TÉCNICO LISBOA

Accurate and Well-Calibrated ICD Code Assignment with a Chunk-Based Classifier Attending over Diverse Label Embeddings

Gonçalo Emanuel Cavaco Gomes

Thesis to obtain the Master of Science Degree in

Data Science and Engineering

Supervisor(s): Prof. Bruno Emanuel da Graça Martins

Examination Committee

Chairperson: Prof. Chrysoula Zerva
Supervisor: Prof. Bruno Emanuel da Graça Martins
Member of the Committee: Prof. Maria do Rosário De Oliveira Silva

November 2023

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgements

This dissertation is the culmination of not only a semester of hard work, but also the reflection of years of dedication throughout my bachelor and master degree. I want to thank all the professors of Instituto Superior Tecnico (IST) that shared with me all the knowledge and support throughout my academic years. I want to thank the Instituto Superior Técnico (IST) for allowing me to embark on a mobility program to Switzerland, which I consider to have been one of my most profound and enriching academic experiences, which completely changed my academic path, and which I will carry with me very fondly for the rest of my life. I want to express great words of gratitude to my supervisors, Professor Bruno Martins and Engineer Isabel Coutinho. I want to express my sincere acknowledgment to them for allowing me to delve into a field that I am particularly passionate about. The support, knowledge and experience they so kindly shared with me are elements I will carry with me for the rest of my life. The last few months have been a period of intense learning and development, not only academically but also personally. I manifest also my sincere gratitude to INESC-ID to support me throughout this dissertation. I want to thank all my friends, whose constant presence over the years has been immeasurable. For all your friendship and support, I express my gratitude. Specially, I want to manifest my acknowledgements to my closest friends at IST, *Hugo Matias*, *José Reis*, *João Luzio*, and *Pedro Taborda*, for their companionship through thick and thin, making this academic journey a highway worth driving all night long. I want to extend a special and heartfelt thanks to my girlfriend and high school friends, whom I hold dear as my chosen family. In particular, I want to acknowledge *Sofia Anibal*, *Daniel Silva*, *Diogo Saavedra*, *Nuno Matos*, *Pedro Cavaco* and *Victor Almeida*. Their unwavering emotional support and strong belief in me have enabled me to overcome the most challenging moments, and the joyful and memorable times we shared will stay with me for a lifetime. This journey would definitely not have been possible without all of you and i will forever cherish the wonderful memories we have created together. Lastly, I would like to express my deepest gratitude to my parents, brother and the rest of my family. I fully recognize that this gesture of appreciation is insufficient in the face of everything you have done for me. Your words of comfort in times of frustration and worry, as well as your genuine joy and pride in my modest victories, do not go unnoticed. Your unceasing concern, protection and love have been the strength that has enabled me to overcome every obstacle. To all of you, a heartfelt thank you.

Resumo

A Classificação Internacional de Doenças (CID) foi adoptada em larga escala no domínio dos cuidados de saúde, por exemplo, para resumir as principais informações em documentos clínicos. Uma vez que a codificação manual da CID é dispendiosa, demorada e propensa a erros, foram propostos algoritmos de aprendizagem profunda para automatizar esta tarefa, tratando o problema como uma tarefa de classificação de texto com vários códigos. Trabalhos anteriores identificaram vários desafios relacionados com esta tarefa, incluindo o grande número de classes utilizados para a classificação, a necessidade de lidar com textos longos e a distribuição desequilibrada de rótulos. Estes desafios exigem técnicas de modelação avançadas, e a utilização comum de modelos de linguagem pré-treinados baseados em *Transformers* demonstrou ter um desempenho inferior nesta tarefa em comparação com arquitecturas especializadas. Este documento descreve uma abordagem estado de arte para a codificação de CDI, combinando um modelo de transformador pré-treinado com várias ideias de trabalhos anteriores relacionados, obtendo um desempenho superior ao de modelos anteriores e mostrando que a utilização de modelos de linguagem pré-treinados baseados em transformadores é uma abordagem válida e viável neste domínio. Estas incluem: (a) o processamento de documentos longos em segmentos, que são depois processadas com uma operação de ”max-pooling”, com base na ideia de que a existência de um único ”chunk”que descreva um determinado diagnóstico ou procedimento é prova suficiente para a atribuição do respectivo código no documento; (b) a utilização de um mecanismo de incorporação de classes que explora diversos sinónimos de códigos da CID, (c) uma segunda etapa de treino utilizando uma taxa de aprendizagem mais baixa e funções de perda especificamente adaptadas para incluir uma estimativa da da distribuição de cada classe, que, ao mesmo tempo, demonstrou ter a vantagem de melhorar a calibração do modelo, e (d) o pós-processamento dos resultados da classificação com um modelo MLP que estima a frequência relativa dos códigos da CID com base num conjunto de texto de registos clínicos, capturando simultaneamente as correlações entre códigos. Os aspectos (c) e (d) da enumeração anterior mereceram uma atenção especial, uma vez que considerámos não só a classificação de instâncias individuais, mas também o problema da quantificação do texto, que diz respeito à estimativa da prevalência (ou seja, a frequência relativa) dos códigos CID num determinado conjunto de instâncias num modelo único, unificado e bem calibrado. Os modelos corretamente calibrados são importantes para aplicações como a quantificação de texto, uma vez que estes métodos podem explorar estimativas precisas de probabilidade por classe e associações existentes entre os diferentes códigos da CID. Os resultados experimentais em diferentes divisões do conjunto de dados MIMIC-III mostram que a abordagem proposta atinge os melhores resultados de classificação até à data, ao mesmo tempo que produz estimativas de probabilidade bem calibradas que podem apoiar uma quantificação exacta de conjuntos de texto.

Palavras-chave: Codificação CID Automática, Notas de Alta Hospitalar, Aprendizagem com Redes Profundas, Processamento de Linguagem Natural, Classificação *Multi-Label*, *Pre-trained Language Models*, *Transformers*, Quantificação, MIMIC-III, Calibração de Modelos de Classificação

Abstract

The International Classification of Diseases (ICD) has been adopted worldwide in healthcare, e.g., to summarize the key information in clinical documents. Since manual ICD coding is expensive, time-consuming, and error-prone, deep learning algorithms have been proposed to automate this task, treating the problem as a multi-label text classification task. Prior work has identified several challenges in connection to this task, including the large number of labels used for classification, the need for handling long texts, and the imbalanced label distribution. These challenges require advanced modeling techniques, and the standard use of pre-trained language models based on Transformers has been shown to underperform on this task compared to specialized architectures. This paper describes a state-of-the-art approach for ICD coding and ICD quantification, combining a pre-trained transformer model with several ideas from previous related work, achieving superior performance to previous models and showing that using pre-trained language models based on Transformers is a valid approach under this domain. These include (a) the processing of long documents into chuck representations, which are then processed with a max-pooling operation under the idea that having a single chunk describing a given diagnosis or procedure is enough evidence for the assignment of a label to a document, (b) the use of a label embedding mechanism that explores diverse ICD code synonyms, (c) a second training step using a lower learning rate and loss functions specifically tailored to include a prevalence estimation of each class, which at the same time showed to have the benefits of improving model calibration, