are contextual – rather than considering each input word in isolation as a semantic feature, it considers the context words as well. This allows for multiple different representation of the same surface form – *e.g.*, homographic words, whose meaning differs given context, can be more accurately represented in this approach (*e.g.*, the string "HR" meaning "hour" or "heart rate" in different contexts).

5.2.1.3 Large Language Models

In employing PLMs, researchers discovered that their performance on downstream tasks tends to scale with the size of the model and the amount of available pre-training data. This led to the progressive exploration of larger numbers of trainable parameters leading to larger PLMs – *e.g.*, GPT-3 (Brown et al., 2020) and PaLM (Singhal et al., 2023a). Rather than merely leading to incremental improvement on performance, these models displayed surprising capabilities (referred to as *emergent abilities* (Wei et al., 2022)) in solving complex NLP tasks. A notable example is the ability to perform tasks based on a prompt including instructions or examples of input-output pairs (rather than being fine-tuned for the task on a dataset) – a phenomenon referred to as *in-context learning* (Dong et al., 2022b). To differentiate models with emergent abilities from the previous generation of PLMs that did not display them, these models have been named *Large Language Models* (LLMs). Interestingly, the basic concept of an LLM does not differ in model architecture or training method from a PLM. The difference is in scale – the amount of available training data, the number of trainable parameters, and the computational resources used in training the model. As such, model development in this paradigm further blurred the line between research and engineering within machine learning and natural language processing. LLMs became known to the general population with the release of the ChatGPT chat interface using the GPT-3.5 model as its backend. Its release has had major societal and ethical implications – *e.g.*, in education (Lo, 2023).

LLMs commonly follow one of two paradigms – decoder-only or encoder-only. In decoder-only models (*e.g.*, the various iterations of GPT) the decoding of the continuous representation of the input is autoregressive – the self-attention mechanism is masked for future position ensuring that the predictions for a current position can only

depend on known (already generated) outputs. In an encoder-only LLM a decoder is present, but the self-attention mechanism employed is not masked granting full access when decoding regardless of the position of the currently decoded state. BERT is an example of an encoder-only architecture. Since the release of ChatGPT, there have been several releases of LLMs, mostly by large corporations with sufficient resources to train such models. Some, such as LLaMA2 (Touvron et al., 2023), are publicly available – the model weights can be downloaded and studied. Others, *e.g.*, OpenAI's GPT-3.5 or Google's BARD, while usable through APIs, do not at the time of writing have their parameters available to the public.

5.2.2 Large Language Models in Clinical Natural Language Processing

Agrawal et al. (2022) explored the performance of GPT-3 on a variety of token-level classification tasks in the clinical domain, including clinical sense disambiguation, coreference resolution, and the extraction of relevant information, such as biomedical evidence, medical status, or attributes of medication. Their experiments included instructing the model to produce structured output, bridging the gap between the free text output of generative models and structured output layers of traditional single-task models. The authors report that GPT-3 (a general-domain LLM) outperformed previous solutions in the few-shot and zero-shot scenario suggesting that even general-domain LLMs can be utilised in the medical domain for low-resource scenarios.

Similar to the case of ClinicalBERT or BioBERT, domain-specific biomedical and clinical-domain LLMs were developed – *e.g.*, Med-PaLM (Singhal et al., 2023a), Med-PaLM2 (Singhal et al., 2023b), and GatorTRON (Yang et al., 2022). Med-PaLM and its follow-up project Med-PaLM2 have displayed high performance on medical question answering tasks. Med-PaLM surpassed the previous state of the art on MedQA (US Medical Licensing Exam-style questions) by over 17% with an accuracy of 67.6%, which underperforms with regard to human experts. Med-PaLM2 achieved an accuracy of 86.5% on the same task, benefiting from a stronger base model (PaLM2), further in-domain fine-tuning and improved prompting strategies. Clinical staff was involved in the qualitative evaluation of the models, especially on the utility of their answers. In the case of Med-PaLM2 clinicians preferred the model's answer to those of clinicians on 8 out of 9 considered axes of utility (*e.g.*, the answer being aligned to the most up-to-date medical consensus, or the reasoning provided within the answer).

GatorTron (Yang et al., 2022) and GatorTronGPT (Peng et al., 2023) are academialed medical LLM projects (Med-PaLM and Med-PaLM2 were developed by Google and DeepMind). GatorTron was pre-trained on health data from the University of Florida, Pubmed, Wikipedia, and MIMIC-III with a BERT-like transformer architecture (the highest number of parameters explored was 8.9 billion – more comparable to the BioMegatron developed by Shin et al. (2020)). GatorTronGPT uses a GPT-3 architecture, with sizes of 5 and 20 billion parameters explored. Similar to the various incarnations of the GPT, GatorTronGPT is a decoder-only LLM. This is in contrast with GatorTron, which is an encoder-only LLM.

GatorTron has been successfully employed in medical relation extraction, semantic textual similarity, medical natural language inference (NLI), and question answering and achieved state-of-the-art performance on several medical NLP benchmarks (Peng et al., 2024). GatorTronGPT has been applied successfully to a vast array of classical clinical NLP tasks, such as relation extraction, NLI, or abbreviation disambiguation. Furthermore, it has been used in generation of sections of clinical notes. Kraljevic et al. (2022) introduced Foresight – a GPT-based model capable of modelling a patient's journey in secondary care based on structured and unstructured input data. Given a timeline of the patient's previous conditions and procedures, the model is capable of forecasting future concepts relevant to the patient, *e.g.*, diseases, symptoms and medications. Ellershaw et al. (2024) use GPT-4 (Achiam et al., 2023) for generating discharge summaries based on discharge summary guidelines and physicians' notes. The output of the model was evaluated by qualified clinicians with prior experience in writing discharge summaries on the occurrence of different error types (including missing a piece of data, hallucination, or addition of irrelevant information). The results – while promising in several metrics – show safety-critical errors, including hallucinations, indicating that the implementation of LLMs in the explored scenario to be challenging and the need of involving a human in the loop for the purposes of review.

To the best of our knowledge, GPT's performance in generating discharge summaries based solely on input code descriptions (representing conditions and procedures) and its ability to perform ICD coding has not yet been reported. Similar to Ellershaw et al. (2024), we use a large language model to generate discharge summaries and have clinicians evaluate their quality. Our research differs in its primary motivation for the generation of discharge summaries being data augmentation for training ICD coding models, and the assessment of the quality and clinical acceptability of the generated documents for a hospital setting being a secondary goal.

5.3 Methodology

GPT-3.5 (gpt-3.5-turbo)[3] (through the OpenAI Python API[4]) was prompted to generate a patient discharge summary based on specific conditions and procedures represented by ICD-10-CM and ICD-10-PCS code descriptions from gold standard labels associated with MIMIC-IV discharge summaries (from the table `hosp/d_icd_diagnoses.csv.gz`). These produced label descriptions chosen to represent realistic combinations of conditions and procedures are not considered a part of the MIMIC-IV dataset. Note that sharing access to MIMIC data via any online API is prohibited[5][6].

Various dataset splits have been proposed since the release of coded discharge summaries in MIMIC-IV. The dataset split proposed by Edin et al. (2023) was deemed unsuitable due to its exclusion of the long tail, which contrasts with the aim of addressing the data sparsity issue through generation techniques (as removing the long tail removes sparse data). Instead, we chose the dataset split proposed in Nguyen et al. (2023), which preserves the long-tail (thus aligning better with the data sparsity focus). The implementation of common ICD coding models produced by Edin et al. (2023) was retained for model training and analysis.

5.3.1 Label Selection

Candidate source documents for generation were selected from MIMIC-IV based on label populations in the selected split. Codes common across training, validation, and test sets were identified and further filtered to those appearing up to 5 times (few-shot) in the training set, resulting in 195 unique codes (compared to 15,353 unique few-shot codes in training).

For these 195 few-shot codes, a list of codes belonging to their families (the codes themselves, and siblings) was produced. Families with at least one relatively frequent code (population > 100 in training) and at least one code exclusive to the test set (zero-shot) were retained. The constraints of including few-shot and zero-shot codes stem from the aim to evaluate GPT-3.5 as a generator for these low-resource labels. The constraint of having at least one frequent code was included to explore the scenario where the model may predict a label due to its dominance in the population, and the

[3] `https://platform.openai.com/docs/models/gpt-3-5`

[4] `https://platform.openai.com/docs/api-reference?lang=python`

[5] `http://web.archive.org/web/20240206013403/https://physionet.org/news/post/415`

[6] The method was consulted with and approved by PhysioNet, as it merely uses the descriptions of attached codes

increase in population of other labels potentially leading to more confusion.

Families with a single candidate code were removed, as their their within-family performance would be trivial, leaving 16 suitable families of which 10 were randomly selected to generate for. This reduction was chosen to balance variety with resources necessary for generation (credit for use of the GPT API and time). For each family, its descendant codes appearing in the training set (139) and the zero-shot codes (16) were gathered, totaling 155 codes. Of these, 114 codes had a population lower than 100 and are henceforth termed *generation codes* (list of codes available in Appendix A).

5.3.2 Preparation of Samples for Generation

In the Nguyen dataset split of MIMIC-IV used for training documents containing at least one of the 98 relevant few-shot codes were selected (the 16 zero-shot were by-definition absent). Some documents contain multiple relevant codes. Documents for each of the relevant codes were collected and duplicated to bring their population up to 100. The gold standard labels from each of the documents were retrieved. This was done in order to replicate realistic scenarios coming from patients, as different conditions and procedures co-occur at different rates (*e.g.*, diabetes leading to diabetic complications, or cancer co-occurring with chemotherapy or radiotherapy). To introduce variety in the dataset, up to 5 of the assigned non-relevant labels coming from duplicates were randomly dropped to create the new set of labels for generation (from hereon referred to as the silver standard).

Documents containing siblings of the 16 zero-shot labels were identified. A silver standard set was created for each of these documents substituting the sibling code with the zero-shot code (similar to the zero-shot approach introduced in Section 4.3.4 and published in Falis et al. (2022)). If multiple candidate "unspecified" siblings were present, a random one was replaced with the zero-shot code.

This preparation resulted in 9,606 input sets of labels, with 6,779 unique and 2,827 duplicated.

5.3.3 Generation

Natural Language Generation is the task of producing output natural language text based on a set of input data and parameters. The "gpt-3.5-turbo-0613" model (GPT-3.5) was used for generating discharge summaries based on input ICD code descriptions. It was chosen given its wide recognition in the field of LLMs, relative cost-

effectiveness, and time efficiency (compared to GPT-4). For distinct label sets, a temperature (parameter in the 0-1 range controlling randomness) of 0 was utilised to produce deterministic outputs. A temperature of 0.1 was employed for duplicate label sets, allowing output variation.

Within the prompt (the generic prompt and an illustrative example prompt are presented in Sections 5.3.3.1 and 5.3.3.2 respectively), the task of writing the discharge summary for a patient was stated along with a list of standard descriptions of their conditions and procedures based on the produced silver standard. The task also included producing a "DISCHARGE DIAGNOSIS AND PROCEDURES" section where the model was to list the ICD-10 codes (code and its standard description) to the discharge diagnoses and procedures. This enabled comparison between the codes whose descriptions were included in the prompt, and the ones predicted to be corresponding to said descriptions by the model. These further specifications were appended to the initial prompt:

- Length of up to 4,000 words (following the maximum cutoff point in previous work (Edin et al., 2023)). The overall input and output token restriction of GPT-3.5 is 4,096 (inclusive of the prompt);
- Inclusion of social and family history;
- Anonymisation was required for personal and location data (due to uncertainty of the anonymity of the data used in training of GPT-3.5), maintaining numeric information when relevant despite potential removal in pre-processing;
- Explicit ICD code mentions within the main text were to be avoided to prevent model association or potential errors;
- Clear numeric values were preferred rather than ranges;
- Providing a specific concept for codes involving the umbrella term "other" encompassing a range of conditions;
- Omission of the keyword "unspecified" present in standard descriptions opting for a more natural means of expression;
- Coding of the discharge summary was to be positioned at the end in a regular pattern (codes in square brackets) for model coding assessment.

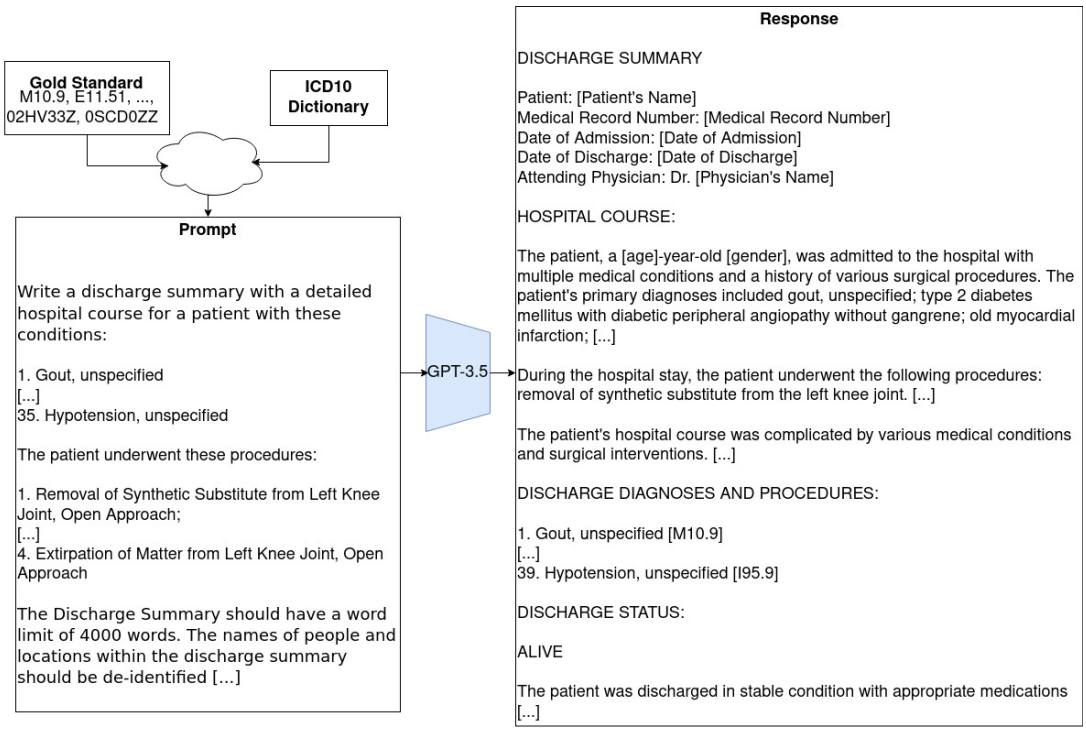

Figure 5.1: An example generation of a synthetic discharge via GPT-3.5

An example of the generation process is presented in Figure 5.1. The generated documents were processed to find the discharge diagnoses and extract the assigned codes with a regular expression.

In total, 9,606 synthetic training documents were generated. All mentions of ICD-10 codes were removed and the documents were pre-processed the same as the baseline data. These documents were merged with the *baseline* training set (110,442 MIMIC-IV documents), forming the *augmented* training set (120,048 documents). The baseline and augmented settings used the same validation and test sets with 4,017 and 7,851 real documents, respectively.

Analysis between the generated text, and MIMIC-IV ICD-10-coded discharge summaries (the entire dataset, and the subset used for source label sets used in generation) showed differences in word count (per label and overall – Figures 5.2a and 5.2b) between synthetic and real data while retaining similar distributions of label counts (Figure 5.2c). The synthetic discharge summaries tend to be shorter compared to the original MIMIC-IV discharge summaries (coming from a hospital in the US) – which are commonly longer than discharge summaries in the UK. This could be the result of GPT-3.5's 4,096 token limit. The difference between the distributions (the synthetic text being on average shorter than the real counterparts for similar sets of labels

in MIMIC-IV) is one of several ways in which the documents produced by GPT-3.5 differ from MIMIC-IV discharge summaries – along with the MIMIC-IV documents sharing fairly similar structure, the presence of typos in MIMIC-IV, etc. – which may have a negative effect on the model's generalisation to unseen real data (*e.g.*, the test set coming from MIMIC-IV). The model will likely overfit to this out-of-distribution representation of zero-shot-label-related text the most, as these do not have an real-world counterparts in the training data, which could provide a more realistic representation.

5.3.3.1 Generic Prompt

When prompting GPT-3.5 to produce synthetic discharge summaries, we have used this template prompt outlining the task for the model along with the lists of ICD condition and procedure codes. The initial prompt consisted of the list of input conditions and procedures and statement of the task of generating a discharge summary for a patient with such conditions and procedures and coding it with ICD-10 codes. The prompt was further refined by adding formatting statements for easier retrieval of the relevant output (predicted ICD-10 codes), while not mentioning individual ICD-10 codes in the main body of the text outside the Discharge section. A maximum length has been set to 4000 based on the input cutoff limit for Mullenbach et al. (2018) which was also used in the codebase of Edin et al. (2023). Realistically this limit was superseded by the limits of GPT-3.5 – 4096 (subword) tokens in the combined text of the prompt and the output. No minimum response size was specified, as no such limitation was considered in the local neural network models used in the experiments. However, a constraint on the minimum input length could have been added – *e.g.*, to match the minimum length of discharge summaries with the same number of assigned codes or overall minimum length in MIMIC-IV. Further prompt engineering involved an iterative process of prompting GPT to produce text for a sample set of codes and identifying issues in the output – *e.g.*, opting for numeric ranges instead of specific numbers, or the obsolete use of *"unspecified"*.The following is a generic version of the prompt used for generating the synthetic data. Except for the two placeholder strings corresponding to the *"list of conditions"* and the *"list of procedures"* which were populated by the code descriptions corresponding to input labels, the rest of the text is common across all of the prompts used.

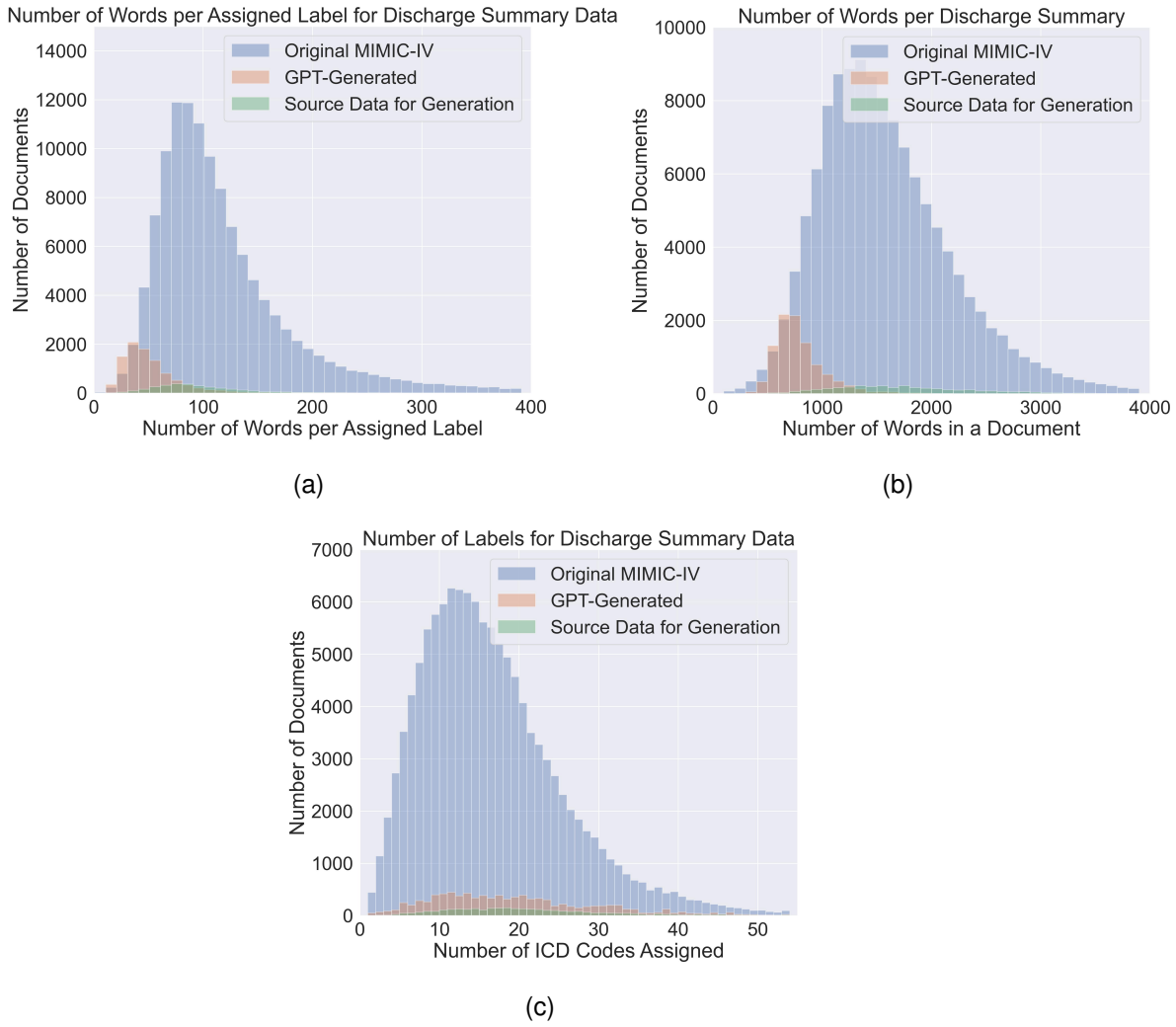

Figure 5.2: A comparison between the discharge summary data in MIMIC-IV, seed MIMIC discharge summaries for generation (the source data), and the generated discharge summaries. Subfigures 5.2a and 5.2b focus on the number of words in documents, indicating that the GPT-generated data generally contains fewer words overall and per assigned label compared to the real data from MIMIC-IV. Subfigure 5.2c demonstrates that, although there's a variance in document size, the distribution of the number of labels per document remains relatively similar across the datasets.

Write a discharge summary with a detailed hospital course for a patient with these conditions:
(list of conditions)

The patient underwent these procedures:
(list of procedures)

The Discharge Summary should have a word limit of 4000 words. The names of people and locations within the discharge summary should be de-identified. Do not state ICD-10 codes in the main body of the text. For any condition involving a numeric range, state explicitly a number within that range (e.g., for a patient with blood glucose between 7.0 and 11.0 mmol/l generate ''patient's glucose level was 8.6 mmol/l''). For conditions with the keyword ''other'' specify the condition that falls into the set of ''other'' - e.g., for ''Other specified congenital malformations of skin'' generate ''Aplasia cutis congenita''. For conditions with the keyword ''unspecified'', do not include the word ''unspecified'' within the main body of text - e.g., for the code H35.00 (Unspecified background retinopathy) generate the statement "the patient suffers from background retinopathy". At the end of the discharge summary add a paragraph with the header "DISCHARGE DIAGNOSES AND PROCEDURES" and assign ICD-10 codes to the discharge diagnoses and procedures, for each concept stating its ICD-10 code and its description (e.g., Essential (Primary) Hypertension [I10]). Finish the document with a DISCHARGE STATUS section for the patient - either "DEAD" or "ALIVE".

5.3.3.2 Example prompt

The following is an example prompt with the lists of conditions and procedures populated.

Write a discharge summary with a detailed hospital course for a patient with these conditions:
1. Type 1 diabetes mellitus with proliferative diabetic retinopathy with traction retinal detachment not involving the macula, right eye
2. Non-pressure chronic ulcer of left heel and midfoot with unspecified severity
3. Type 1 diabetes mellitus with diabetic polyneuropathy

4. Type 1 diabetes mellitus with foot ulcer

5. Type 1 diabetes mellitus with proliferative diabetic retinopathy without macular edema, bilateral

6. Long term (current) use of insulin

7. Major depressive disorder, single episode, unspecified

8. Personal history of nicotine dependence

9. Fever, unspecified

The patient underwent these procedures:

1. Fusion of Left Tarsal Joint with Internal Fixation Device, Open Approach

2. Division of Left Foot Tendon, Open Approach

3. Insertion of External Fixation Device into Left Tibia, Percutaneous Approach

The Discharge Summary should have a word limit of 4000 words. The names of people and locations within the discharge summary should be de-identified. Do not state ICD-10 codes in the main body of the text. For any condition involving a numeric range, state explicitly a number within that range (e.g., for a patient with blood glucose between 7.0 and 11.0 mmol/l generate ''patient's glucose level was 8.6 mmol/l''). For conditions with the keyword ''other'' specify the condition that falls into the set of ''other'' - e.g., for ''Other specified congenital malformations of skin'' generate ''Aplasia cutis congenita''. For conditions with the keyword ''unspecified'', do not include the word ''unspecified'' within the main body of text - e.g., for the code H35.00 (Unspecified background retinopathy) generate the statement "the patient suffers from background retinopathy". At the end of the discharge summary add a paragraph with the header "DISCHARGE DIAGNOSES AND PROCEDURES" and assign ICD-10 codes to the discharge diagnoses and procedures, for each concept stating its ICD-10 code and its description (e.g., Essential (Primary) Hypertension [I10]). Finish the document with a DISCHARGE STATUS section for the patient - either "DEAD" or "ALIVE".

Since the labels translated into code descriptions using an external ICD-10 dictionary are not data lifted verbatim from the corpus, this is acceptable to be sent through the GPT API without employing any further precautions.

5.3.4 Data Augmentation for Local Neural Models

Local neural networks provide the user with control and security with regard to data by allowing to process data without it ever leaving the user (mitigating security concerns for sensitive data, such as hospital records). Furthermore, they are smaller than LLMs – the model size needs to be considered for deployment on hospital hardware. For this reason, the main aim of this chapter is the exploration of viability of GPT-3.5 as a source of synthetic data for augmenting the training of local neural ICD coding models.

Most recent LMTC neural architectures are encoder-decoder models whose encoder processes the input text to generate a latent representation. Architectures using a non-BERT-like encoder – *e.g.*, in CAML (Mullenbach et al., 2018), LAAT (Vu et al., 2020b), or Multi-Res CNN (Li and Yu, 2020b) (described in Sections 2.5.2, 2.5.4, and 2.5.5 respectively) utilise non-contextual (*e.g.*, Word2Vec Mikolov et al. (2013a)) word embeddings, while BERT(Devlin et al. (2018))-like encoders (*e.g.*, in PLM-ICD of Huang et al. (2022b)) enable contextual token representation. The decoder determines a probability for each label based on the latent representation. A probability threshold determines positive predictions.

5.3.5 Experimental Design

The following four evaluation rounds were conducted:

- **Local Neural Model Evaluation:** Assessing CAML, LAAT, and Multi-Res CNN models' performance on Nguyen's test set. Models were trained either solely on Nguyen's training set or enhanced with data generated by GPT-3.5 (the augmented training set).
- **GPT's coding on real data:** Evaluation of GPT-3.5's coding ability on MIMIC-IV using Nguyen's test set;
- **GPT's coding on GPT-generated data:** Evaluation of GPT-3.5's coding ability on generated documents (with provided code descriptions in the prompt)
- **Acceptability of Generated Data in Clinical Practice:** Reviewing GPT-3.5-generated discharge summaries by clinical professionals to gauge their suitability in clinical settings.

An example evaluation by a clinician can be seen in Figure 5.3.

Synthetic Discharge Summary	**Non-Low-Resource Labels**
DISCHARGE SUMMARY Patient: [Patient's Name] Medical Record Number: [Medical Record Number] Date of Admission: [Date of Admission] Date of Discharge: [Date of Discharge] Attending Physician: Dr. [Physician's Name] HOSPITAL COURSE: The patient, a [age]-year-old [gender], was admitted to the hospital with multiple medical conditions and a history of various surgical procedures. The patient's primary diagnoses included gout, unspecified; type 2 diabetes mellitus with diabetic peripheral angiopathy without gangrene; old myocardial infarction; [...] mechanical loosening of internal left knee prosthetic joint, initial encounter; [...] During the hospital stay, the patient underwent the following procedures: removal of synthetic substitute from the left knee joint, open approach; [...] The patient's hospital course was complicated by various medical conditions and surgical interventions. The patient received appropriate medical management, [...] The patient's surgical procedures, including the removal of synthetic substitute from the left knee joint and insertion of a spacer, [...] DISCHARGE DIAGNOSES AND PROCEDURES: 1. Gout, unspecified [...] 39. Hypotension, unspecified DISCHARGE STATUS: ALIVE The patient was discharged in stable condition with appropriate medications [...]	M10.9: Gout, unspecified; E11.51: Type 2 diabetes mellitus with diabetic peripheral angiopathy without gangrene;[...] 0SCD0ZZ: Extirpation of Matter from Left Knee Joint, Open Approach **Low-Resource Labels** T84.033A: Mechanical loosening of internal left knee prosthetic joint, initial encounter **Correctness** Correctness Score - Non-Low-Resource (1-5): *5* Correctness Score - Low-Resource (1-5): *5* Correctness Comment: *All codes are present* **Informativeness** Informativeness Score - Non-Low-Resource (1-5): *3* Informativeness Score - Low-Resource (1-5): *2* Informativeness Comment: *Within the discharge summary text, the code descriptions are listed in a long paragraph and provided with minimal context or narrative. The description of the low resource code is limited to naming the code only.* **Authenticity** Authenticity Score - Patient (1-5): *1* Authenticity Score - Scenario (1-5): *2* Authenticity Comment:*This would be a very complex discharge summary to write given the very large number of codes. Unfortunately the summary has little or no attempt at providing a narrative or context for why the given conditions occurred. There is minimal effort at describing the treatment, or joining up the conditions, or providing a sense of what is most important in this clinical case.* **Acceptability** Acceptability Score (1-5): *1* Acceptability Comment: *Too vague, generic phrases, no context or narrative.*

Figure 5.3: An example evaluation of a synthetic discharge summary by a clinical expert.

5.3.5.1 Local Neural Model Evaluation

The codebase and training procedure of Edin et al. (2023) was employed to train and evaluate CAML, LAAT, and Multi-Res CNN on the ICD coding task using Nguyen's split. Each model was trained with 20 epochs on the training set. At the end of each training round the model was evaluated on the validation set, selecting the model with the highest mean average precision from each run as the final model. Subsequently, this final model underwent evaluation on the test set. Additionally, PLM-ICD, considered a state-of-the-art model for this task, was explored. However, its performance on the baseline and augmented data was notably lower than previously reported. The implementation of this model has been reported to be unstable during training (Edin et al., 2023) while the research presented in this thesis was conducted, with its performance being sensitive to random seeds. In the interest of avoiding random-seed fine-tuning, which would ultimately not lead to representative model performance, PLM-ICD was not included in the final experiments and hence only the performance for CAML, LAAT, and Multi-Res CNN is reported.

For evaluating the model's test performance, standard information retrieval metrics commonly used in LTMC tasks were employed: micro- and macro-averaged Precision, Recall, and F_1-scores. Micro-averaging assigns equal weight to each prediction, favouring high-population classes (*e.g.*, hypertension). Macro-averaging, in contrast, computes the performance for each unique label and averages across the label space, giving each label's average result equal weight regardless of their population. This highlights poor performance in less common classes. The primary evaluation metrics common with the majority of previous work are micro-F1 and macro-F1 scores. For further explanation of standard metrics, please consult Section 3.3.1.

5.3.5.2 GPT's coding of real clinical notes

To test GPT's ability to assign diagnosis codes based on real clinical notes, the Azure AI Services API[7] was employed. The API returns a free text response which was post-processed to retrieve the predicted ICD-10 codes. The post-processing starts with a response format validation. For correctly structured arrays of JSON objects, the predictions were simply extracted. For incorrectly structured outputs, a regular expression pattern was employed to extract all diagnoses and ICD code pairs. The result is a list of predicted diagnoses and corresponding ICD-10 codes for each clinical note. Figure 5.4

[7] https://azure.microsoft.com/en-gb/products/ai-services

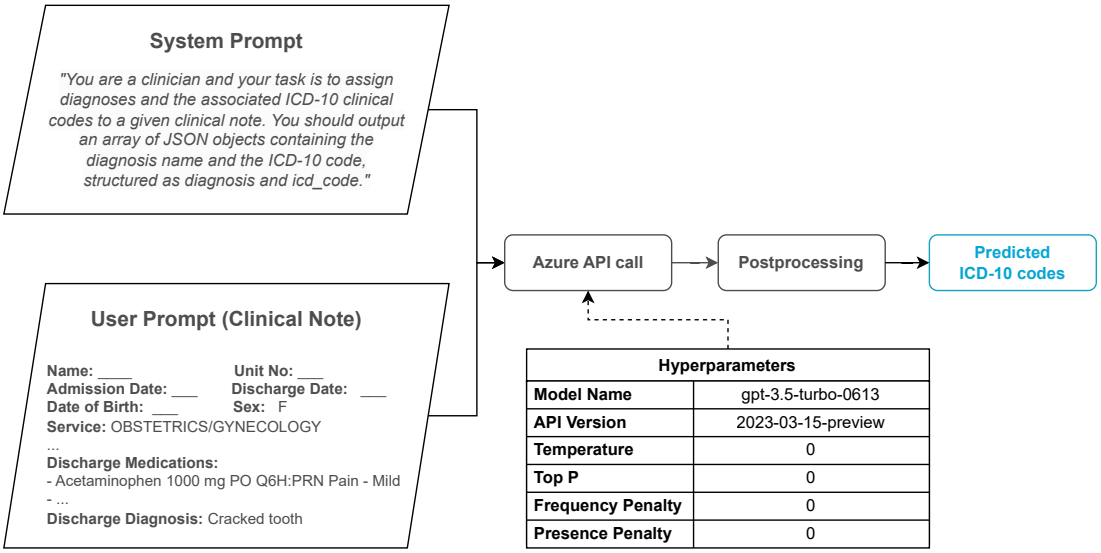

Figure 5.4: The workflow of the GPT-3.5 prediction. Azure AI Services API was used to query GPT-3.5 followed by a postprocessing was step to extract the predicted diagnoses and ICD-10 codes for each clinical note.

illustrates the API call workflow.

For reproducibility, the model version and the API version were specified as *"gpt-3.5-turbo-0613"* and *"2023-03-15-preview"*, respectively. All parameters were set to zero for deterministic responses from GPT-3.5, including temperature, top P, frequency penalty, and presence penalty. The system prompt directed GPT-3.5 to act as a clinician assigning ICD-10 diagnosis codes to clinical notes, specifying the expected output format as JSON objects with keys *"diagnosis"* and *"icd_code"*. The Hyperparameters table in Figure 5.4 presents all hyperparameter details. The code implementation can be found in a Github repository [8].

Furthermore, human review of the data was opted out of for two reasons. Firstly, the terms of the data use agreement of MIMIC-IV[9] did not grant us the authority to permit a third party to process the data for abuse detection. Secondly, the likelihood of harmful misuse was assessed to be low given the sensitive nature of the clinical notes. With these conditions fulfilled further human review (which we could not ensure would be done by a PhysioNet-cerdentialed employee) of the clinical data used via the API was not necessary.

In the evaluation of GPT's performance, beyond standard evaluation metrics, hierarchical evaluation techniques were employed – set-based hierarchical evaluation and

[8] https://github.com/EdinburghClinicalNLP/chatgpt_icd_coding
[9] http://web.archive.org/web/20240206013403/https://physionet.org/news/post/415

Count-Preserving Hierarchical Evaluation (CoPHE) Falis et al. (2021) (introduced in Chapter 3). These metrics award partial credit to mispredicted labels by extending both the prediction and gold standard sets with their ancestor labels. Set-based evaluation determines the presence of ancestor codes, depending on the existence of at least one descendant code.

In CoPHE, ancestor labels link to the count of descendant codes, penalizing both over- and under-predictions within code families. Comparing set-based and CoPHE results helps to evaluate a model's tendency to over-/under-predict. A lower CoPHE score indicates this phenomenon. See Section 3.3.2 for further details on calculating hierarchical scores.

Weak Hierarchical Confusion Matrices (WHCM) (Falis et al., 2022)(introduced in Chapter 3) were utilised to summarise and visualise in-family versus *Out of Family* (OOF) prediction errors. These metrics were chosen to explore how expanding the population of codes within code families to a minimum of 100 instances impacts within-family performance. Within-family errors involve false positives that align with false negatives within the same family in the gold standard. On the other hand, an OOF error for a false negative in the gold standard lacks a false positive within the prediction set from the same family to match. The primary goal in generating synthetic data is to reduce OOF errors, enhancing true positive predictions or ensuring mispredictions occur within a family.

5.3.5.3 GPT's coding on Synthetic data

The prompt asked GPT-3.5 to code the conditions and procedures mentioned in the document it generated. This experiment tested GPT-3.5's ability to assign ICD-10 codes to concepts presented in their standard descriptions. Alongside this, the prompt required creating a patient's social and family history, which might have led to the model introducing new conditions like substance abuse and potentially coding them, despite not being part of the initial prompt.

5.3.5.4 Acceptability of Generated Data in Clinical Practice

Four clinical professionals (co-authors of Falis et al. (2024) SB, LD, MH, and RP) assessed the quality of the generated data. As the data was generated based on labels associated with MIMIC-IV discharge summaries, this evaluation included both synthetic discharge summaries generated by GPT-3.5 and discharge summaries from

MIMIC-IV. The clinicians were presented with 20 discharge summaries – 10 synthetic and 10 real (based on whose adjusted gold standard the synthetic ones were generated).

Each discharge summary was assessed for:

- Correctness – accuracy in describing patient conditions and procedures (reported on frequent and rare codes separately);
- Informativeness – clarity and sense in supporting information – *e.g.*, test results, medication suggestions (reported on frequent and rare codes separately);
- Authenticity (patient) – whether such a patient could exist (considering all codes regardless of rarity);
- Authenticity (clinical scenario) – whether the hospital course was plausible as reported (considering all codes regardless of rarity);
- Acceptability – suitability of the document for clinical use (considering all codes regardless of rarity).

Similar dimensions for evaluation have been used in previous work on the use of LLMs in medicine – *e.g.*, the CLEAR tool by Sallam et al. (2023) focusing on assessing the generated text on its completeness, lack of incorrect information, presence of supporting evidence, appropriateness, and relevance; or the evaluation of Cocci et al. (2024) focusing on accuracy, comprehensiveness, and clarity in LLM-generated responses to urology case studies.

Additionally, they separately evaluated correctness and informativeness for both non-low-resource and low-resource labels to gauge GPT-3.5's ability to generate low-resource data. Scores from 1 (failure to perform) to 5 (perfect performance) were assigned to each metric, with accompanying comments justifying the score.

5.4 Results

5.4.1 Local Neural Model Evaluation

Performance was assessed across three code subsets: the entire codeset in MIMIC-IV (*overall*), a restricted generation set (g) containing only the 114 low-population candidate generation labels, and a set of codes from the code families present in g (f). Results are shown in Table 5.1. Baseline results for CAML and LAAT align closely

with prior findings by Nguyen et al. (2023) for *overall* metrics. While Nguyen et al. (2023) did not report on Multi-Res CNN, its performance trends were similar to CAML and LAAT in comparison to Edin et al. (2023) on non-filtered codesets. LAAT excels in micro-F_1, Multi-Res CNN leads in macro-F_1, and CAML generally lags behind. Baseline models outperform augmented ones in *overall* micro-F_1, a common observation when enhancing lower-resource label performance. Nonetheless, macro-F_1 scores improved for two out of three models within the *overall* codeset and for all models on f and g. Multi-Res CNN and CAML macro-F_1 scores display sizable relative improvement (26% and 78% respectively) in g.

Augmented models performed on par with or outperformed baseline models in micro-F_1 scores for f and g. Augmented Multi-Res CNN outperforms its baseline in micro-F_1 for both f and g, indicating benefits for the code family from augmenting less-populous members. Augmented LAAT shows improvement in macro-F_1 in both f and g but lags in micro-F_1. LAAT's performance may have been biased towards high-population classes and the augmentation's boosting of low-frequency classes (misrepresenting their frequency) may have introduced confusion. Apart from having a recurrent encoder (Bi-LSTM), the LAAT model employed in this experiment is about twice the size of the Multi-Res CNN (21.9M versus 11.9M parameters). This added model complexity may have enabled better performance on already frequent labels, but increased the need for more examples of lower-resource labels.

The results in Table 5.1 show an improvement for macro-averaged F_1-score across all sets (overall, all codes within families in generation, and generation-codes only) for the CAML and MRCNN models. The only instance where the macro-averaged F_1-score for a model trained on the augmented training set was worse than its baseline counterpart is for the overall codeset with LAAT. This may be due to the model having significantly more parameters than CAML and MRCNN. When focusing on the macro-averaged F_1-scores for the candidate codes used in generation (the g codeset), the improvement is relatively modest for LAAT (14.98 on augmented versus 14.48 in the baseline), while CAML and MRCNN experienced a significant boost, and the augmented MRCNN surpassed augmented LAAT on this metric. This is especially interesting, as baseline LAAT performed better than baseline MRCNN on these candidate codes. While MRCNN surpassed LAAT's performance on the f codeset comprising all the codes from the families chosen for generation both in the baseline and augmented scenario, the gap between the augmented models is significantly wider (absolute difference of 0.08 for baseline, and 0.47 for augmented) with MRCNN benefiting in no

Table 5.1: A comparison between local neural network models (MRCNN stands for Multi-Res CNN) trained on baseline ($base$) and augmented (aug) training sets is evaluated using micro- and macro-averaged F_1 scores (mi and ma respectively) on three codesets – ov (overall) on all codes present in MIMIC-IV; f comprising all codes within the families chosen for generation; and g corresponding to candidate codes used in generation with a population of at most 100 in the training set. The highest score in each metric for each model pair (baseline versus augmented) is highlighted in **bold**. Weak Hierarchical Confusion Matrix (WHCM) error rates are produced for codesets f and g. Performance on the common test set is reported using the macro-averaged proportion of errors that were Out-of-Family (OOF) and within-family (IF). The best (lowest) error rate for each error type for each model pair (baseline versus augmented) is presented in bold.

Experiment	$F_1 \uparrow$						WHCM error $\downarrow$			
	mi_{ov}	ma_{ov}	mi_f	ma_f	mi_g	ma_g	OOF_f	IF_f	OOF_g	IF_g
CAML$_{base}$	**53.65**	3.87	**38.43**	3.03	17.41	6.64	66.53	25.05	83.81	9.83
CAML$_{aug}$	53.54	**3.90**	38.41	**3.78**	**20.68**	**11.86**	**65.98**	**23.77**	**79.79**	**9.17**
LAAT$_{base}$	**57.29**	**6.18**	**43.59**	4.96	**26.79**	14.48	58.57	**28.35**	74.03	12.20
LAAT$_{aug}$	57.18	6.09	43.36	**5.38**	25.70	**14.98**	**55.93**	29.78	**73.65**	**11.98**
MRCNN$_{base}$	**55.66**	6.40	40.16	5.04	26.80	13.92	52.72	32.41	**69.68**	15.24
MRCNN$_{aug}$	54.69	**6.46**	**42.69**	**5.85**	**30.39**	**17.68**	**49.65**	**32.36**	70.41	**10.22**

small part from the significant improvement on the g codeset.

Given that the WHCM error statistics are macro-averaged, the improvements seen in macro-averaged F_1 scores is also translated into the WHCM error metrics. CAML – with the lowest number of parameters and worst-performing across the F_1 metrics among the three models – improved on all reported WHCM error metrics. The most substantial improvements for CAML are in the OOF_g and IF_f error metrics. The modest changes to the remaining WHCM error metrics for CAML (OOF_f and IF_g) imply that the model's performance improved for the codes targeted for generation without introducing more in-family errors with respect to the families of the generated codes.

In the case of LAAT, the improvements on the WHCM error metrics in the g codeset are comparatively modest, though it should be noted that compared to both baseline and augmented CAML, even baseline LAAT tended to produce fewer errors, whose proportion was lower on OOF than CAML's while higher on IF. A significant improvement can be observed on OOF_f accompanied by increase in IF_f. This shows the augmentation leading to a shift from predicting codes from families outside of f to predicting codes within f that do not match the gold standard code coming from f. Hence the macro F_1 scores for f in LAAT improved thanks to more codes from f being predicted, even though some of the extra predictions were erroneous.

In contrast, for MRCNN OOF_g deteriorated with augmentation – the only model displaying such behaviour (note that MRCNN_{base} has the lowest OOF_g of all the reported models followed closely by MRCNN_{aug}). Hence, more codes were predicted outside of the family when considering codeset g than before augmentation. However, this was coupled with a significant improvement in IF_f. Furthermore, even with the slightly worse OOF_g the error statistics on f showed improvement for both OOF and IF. This shows that the improvement on the codes used in generation (codeset g) was not to the detriment of performance on the codes of their families.

Unlike baseline models, models trained on augmented data occasionally predicted codes absent from the original data, although incorrectly, except for one correctly predicted code (S02.63XA) by a Multi-Res CNN model trained on the augmented dataset. While consistent enhancement in zero-shot code performance was not achieved through augmentation, the potential for improvement is evident.

Table 5.2: Results of GPT-3.5's coding ability on real and self-generated data. Note that these metrics are not directly comparable – rather they are meant to be used in tandem. For further explanation please refer to Section 3.4.1

Evaluation Set	Leaf-Only			Set-Based			CoPHE		
	P	R	F_1	P	R	F_1	P	R	F_1
GPT-3.5 Real	9.46	33.51	14.76	10.59	44.87	17.13	10.30	44.33	16.72
GPT-3.5 Synthetic	59.06	40.72	48.20	66.46	41.32	50.96	67.20	41.55	51.35
Baseline LAAT Real	60.42	54.46	57.29	61.28	54.50	57.68	60.84	54.33	57.39

5.4.2 GPT's coding ability on real and synthetic data

GPT-3.5's ability to code real MIMIC-IV documents and generate coded documents with explicit code descriptions in the prompt was examined. The results (Table 5.2) show that the performance on prompt-guided self-generated (synthetic) data resembles that of local models on the MIMIC-IV test set, not surpassing it. Hierarchical metrics show higher precision, recall, and consequently, F_1-score in CoPHE compared to set-based hierarchical evaluation indicating errors coming from within-family misprediction, rather than incorrectly estimating the number of expected labels.

However, the performance on the MIMIC-IV test set is notably low, especially in precision. The improvement in the precision from leaf-only results to hierarchical is minimal. This implies that incorrect predictions were more likely to be out-of-family. Moreover, results on CoPHE are lower than on the set-based hierarchical evaluation indicating a tendency of the model to over-/under-predict within the scope of the family – an issue previously reported in local ICD coding models (Falis et al., 2021), and present in the reported hierarchical results for baseline LAAT.

These results demonstrate that GPT-3.5 can identify ICD-10 codes based on provided descriptions if presented within the prompts. Its performance when tasked with standard ICD coding without explicitly identified concepts or non-standard surface forms of the concepts significantly deteriorates.

5.4.3 Acceptability of Generated Data in Clinical Practice

The inter-evaluator agreement for the 7 metrics was calculated using Fleiss' kappa (κ) Fleiss (1971). As κ is designed for categorical variables and does not fully capture ordinal scores, also produced the mean scores for each metric (μ). The results are

presented in Table 5.3.

The evaluators' agreement was poor ($\kappa < 0$) in examples from MIMIC-IV for Correctness and Informativeness of non-low-resource codes, and the Acceptability of the discharge summaries. For the other metrics, a $\kappa > 0$ was reached but never exceeded 0.4 (lower than moderate agreement). All mean scores are higher than 4. Hence, while the clinicians disagreed on the exact scores, they rated real discharge summaries positively. The disagreement may be due to clinicians being UK-based with significant differences in reporting style within the UK and the US (where MIMIC-IV is from).

For GPT-generated summaries, slight agreement was seen in Acceptability, and fair agreement in the Correctness of low-resource labels. All other metrics had poor agreement. Both Correctness metrics scored above 4, with low-resource Correctness surpassing 4.5—an encouraging outcome for the primary goal of generating low-resource code data. Mean Informativeness in the low-resource scenario and authenticity of scores were at least 3. Once again, performance on the low-resource codes exceeded non-low-resource codes. Other metrics had μ scores above 2. Informativeness and Authenticity for non-low-resource codes had a poor agreement ($\kappa < 0$), while Acceptability had some agreement with the lowest mean score of 2.225.

Table 5.3: Evaluator agreement (κ) and mean scores (μ) for samples from MIMIC-IV (real), versus GPT-generated (synthetic) data.

Metrics	κ_{real}	μ_{real}	$\kappa_{\text{synthetic}}$	$\mu_{\text{synthetic}}$
Correctness – Non-Low-Resource	-0.386	4.175	-0.163	4.375
Correctness – Low-Resource	0.043	4.350	0.206	4.525
Informativeness – Non-Low-Resource	-0.155	4.550	-0.220	2.775
Informativeness – Low-Resource	0.241	4.675	-0.277	3.000
Authenticity – Patient	0.340	4.750	-0.078	3.150
Authenticity – Scenario	0.373	4.775	-0.333	2.250
Acceptability	-0.056	4.550	0.035	2.225

We have opted for human expert evaluation for the analysis of the generated text. Machine evaluation, such as *Recall-Oriented Understudy for Gisting Evaluation (ROUGE)(Lin and Och, 2004)* – a set of metrics which is often employed for tasks involving text generation (*e.g.*, automatic summarisation or machine translation) and its different metrics focus on concepts, such as phrase overlap between the produced text and a reference

gold standard text or co-occurence statistics – could have been employed in order to compare the synthetic discharge summaries to MIMIC-IV records with similar assigned labels. However, the GPT-generated discharge summaries differed significantly from the real documents (*e.g.*, through writing style, adherence to grammatical rules, and the ICD-10-related concept-level information heavily reusing the ICD-10 code descriptions provided in the prompt). Moreover, the surface-form-level focus of these metrics would not capture the holistic features of the discharge summaries.

The clinical evaluators have identified several challenges in the generation of natural-looking clinical notes:

GPT-3.5 tends to do verbatim reproductions of the prompted diagnoses list: GPT-3.5 tends to copy all concepts mentioned in the prompt when generating a clinical note. While instruction-following is a desirable behaviour, excluding non-crucial details is essential when generating a natural-looking clinical note. Real clinical notes often omit irrelevant and less critical findings for brevity, particularly if the information is inferrable from surrounding contexts such as medications and treatments. For instance, GPT-3.5 unnecessarily noted a normal BMI.

GPT-3.5 may phrase diagnoses in an unnatural manner: GPT-3.5 tends to use an overly technical and unnatural style when specifying diagnoses. For instance, GPT-3.5 mentioned *"anaemia, which was unspecified."* in the generated clinical note as it was prompted with *"D64.9: Anemia, unspecified"*. GPT-3.5 also occasionally introduces vague phrases (*e.g.*, *"geriatric team provided supportive care, including behavioural interventions and medication management"*) without further detail. This contrasts with the more streamlined language of real clinical notes.

GPT-3.5 lacks details when introducing supporting information: GPT-3.5 tends to introduce crucial supporting information without sufficient details. For instance, GPT-3.5 mentioned *"Following a traumatic event"* without further specification of the mentioned traumatic event, which is unacceptable in the clinical setting.

This omission limits the overall informativeness of the patient's medical context, potentially hindering the notes' usability for a comprehensive view.

GPT-3.5 may introduce spurious supporting information: GPT-3.5 sometimes introduces improbable but possible details. For instance, GPT-3.5 overemphasised the significance of a patient's anxiety disorder regarding an episode of syncope and a subsequent facial fracture, which the clinicians consider unlikely.

GPT-3.5 failed to present diagnoses as interconnected events: GPT-3.5 does not effectively present diagnoses as interconnected, resulting in fragmented notes that lack

coherence. The clinicians described GPT-3.5-generated clinical notes as collections of unrelated facts. For example, GPT-3.5 presented complications of Type 1 diabetes mellitus (*i.e.*, *E10.43: Type 1 diabetes mellitus with diabetic autonomic (poly)neuropathy*, *E10.621: Type 1 diabetes mellitus with foot ulcer*, *E10.628: Type 1 diabetes mellitus with other skin complications*, *E10.21: Type 1 diabetes mellitus with diabetic nephropathy*, *E10.319: Type 1 diabetes mellitus with unspecified diabetic retinopathy without macular edema*) separately without conveying their relation. Within the same note GPT-3.5 failed to recognise the connection between vascular complications and diabetes, presenting them as independent facts. The lack of coherence between diagnoses may impede the plausibility of the clinical note and undermine the overall acceptability and usefulness of synthetic notes.

GPT-3.5 failed to prioritise and emphasise critical diagnoses: GPT-3.5 struggles to prioritise diagnoses based on clinical significance, which undermines the authenticity of the portrayed scenario. For example, GPT-3.5 often places critical conditions on the same level as minor issues, such as impacted ear wax, cataracts, and conjunctival haemorrhage. Hence, it was concluded that GPT-3.5 struggles to effectively convey the relative clinical significance of certain diagnoses.

5.5 Conclusion and Future Work

This work investigated the capability of GPT-3.5's potential in augmenting ICD-10 coding for local neural models in low-resource scenarios. While overall performance dipped with synthetic data augmentation, filtered codeset evaluation showed improvements, especially in advanced models like LAAT and Multi-Res CNN. Error analysis indicated augmented models made fewer out-of-family predictions, with some shift to within-family errors (closer to the correct answer). Augmentation showed promise in improving the prediction of generated codes and their siblings. Zero-shot labels did not consistently benefit from the augmentation, emphasising the need for real data in augmentation success. However, a zero-shot code learned from the synthetic data was predicted correctly. The potential of LLM-generated discharge summaries should further be explored with different (*e.g.*, local or specialised) LLMs, prompt engineering, and further supplementing the prompt with external knowledge (*e.g.*, from ontologies).

In guided synthetic settings with ICD-10 descriptions, GPT-3.5 showed partial code identification ability displaying lesser over-/under-prediction tendencies than previously reported local models. It, however, struggled in the realistic scenario without

in-prompt aid, performing below locally-trained models. Hence, the explored setup of producing a synthetic document based solely on the associated ICD codes is unsuitable for deployment in a clinical setting.

Clinician-evaluated synthetic discharge summaries showed correctness in individual codes, yet lacked naturalness and coherence compared to real data, resulting in lower informativeness, authenticity, and acceptability scores. Synthetic summaries failed to represent holistic patient narratives or prioritise critical diagnoses.

One potential solution to generating synthetic discharge summaries involves restructuring the prompt to order diagnoses chronologically, providing their corresponding timestamps. This could guide LLMs in creating synthetic notes mirroring the chronological progression of a patient's medical journey, enhancing coherence and prioritisation.

Another promising solution is to retrieve real clinical notes as in-context learning examples to help guide the generation process Lewis et al. (2020b) to aid LLMs in generating more realistic and coherent content. As this study focused on evaluating LLMs' existing capability, an evaluation in a zero-shot framework was selected. Future work may explore this idea's potential for generating more realistic-looking clinical notes.

5.6 Limitations

In this study, while the annotation experts are involved as co-authors, we ensured that they were independent from the development of the algorithms that involved the synthetic data. While the evaluation utilised few clinical experts (n=4), they provided sufficient expertise in evaluating the notes. The study was blinded with respect to the real/synthetic status of documents, but according to the experts the synthetic data differed from real enough to be distinguishable.

While some features of MIMIC-IV may be difficult to replicate – *e.g.*, the writing styles of individual doctors or presence of typos – others could be addressed with increased input limit in the prompt. Examples of these include guidelines on formatting, structure of the report, or the expected length of a discharge summary given an input set of codes and statistics drawn from MIMIC-IV data (notably an issue visible in the shifts of distributions for number of words per label and number of words per document between real and synthetic data). Addressing these differences would potentially also improve performance of models trained on the synthetic data, as the synthetic

documents seen during training would resemble the MIMIC-IV data and hence be less likely to be outliers with respect to the real training and evaluation documents.

Chapter 6

Conclusion

In this thesis we have explored the integration of ontological knowledge – structure, verbal descriptions of concepts, links among ontologies – in the task of coding discharge summaries with the *International Classification of Diseases* (ICD). Our primary focus was on addressing concept sparsity, which we have pursued through data augmentation and synthesis with rule-based and *Large Language Model* (LLM)-based methods. We have further investigated evaluation approaches in the space of *Large-Scale Multi-Label Text Classification* (LMTC) to facilitate analysis of ICD coding models' performance. In this chapter we will summarise the conclusions reached within Chapters 3, 4, and 5, comment on the ethical considerations of the thesis, and propose directions for future work.

6.1 Summary of Conclusions and Contributions

With the aid of the hierarchical evaluation approaches which we designed as part of the thesis (*Count-Preserving Hierarchical Evaluation* (CoPHE) and Weak Hierarchical Confusion Matrices (WHCM)), we have analysed the performance of local neural LMTC models. Through CoPHE, whose count-preserving hierarchical property assigns partial credit to incorrect predictions based on the hierarchical ontological structure, we have found that local neural LMTC models (*e.g.*, the original CAML (Mullenbach et al., 2018) or the more recent LAAT (Vu et al., 2020a)) have higher performance reported by set-based hierarchical evaluation compared to CoPHE. This indicates that, given an input document, these models tend to predict incorrect number of codes from the families of codes present in the gold standard. Hence, these errors stem from over- and under-prediction, rather than within-family confusion. This

observation has been confirmed through the use of WHCM, indicating a major issue of false negatives stemming from underprediction with regard to families – *Out of Family* (OOF) errors – compared to within family confusion. A further interesting observation regarding CoPHE versus set-based hierarchical evaluation performance was made in the case of a LLMs, where GPT-3.5's poor performance on coding real test data displayed the previously mentioned pattern of higher set-based score compared to CoPHE, but demonstrated that a model can have a higher CoPHE score compared to set-based in the scenario where GPT-3.5 coded synthetic data as part of the generation process. We hypothesise the model being guided by the list of code descriptions within the prompt (which were reproduced as discharge diagnoses and procedures and matched with codes) led to similar numbers of codes being produced as prompted, which contributed to lower prevalence of over-/under-prediction. These findings were made possible through our development of hierarchical evaluation techniques.

With regard to Data Augmentation and Data Synthesis, both the rule-based and LLM-based data enrichment methods have improved on the baseline demonstrating potential in approaching the few-shot/zero-shot scenario through data. In particular, in rule-based data augmentation and synthesis, we have shown the potential of combining *Named Entity Recognition and Linking* (NER+L) with knowledge from ontologies. Verbal descriptions of concepts, their synonyms, and connections with other ontologies were utilised to increase the number of datapoints for training with a variety of surface forms while keeping the gold standard labels unchanged. With the use of the ontological structure we have further proposed the approach of "specifying the unspecified" sibling concepts in order to introduce new concepts into the training data.

We have found that Large Language Models can be utilised for synthesising new documents based on lists of descriptions of conditions and procedures (even rare ones), which can then be used to supplement the training of local LMTC models. By employing standard descriptions of ICD-10 codes and applying the "specifying the unspecified" approach, we have improved the overall macro-F_1 performance of local LMTC models. Performance was further evaluated on the code families on which generation was performed (involving both rare codes targeted for generation and their siblings), and solely the codes specifically selected for generation using the WHCM. Augmenting the training set with synthetic data improved the error rates, particularly the OOF error rate. The most encouraging finding was that a model trained on an augmented training set managed to correctly predict a code within the test set, which was originally absent from the baseline training set. This result suggests that the model learned

to predict this code from the synthetic data.

Finally, with the aid of clinical experts we have found that GPT-3.5-generated discharge summaries mention individual clinical concepts correctly (regardless of whether the concept is rare or not), but the interaction between concepts and the overall narrative suffers. The results of data augmentation experiments show that these synthetic discharge summaries, while not authentic enough for clinical use, can be utilised for training local models.

The thesis' contributions are as follows:

1. A Count-Preserving Hierarchical Evaluation metric implemented for ICD coding of discharge summaries, but applicable to LMTC tasks in general (published in Falis et al. (2021));

2. An extension to the popular Confusion Matrix analysis tool – Weak Hierarchical Confusion Matrix – enabling its use in LMTC tasks (published in Falis et al. (2022));

3. Rule-based data augmentation combining outputs of existing NER+L methods with synonyms of relevant medical concepts based on medical ontologies (published in Falis et al. (2022));

4. Synthesis of documents containing labels unseen within the training data based on mentions of related labels "unspecified" aetiologies (published in Falis et al. (2022));

5. Evaluation of the Large Language Model GPT-3.5 in the context of (1) generating discharge summaries using the ICD-10 descriptions of conditions and procedures assigned to real patients within the prompt in order to produce data for less common labels for training smaller local ICD-coding models; (2) clinical viability of the generated discharge summaries according to clinical professionals with experience in writing such documents; and (3) classification of MIMIC-IV discharge summaries with ICD-10 codes (all published in Falis et al. (2024));

6. A dataset of GPT-3.5-generated discharge summaries produced as part of Falis et al. (2024) available via PhysioNet[1].

[1] https://physionet.org/content/generated-codes-low-resource/1.0.0/

6.2 Discussion

6.2.1 Evaluation Metrics

The thesis focused on two revisions of the ICD - ICD-9 and ICD-10. While different in the amount of detail, number of codes, and structure of the chapters, these revisions follow similar design principles (tree structure) and hence translation of evaluation methods designed for ICD-9 in the early stages of the thesis (CoPHE and WHCM in Chapter 3) to ICD-10 was relatively straightforward. However, the design of ICD-11 differs significantly from ICD-9 and ICD-10 (as discussed in Chapter 2) and hence it is worth considering how the evaluation methods developed as part of the thesis would translate to ICD-11.

ICD-11 utilises a continually updated foundation component, which is a generic graph. The foundation component can be used to produce tree-structured derivatives via the process of linearisation. Such derivatives can then be used for specific purposes – *e.g.*, conditions for coding in a given healthcare system. If we were to compare ICD-11 to ICD-10, the foundation component of ICD-11 is similar to the original ICD-10 tree in that they are the core form of the ontology, whereas derivatives produced from the foundation component of ICD-11 via linearisation are analogous to ICD-10-CM, as these are derived from the respective core form of the ontology and designed for a specific purpose. CoPHE and WHCM were designed to work with a tree structure. As such, they can be applied to the linearised derivatives of ICD-11, as, like ICD-10-CM (or the base ICD-10), these are structured as trees. This is not the case for the foundation component, as it is a generic graph.

While the design of CoPHE and WHCM was intended for evaluating model performance in the task of document classification, for which the linearised derivatives (rather than foundation component) would be used, it is worth exploring how these methods could be extended to the generic graph scenario – *e.g.*, in order to explore performance across a range of possible derivatives of the foundation component. Such extensions to generic graphs would also be useful in other ontologies with similar structure, such as SNOMED CT.

6.2.2 Real-World Applications

Beyond the trivial application of the research in supporting training models capable of recognising rare conditions and procedures, and providing analysis of how well their

predictions follow the structure of the ontology, the most relevant real-world application of our research comes from evaluation techniques, especially the WHCM. This tool not only provides macro-level analyses, but can also be used to study a model's performance within the scope of a particular family of codes. This is highly relevant for scenarios where a human coder is aided by a machine, with both producing predictions. By having access to family-level data, especially the visualised confusion matrices (as shown in Chapter 3) when presented with a model's prediction, the coder can consider the viability of the predicted labels with the additional information on what errors the model tends to make – given a prediction of a code, how likely is it to be a correct prediction, and, should the prediction be incorrect, which other codes within the same family are likely to be the correct code to assign? Furthermore, WHCM does not need to strictly follow the structure of the ontology – a custom family of codes can be defined in order to study confusion between semantically similar codes across families.

6.2.3 Non-Positive Mentions

The thesis focused on the standard scenario of medical document coding where the assigned code indicates the presence of a condition or a procedure based on positive mentions of concepts relevant to the code within the input document. It is, however, worth considering the alternative scenario of using medical document coding for non-positive mentions. By this we mean mentions that imply normalcy of a given system – *e.g.*, stating that a chest CT scan was *unremarkable*, or stating *normal* HbA1C levels. Such data could be of use in longitudinal studies. Further to this, there is also a case for coding absence, *e.g.*, the absence of adverse drug events or the absence of adverse response to other types of treatment.

A relatively simple way of acquiring data on non-positive mentions would be to retrieve all the relevant concepts of conditions and procedures relating directly to the patient (rather than family history) – regardless of whether they were negated – with the aid of a NER engine, such as SemEHR (Wu et al., 2018) or MedCAT(Kraljevic et al., 2019), translating them into the label space used to code the document (*e.g.*, ICD-10) and designating NER-reported codes which were absent from the gold standard as non-positives.

6.3 Ethics

This section is dedicated to the ethical analysis of the task of automated ICD coding, and the ethical impact of the research presented within this thesis. It focuses on stakeholder groups involved in ICD coding, how automation would affect them, and – where applicable – what steps need to be taken to better cater to their needs.

While on one hand automating ICD coding aims to speed up the process and lead to lowering the workload of coders, depending on the implementation – once performance matches or surpasses humans – the automation of ICD coding may lead to significant transformation or even elimination of the clinical coder position. The general trend within the field seems to aim to implement solutions that aid coders, rather than replace them – the target application not being full automation, but so-called computer-aided-coding (CAC) (Dong et al., 2022a). In order to both help advance the field and steer it towards the CAC solution, clinical coders have been involved in automatic ICD coding research, be it for performance comparison (Kim et al., 2022b), evaluation of interpretability(Kim et al., 2022a), or providing insight on the process of clinical document coding (Dong et al., 2022a). To ensure that the needs of the coders are met, they must further be involved in the research, both to provide a better understanding of the task, and to steer implementation – particularly in human-computer interaction.

Insurance companies act as a stakeholder in healthcare systems that determine insurance payouts based on the assigned ICD codes. This issue is an ethical concern even within manual ICD coding – are documents coded in a way that maximises the correspondence of the assigned codes to the patient's hospital course, or are they trying to maximise the financial return? The medical community as a whole plays an important role as well, as ICD codes assigned to patients can be involved in driving research and decision-making, *e.g.*, through drawing statistics or building patient cohorts. Finally, it is in the interest of the patient to have their documents coded correctly, as these provide a fast reference for their current conditions and medical history. An error in coding may lead to incorrect assumptions, *e.g.*, an incorrectly coded prior condition that may qualify or disqualify them for a treatment/trial, or may be a contraindication for certain procedures/medications. All these stakeholder groups benefit from accurate coding. One way through which the ICD can contribute to improve the situation is by providing more refined codesets. This is the general trend of the ICD, progressively allowing codes to capture more detail, but consequently resulting in larger codesets making the task more difficult to build a machine learning model for. A simple yet radical example

of such refinement is the introduction of laterality within ICD-10 – ICD-9 allows codes for the injury of a lower extremity, but is not designed to state whether it was the left or the right one. In the context of the task of automatic ICD coding this means that – if available – we should use data coded with the most up-to-date standard, providing as detailed predictions as the coding standard allows. Furthermore, how does a model trained on data from one healthcare system adjust to a different system where prevalence of different diseases, methods of writing hospital records, or financial incentives from the insurance providers may differ? The field should involve more evaluation on datasets from a variety of institutions and/or healthcare systems in order to analyse the models' transferrability. While the standard F_1-based global measures only tell us whether the model performs well on new data, a more thorough quantitative and qualitative analysis should be conducted to investigate why this is, as it may point to differences in how coding is handled in different institutions.

Furthermore, the use of Large Language Models for generating synthetic clinical data gives rise to concerns for patient privacy. An LLM can produce a variety of synthetic datapoints given the same input prompt. In the thesis, we have employed this capability in the case of labels associated with few training documents (yielding an insufficient number of input prompts) in generation through the temperature parameter. An LLM may, however, be able to reproduce the data it was trained on. When using an LLM for generating synthetic data meant to correspond to data potentially involving personal information of patients, if at all possible, one should first confirm what data the LLM was trained on and whether it may have been exposed to personal identifiable information. Regardless of whether personal identifiable information was present in training, the generated data should undergo de-identification procedures.

6.4 Future Work

6.4.1 Evaluation Metrics

The ideas implemented in hierarchical evaluation metrics developed as part of this thesis can be further explored in model training – *e.g.*, as the loss function. In their current form, they are merely used in comparing models, and determining error prevalence. As loss functions, they could directly influence the training. A potential means of extending the task would be exploring the possibility of having a model predict how many codes in a given family of codes are to be assigned, effectively turning the concept

behind the CoPHE evaluation metric into a task for the model.

6.4.2 Generation of Clinical Data with Large Language Models

The exploration of generation of discharge summaries was limited to a single LLM – GPT-3.5. Further research could be conducted on LLMs that are either local (with access to the model and the pre-trained weights), more complex models (*e.g.*, GPT-4), or domain-specific (models trained on and intended for producing clinical or biomedical data). Local models could undergo further fine-tuning to be more suited for the task. As the local-model scenario puts the researcher in full control of the model, the training procedure, and does not involve sending data to be processed by a third party, there are fewer information governance concerns (*e.g.*, no patient data is processed through an API or observed by humans for moderation of inputs).

Furthermore, since the release of GPT-3.5, more complex LLMs have been released. An example of such a model is GPT-4. This model was trained on a significantly larger dataset (about 10 times larger). Another advantage of GPT-4 is a larger context window allowing for producing longer outputs or providing further instructions. Ellershaw et al. (2024) in their generation of clinical documents (while not focusing on ICD coding specifically) include the full clinical guidelines for writing such documents. In the space of ICD coding, the relevant coding guidelines could be included within the prompt. GPT-4 is also supposedly better at following instructions, which may reduce undesired behaviours which remained in GPT-3.5's output despite our instruction.

Models fine-tuned on data in the target domain have been shown to lead to better results both in the previous (*e.g.*, BioBERT (Lee et al., 2020)) and current generation (*e.g.*, Gatortron (Peng et al., 2024)) of pre-trained language models. Hence, domain-specific LLMs, or ones able to be further fine-tuned on clinical data in a secure way should be further investigated as generators of additional training data for smaller, more specialised models that can run on limited hospital hardware. Another promising solution in a data-secure setting is to provide real clinical notes within the prompt as examples for in-context learning of an LLM to help guide the generation process (Lewis et al., 2020b). Such guidance could lead to producing more realistic narratives.

Finally, further investigation can be conducted into different prompts and prompting strategies. The synthetic discharge summaries generated by GPT-3.5 suffered from issues that included lack of variety of the relevant medical terminology, incoherent nar-

ratives, and the generated data being insufficiently authentic. These could be addressed by involving further specifications within the prompt (*e.g.*, clinical coding guidelines), inclusion of discharge summary templates (or real clinical notes if the setting permits it), or further processing of the input codes – *e.g.*, proposing different surface forms for concepts based on medical vocabularies or grouping diagnoses and codes that tend to co-occur (*e.g.*, different complications of diabetes) to be presented in the same paragraph. Additional tasks could be included in the prompt to encourage desired behaviour – *e.g.*, ranking the conditions and procedures by severity or importance to the case and dedicating more content to the concepts with higher ranks.

6.4.3 Automatic ICD Coding

Within the context of the task of automatic ICD coding, the thesis highlighted the over-/underprediction issues of the most commonly used neural approaches. Future work should consider investigating means of addressing these issues. The predictions of ICD codes as part of the synthetic data generation guided by a list of input descriptions seemed to be affected by this to a lesser extent than the local models. Unfortunately, this was not replicated within coding real clinical notes. Secondly, more complex approaches to the output layer of LMTC models should be explored. Grivas et al. (2024) show that the most commonly used LMTC sigmoid output layer's assumptions in multi-label classification (including ICD-9 coding in MIMIC-III) result in an exponential number of label combinations that cannot be predicted irrespective of the input. The authors argue for a Discrete Fourier Transform output layer instead to address this issue. Neuro-Symbolic Learning – combining neural network models with symbolic rules – could also be employed in order to account for constraints of mutually exclusive codes or codes that are likely to co-occur (*e.g.*, a condition along with its corresponding treatment). The semantic probabilistic layer (Ahmed et al., 2022) is an example of such a layer with constraints.

6.4.4 Implementation within industry

Automated ICD coding systems are part of efforts within industry. An example of this is research conducted prior to the thesis project with Canon Medical Research Europe (CMRE) (Falis et al., 2019). Our work on rule-based data augmentation was partly inspired by research conducted within CMRE (Schrempf et al., 2021). Over the course of conducting the research presented within this thesis some of the authors discussed

the task of ICD coding with the representatives of startup companies. Our insights on hierarchical evaluation were used in a publication by one of the companies (Kim et al., 2022a). However, rule-based and LLM-based methods for creating more training data were not further discussed with industrial partners. In general, further effort should be made in integrating the content of the thesis into applications – be it through industry or further evaluation on hospital data in the UK.

Appendix A

List of Codes Targeted in Generation with GPT-3.5

E10.3299, E10.3519, E10.3531, E10.3599, E10.52, G43.009, G43.501, G43.919, G43.A0, G43.B0, G43.D0, H35.3110, H35.373, H81.21, H81.399, H81.8X3, H81.91, H81.92, S00.11XA, S00.531A, S00.532A, S00.81XD, S00.93XA, S02.0XXA, S02.113A, S02.119A, S02.31XA, S02.32XA, S02.3XXA, S02.401A, S02.402A, S02.40CA, S02.40DA, S02.40EA, S02.40FA, S02.411A, S02.412A, S02.413A, S02.5XXA, S02.601A, S02.609D, S02.611A, S02.61XA, S02.621A, S02.622A, S02.63XA, S02.652A, S02.66XA, S02.81XA, S02.82XA, S02.8XXA, S06.0X9A, S06.1X0D, S06.2X0A, S06.2X6A, S06.2X7A, S06.300A, S06.339A, S06.359A, S06.5X0D, S06.5X1A, S06.5X7A, S06.5X9D, S06.6X1A, S06.6X6A, S06.890A, S06.9X0S, S06.9X3A, S06.9X9A, T82.03XA, T82.09XA, T82.190A, T82.223A, T82.310A, T82.330A, T82.338A, T82.398D, T82.49XD, T82.594A, T82.6XXA, T82.856A, T82.857A, T82.867A, T82.868S, T84.020A, T84.023A, T84.032A, T84.033A, T84.052A, T84.226A, T84.296A, T84.51XA, T84.53XD, T84.59XA, T84.620D, T84.623A, T84.63XA, T84.7XXA, T84.89XA, T85.02XS, T85.43XA, T85.518A, T85.520A, T85.528A, T85.598A, T85.611A, T85.691A, T85.694A, T85.698A, T85.71XA, T85.72XA, T85.848A, T85.898A, T85.9XXA

Bibliography

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. (2023). Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Afkanpour, A., Adeel, S., Bassani, H., Epshteyn, A., Fan, H., Jones, I., Malihi, M., Nauth, A., Sinha, R., Woonna, S., et al. (2022). Bert for long documents: A case study of automated icd coding. *arXiv preprint arXiv:2211.02519*.

Agrawal, M., Hegselmann, S., Lang, H., Kim, Y., and Sontag, D. (2022). Large language models are few-shot clinical information extractors. *arXiv preprint arXiv:2205.12689*.

Ahmed, K., Teso, S., Chang, K.-W., Van den Broeck, G., and Vergari, A. (2022). Semantic probabilistic layers for neuro-symbolic learning. *Advances in Neural Information Processing Systems*, 35:29944–29959.

Almeida, F. and Xexéo, G. (2019). Word embeddings: A survey. *arXiv preprint arXiv:1901.09069*.

Alonso, V., Santos, J. V., Pinto, M., Ferreira, J., Lema, I., Lopes, F., and Freitas, A. (2020). Problems and barriers during the process of clinical coding: a focus group study of coders' perceptions. *Journal of medical systems*, 44:1–8.

Alsentzer, E., Murphy, J. R., Boag, W., Weng, W.-H., Jin, D., Naumann, T., and McDermott, M. (2019). Publicly available clinical bert embeddings. *arXiv preprint arXiv:1904.03323*.

Amos, L., Anderson, D., Brody, S., Ripple, A., and Humphreys, B. L. (2020). Umls users and uses: a current overview. *Journal of the American Medical Informatics Association*, 27(10):1606–1611.

Anaby-Tavor, A., Carmeli, B., Goldbraich, E., Kantor, A., Kour, G., Shlomov, S., Tepper, N., and Zwerdling, N. (2020). Do not have enough data? deep learning to the rescue! In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7383–7390.

Antoniou, A., Storkey, A., and Edwards, H. (2017). Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*.

Aronson, A. R. (2001). Effective mapping of biomedical text to the UMLS metathesaurus: the MetaMap program. In *Proceedings of the AMIA Symposium*, page 17. American Medical Informatics Association.

Aronson, A. R. and Lang, F.-M. (2010). An overview of metamap: historical perspective and recent advances. *Journal of the American Medical Informatics Association*, 17(3):229–236.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Barros, J., Rojas, M., Dunstan, J., and Abeliuk, A. (2022). Divide and conquer: An extreme multi-label classification approach for coding diseases and procedures in spanish. In *Proceedings of the 13th International Workshop on Health Text Mining and Information Analysis (LOUHI)*, pages 138–147.

Bidgood Jr, W. D., Horii, S. C., Prior, F. W., and Van Syckle, D. E. (1997). Understanding and using dicom, the data interchange standard for biomedical imaging. *Journal of the American Medical Informatics Association*, 4(3):199–212.

Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford university press.

Bodenreider, O., Cornet, R., and Vreeman, D. J. (2018). Recent developments in clinical terminologies—snomed ct, loinc, and rxnorm. *Yearbook of medical informatics*, 27(01):129–139.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Cartwright, D. J. (2013). Icd-9-cm to icd-10-cm codes: what? why? how?

Chalkidis, I., Fergadiotis, E., Malakasiotis, P., and Androutsopoulos, I. (2019a). Large-scale multi-label text classification on eu legislation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6314–6322.

Chalkidis, I., Fergadiotis, M., Kotitsas, S., Malakasiotis, P., Aletras, N., and Androutsopoulos, I. (2020). An empirical study on large-scale multi-label text classification including few and zero-shot labels. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7503–7515.

Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., and Androutsopoulos, I. (2019b). Extreme multi-label legal text classification: A case study in eu legislation. *arXiv preprint arXiv:1905.10892*.

Chapman, W. W., Bridewell, W., Hanbury, P., Cooper, G. F., and Buchanan, B. G. (2001). A simple algorithm for identifying negated findings and diseases in discharge summaries. *Journal of biomedical informatics*, 34(5):301–310.

Cocci, A., Pezzoli, M., Lo Re, M., Russo, G. I., Asmundo, M. G., Fode, M., Cacciamani, G., Cimino, S., Minervini, A., and Durukan, E. (2024). Quality of information and appropriateness of chatgpt outputs for urology patients. *Prostate cancer and prostatic diseases*, 27(1):103–108.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dinwoodie, H. and Howell, R. (1973). Automatic disease coding: the'fruit-machine'method in general practice. *British journal of preventive & social medicine*, 27(1):59.

Dong, H., Falis, M., Whiteley, W., Alex, B., Matterson, J., Ji, S., Chen, J., and Wu, H. (2022a). Automated clinical coding: what, why, and where we are? *NPJ digital medicine*, 5(1):159.

Dong, H., Suárez-Paniagua, V., Whiteley, W., and Wu, H. (2021a). Explainable automated coding of clinical notes using hierarchical label-wise attention networks and label embedding initialisation. *Journal of biomedical informatics*, 116:103728.

Dong, H., Suárez-Paniagua, V., Zhang, H., Wang, M., Whitfield, E., and Wu, H. (2021b). Rare disease identification from clinical notes with ontologies and weak supervision. In *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 2294–2298. IEEE.

Dong, H., Suárez-Paniagua, V., Whiteley, W., and Wu, H. (2021c). Explainable automated coding of clinical notes using hierarchical label-wise attention networks and label embedding initialisation. *Journal of Biomedical Informatics*, 116:103728.

Dong, Q., Li, L., Dai, D., Zheng, C., Wu, Z., Chang, B., Sun, X., Xu, J., and Sui, Z. (2022b). A survey on in-context learning. *arXiv preprint arXiv:2301.00234*.

Dorr, D. A., Phillips, W., Phansalkar, S., Sims, S. A., and Hurdle, J. F. (2006). Assessing the difficulty and time cost of de-identification in clinical narratives. *Methods of information in medicine*, 45(03):246–252.

Edin, J., Junge, A., Havtorn, J. D., Borgholt, L., Maistro, M., Ruotsalo, T., and Maaløe, L. (2023). Automated medical coding on mimic-iii and mimic-iv: A critical review and replicability study. *arXiv preprint arXiv:2304.10909*.

Ellershaw, S., Tomlinson, C., Burton, O. E., Frost, T., Hanrahan, J. G., Khan, D. Z., Horsfall, H. L., Little, M., Malgapo, E., Starup-Hansen, J., et al. (2024). Automated generation of hospital discharge summaries using clinical guidelines and large language models. In *AAAI 2024 Spring Symposium on Clinical Foundation Models*.

Eyler, J. M. (1979). Victorian social medicine: The ideas and methods of william farr.

Falis, M., Dong, H., Birch, A., and Alex, B. (2021). CoPHE: A count-preserving hierarchical evaluation metric in large-scale multi-label text classification. In *2021 Conference on Empirical Methods in Natural Language Processing*.

Falis, M., Dong, H., Birch, A., and Alex, B. (2022). Horses to zebras: ontology-guided data augmentation and synthesis for icd-9 coding. In *Proceedings of the 21st Workshop on Biomedical Language Processing*. Association for Computational Linguistics.

Falis, M., Gema, A. P., Dong, H., Daines, L., Basetti, S., Holder, M., Penfold, R. S., Birch, A., and Alex, B. (2024). Can gpt-3.5 generate and code discharge summaries? *Journal of the American Medical Informatics Association*, 31(10):2284–2293.

Falis, M., Pajak, M., Lisowska, A., Schrempf, P., Deckers, L., Mikhael, S., Tsaftaris, S., and O'Neil, A. (2019). Ontological attention ensembles for capturing semantic concepts in icd code prediction from clinical text. In *Proceedings of the Tenth International Workshop on Health Text Mining and Information Analysis (LOUHI 2019)*, pages 168–177.

Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Fung, K. W., Xu, J., and Bodenreider, O. (2020). The new international classification of diseases 11th edition: a comparative analysis with icd-10 and icd-10-cm. *Journal of the American Medical Informatics Association*, 27(5):738–746.

Ghosh, S., Evuru, C. K., Kumar, S., Ramaneswaran, S., Sakshi, S., Tyagi, U., and Manocha, D. (2023). Dale: Generative data augmentation for low-resource legal nlp. *arXiv preprint arXiv:2310.15799*.

Görtler, J., Hohman, F., Moritz, D., Wongsuphasawat, K., Ren, D., Nair, R., Kirchner, M., and Patel, K. (2021). Neo: Generalizing confusion matrix visualization to hierarchical and multi-output labels. *arXiv preprint arXiv:2110.12536*.

Grimm, S. (2009). Knowledge representation and ontologies. In *Scientific data mining and knowledge discovery: principles and foundations*, pages 111–137. Springer.

Grivas, A., Alex, B., Grover, C., Tobin, R., and Whiteley, W. (2020). Not a cute stroke: analysis of rule-and neural network-based information extraction systems for brain radiology reports. In *Proceedings of the 11th international workshop on health text mining and information analysis*, pages 24–37.

Grivas, A., Vergari, A., and Lopez, A. (2024). Taming the sigmoid bottleneck: Provably argmaxable sparse multi-label classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 12208–12216.

Grover, A. and Leskovec, J. (2016). node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864.

Gu, Y., Tinn, R., Cheng, H., Lucas, M., Usuyama, N., Liu, X., Naumann, T., Gao, J., and Poon, H. (2021). Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23.

Harrison, P. W., Amode, M. R., Austine-Orimoloye, O., Azov, A. G., Barba, M., Barnes, I., Becker, A., Bennett, R., Berry, A., Bhai, J., et al. (2024). Ensembl 2024. *Nucleic Acids Research*, 52(D1):D891–D899.

Heydarian, M., Doyle, T. E., and Samavi, R. (2022). MLCM: Multi-label confusion matrix. *IEEE Access*, 10:19083–19095.

Hirsch, J., Nicola, G., McGinty, G., Liu, R., Barr, R., Chittle, M., and Manchikanti, L. (2016). Icd-10: history and context. *American Journal of Neuroradiology*, 37(4):596–599.

Huang, C.-W., Tsai, S.-C., and Chen, Y.-N. (2022a). Plm-icd: automatic icd coding with pretrained language models. *arXiv preprint arXiv:2207.05289*.

Huang, C.-W., Tsai, S.-C., and Chen, Y.-N. (2022b). Plm-icd: automatic icd coding with pretrained language models. *arXiv preprint arXiv:2207.05289*.

Humphreys, B. L. and Lindberg, D. A. (1989). Building the unified medical language system. In *Proceedings of the Annual Symposium on Computer Application in Medical Care*, page 475. American Medical Informatics Association.

Johnson, A. E., Bulgarelli, L., Shen, L., Gayles, A., Shammout, A., Horng, S., Pollard, T. J., Moody, B., Gow, B., Lehman, L.-w. H., et al. (2023). Mimic-iv, a freely accessible electronic health record dataset. *Scientific data*, 10(1):1.

Johnson, A. E., Pollard, T. J., Shen, L., Lehman, L.-w. H., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Anthony Celi, L., and Mark, R. G. (2016). Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9.

Jurafsky, D. (2000). *Speech & language processing*. Pearson Education India.

Kang, T., Perotte, A., Tang, Y., Ta, C., and Weng, C. (2021). UMLS-based data augmentation for natural language processing of clinical research literature. *Journal of the American Medical Informatics Association*, 28(4):812–823.

Kim, B.-H., Deng, Z., Yu, P. S., and Ganapathi, V. (2022a). Can current explainability help provide references in clinical notes to support humans annotate medical codes? *arXiv preprint arXiv:2210.15882*.

Kim, B.-H. and Ganapathi, V. (2021). Read, attend, and code: pushing the limits of medical codes prediction from clinical notes by machines. In *Machine Learning for Healthcare Conference*, pages 196–208. PMLR.

Kim, D., Yoo, H., and Kim, S. (2022b). An automatic icd coding network using partition-based label attention. *arXiv preprint arXiv:2211.08429*.

Köhler, S., Gargano, M., Matentzoglu, N., Carmody, L. C., Lewis-Smith, D., Vasilevsky, N. A., Danis, D., Balagura, G., Baynam, G., Brower, A. M., et al. (2021). The human phenotype ontology in 2021. *Nucleic acids research*, 49(D1):D1207–D1217.

Kosmopoulos, A., Partalas, I., Gaussier, E., Paliouras, G., and Androutsopoulos, I. (2015). Evaluation measures for hierarchical classification: a unified view and novel approaches. *Data Mining and Knowledge Discovery*, 29(3):820–865.

Kovačević, A., Bašaragin, B., Milošević, N., and Nenadić, G. (2024). De-identification of clinical free text using natural language processing: A systematic review of current approaches. *Artificial Intelligence in Medicine*, page 102845.

Kraljevic, Z., Bean, D., Mascio, A., Roguski, L., Folarin, A., Roberts, A., Bendayan, R., and Dobson, R. (2019). Medcat–medical concept annotation tool. *arXiv preprint arXiv:1912.10166*.

Kraljevic, Z., Bean, D., Shek, A., Bendayan, R., Yeung, J. A., Deng, A., Baston, A., Ross, J., Idowu, E., Teo, J. T., et al. (2022). Foresight-deep generative modelling of patient timelines using electronic health records. *CoRR*.

Kumar, V. B., Srinivasan, A., Chaudhary, A., Route, J., Mitamura, T., and Nyberg, E. (2019). Dr. quad at mediqa 2019: Towards textual inference and question entailment using contextualized representations. *arXiv preprint arXiv:1907.10136*.

Lecler, A., Duron, L., and Soyer, P. (2023). Revolutionizing radiology with gpt-based models: Current applications, future possibilities and limitations of chatgpt. *Diagnostic and Interventional Imaging*.

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., and Kang, J. (2020). Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Lee, P., Bubeck, S., and Petro, J. (2023). Benefits, limits, and risks of gpt-4 as an ai chatbot for medicine. *New England Journal of Medicine*, 388(13):1233–1239.

Lewis, P., Ott, M., Du, J., and Stoyanov, V. (2020a). Pretrained language models for biomedical and clinical tasks: understanding and extending the state-of-the-art. In *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, pages 146–157.

Lewis, P. S. H., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., and Kiela, D. (2020b). Retrieval-augmented generation for knowledge-intensive NLP tasks. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Li, B., Hou, Y., and Che, W. (2022). Data augmentation approaches in natural language processing: A survey. *Ai Open*, 3:71–90.

Li, F. and Yu, H. (2020a). Icd coding from clinical text using multi-filter residual convolutional neural network. In *proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8180–8187.

Li, F. and Yu, H. (2020b). Icd coding from clinical text using multi-filter residual convolutional neural network. In *proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8180–8187.

Lin, C.-Y. and Och, F. (2004). Looking for a few good metrics: Rouge and its evaluation. In *Ntcir workshop*.

Lipscomb, C. E. (2000). Medical subject headings (mesh). *Bulletin of the Medical Library Association*, 88(3):265.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Lo, C. K. (2023). What is the impact of chatgpt on education? a rapid review of the literature. *Education Sciences*, 13(4):410.

Maharana, K., Mondal, S., and Nemade, B. (2022). A review: Data pre-processing and data augmentation techniques. *Global Transitions Proceedings*, 3(1):91–99.

McAuley, J. and Leskovec, J. (2013). Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 165–172.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013a). Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Mikolov, T., Yih, W.-t., and Zweig, G. (2013b). Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pages 746–751.

Miranda-Escalada, A., Gonzalez-Agirre, A., and Krallinger, M. (2020). Codiesp corpus: gold standard spanish clinical cases coded in icd10 (cie10)-ehealth clef2020. *Zenodo*.

Mullenbach, J., Wiegreffe, S., Duke, J., Sun, J., and Eisenstein, J. (2018). Explainable prediction of medical codes from clinical text. In *NAACL-HLT*.

Nguyen, T.-T., Schlegel, V., Kashyap, A., Winkler, S., Huang, S.-S., Liu, J.-J., and Lin, C.-J. (2023). Mimic-iv-icd: A new benchmark for extreme multilabel classification. *arXiv preprint arXiv:2304.13998*.

Ollagnier, A. and Williams, H. T. (2020). Text augmentation techniques for clinical case classification. In *CLEF (Working Notes)*.

O'Shea, K. and Nash, R. (2015). An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P. F., Leike, J., and Lowe, R. (2022). Training language models to follow instructions with human feedback. In

Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A., editors, *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Peng, C., Yang, X., Chen, A., Smith, K. E., PourNejatian, N., Costa, A. B., Martin, C., Flores, M. G., Zhang, Y., Magoc, T., et al. (2023). A study of generative large language model for medical research and healthcare. *NPJ Digital Medicine*, 6(1):210.

Peng, C., Yang, X., Lyu, M., Smith, K. E., Costa, A., Flores, M. G., Bian, J., and Wu, Y. (2024). Gatortron and gatortrongpt: Large language models for clinical narratives. In *AAAI 2024 Spring Symposium on Clinical Foundation Models*.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Ren, W., Zeng, R., Wu, T., Zhu, T., and Krishnan, R. G. (2022). Hicu: Leveraging hierarchy for curriculum learning in automated icd coding. *arXiv preprint arXiv:2208.02301*.

Rios, A. and Kavuluru, R. (2018a). Emr coding with semi–parametric multi–head matching networks. In *Proceedings of the conference. Association for Computational Linguistics. North American Chapter. Meeting*, volume 2018, page 2081. NIH Public Access.

Rios, A. and Kavuluru, R. (2018b). Few-shot and zero-shot multi-label learning for structured label spaces. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, volume 2018, page 3132. NIH Public Access.

Rios, A. and Kavuluru, R. (2019). Neural transfer learning for assigning diagnosis codes to emrs. *Artificial intelligence in medicine*, 96:116–122.

Sadoughi, N., Finley, G. P., Fone, J., Murali, V., Korenevski, M., Baryshnikov, S., Axtmann, N., Miller, M., and Suendermann-Oeft, D. (2018). Medical code prediction with multi-view convolution and description-regularized label-dependent attention. *arXiv preprint arXiv:1811.01468*.

Sallam, M., Barakat, M., and Sallam, M. (2023). Pilot testing of a tool to standardize the assessment of the quality of health information generated by artificial intelligence-based models. *Cureus*, 15(11).

Sardanelli, F. (2017). Trends in radiology and experimental research.

Savova, G. K., Masanz, J. J., Ogren, P. V., Zheng, J., Sohn, S., Kipper-Schuler, K. C., and Chute, C. G. (2010). Mayo clinical text analysis and knowledge extraction system (ctakes): architecture, component evaluation and applications. *Journal of the American Medical Informatics Association*, 17(5):507–513.

Schrempf, P., Watson, H., Park, E., Pajak, M., MacKinnon, H., Muir, K. W., Harris-Birtill, D., and O'Neil, A. Q. (2021). Templated text synthesis for expert-guided multi-label extraction from radiology reports. *Machine Learning and Knowledge Extraction*, 3(2):299–317.

Searle, T., Ibrahim, Z., and Dobson, R. (2020). Experimental evaluation and development of a silver-standard for the mimic-iii clinical coding dataset. In *Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing*, pages 76–85.

Sennrich, R., Haddow, B., and Birch, A. (2015a). Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Sennrich, R., Haddow, B., and Birch, A. (2015b). Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Shin, H.-C., Zhang, Y., Bakhturina, E., Puri, R., Patwary, M., Shoeybi, M., and Mani, R. (2020). Biomegatron: larger biomedical domain language model. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4700–4706.

Singhal, K., Azizi, S., Tu, T., Mahdavi, S. S., Wei, J., Chung, H. W., Scales, N., Tanwani, A., Cole-Lewis, H., Pfohl, S., et al. (2023a). Large language models encode clinical knowledge. *Nature*, 620(7972):172–180.

Singhal, K., Tu, T., Gottweis, J., Sayres, R., Wulczyn, E., Hou, L., Clark, K., Pfohl, S., Cole-Lewis, H., Neal, D., et al. (2023b). Towards expert-level medical question answering with large language models. *arXiv preprint arXiv:2305.09617*.

Song, C., Zhang, S., Sadoughi, N., Xie, P., and Xing, E. (2021). Generalized zero-shot text classification for icd coding. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 4018–4024.

Spackman, K. A., Campbell, K. E., and Côté, R. A. (1997). Snomed rt: a reference terminology for health care. In *Proceedings of the AMIA annual fall symposium*, page 640. American Medical Informatics Association.

Stanfill, M. H., Williams, M., Fenton, S. H., Jenders, R. A., and Hersh, W. R. (2010). A systematic literature review of automated clinical coding and classification systems. *Journal of the American Medical Informatics Association*, 17(6):646–651.

Tan, P.-N., Steinbach, M., and Kumar, V. (2019). *Introduction to Data Mining, (Second Edition, Global Edition)*. Pearson Education Limited, Harlow, United Kingdom.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. (2023). Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Trabucco, B., Doherty, K., Gurinas, M., and Salakhutdinov, R. (2023). Effective data augmentation with diffusion models. *arXiv preprint arXiv:2302.07944*.

Vasant, D., Chanas, L., Malone, J., Hanauer, M., Olry, A., Jupp, S., Robinson, P. N., Parkinson, H., and Rath, A. (2014). Ordo: an ontology connecting rare disease, epidemiology and genetic data. In *Proceedings of ISMB*, volume 30. researchgate. net.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Vu, T., Nguyen, D. Q., and Nguyen, A. (2020a). A label attention model for icd coding from clinical text. *arXiv preprint arXiv:2007.06351*.

Vu, T., Nguyen, D. Q., and Nguyen, A. (2020b). A label attention model for icd coding from clinical text. *arXiv preprint arXiv:2007.06351*.

Wang, S., Tang, D., Zhang, L., Li, H., and Han, D. (2022). Hienet: Bidirectional hierarchy framework for automated icd coding. In *International Conference on Database Systems for Advanced Applications*, pages 523–539. Springer.

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., et al. (2022). Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Weinreich, S. S., Mangon, R., Sikkens, J., Teeuw, M. E., and Cornel, M. (2008). Orphanet: a european database for rare diseases. *Nederlands tijdschrift voor geneeskunde*, 152(9):518–519.

Weiss, K., Khoshgoftaar, T. M., and Wang, D. (2016). A survey of transfer learning. *Journal of Big data*, 3:1–40.

Wiegreffe, S., Choi, E., Yan, S., Sun, J., and Eisenstein, J. (2019). Clinical concept extraction for document-level coding. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 261–272.

Wiegreffe, S. and Pinter, Y. (2019). Attention is not not explanation. *arXiv preprint arXiv:1908.04626*.

Wu, H., Toti, G., Morley, K. I., Ibrahim, Z. M., Folarin, A., Jackson, R., Kartoglu, I., Agrawal, A., Stringer, C., Gale, D., et al. (2018). SemEHR: A general-purpose semantic search system to surface semantic data from clinical notes for tailored care, trial recruitment, and clinical research. *Journal of the American Medical Informatics Association*, 25(5):530–537.

Wu, H., Wang, M., Wu, J., Francis, F., Chang, Y.-H., Shavick, A., Dong, H., Poon, M. T., Fitzpatrick, N., Levine, A. P., et al. (2022). A survey on clinical natural language processing in the united kingdom from 2007 to 2022. *NPJ digital medicine*, 5(1):186.

Xian, Y., Schiele, B., and Akata, Z. (2017). Zero-shot learning-the good, the bad and the ugly. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4582–4591.

Yang, X., Chen, A., PourNejatian, N., Shin, H. C., Smith, K. E., Parisien, C., Compas, C., Martin, C., Flores, M. G., Zhang, Y., et al. (2022). Gatortron: a large clinical language model to unlock patient information from unstructured electronic health records. *arXiv preprint arXiv:2203.03540*.

Yeung, J. A., Kraljevic, Z., Luintel, A., Balston, A., Idowu, E., Dobson, R. J., and Teo, J. T. (2023). Ai chatbots not yet ready for clinical use. *medRxiv*, pages 2023–03.

Zhang, D., Li, T., Zhang, H., and Yin, B. (2020a). On data augmentation for extreme multi-label classification. *arXiv preprint arXiv:2009.10778*.

Zhang, Z., Liu, J., and Razavian, N. (2020b). Bert-xml: Large scale automated icd coding using bert pretraining. In *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, pages 24–34.

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., et al. (2023). A survey of large language models. *arXiv preprint arXiv:2303.18223*.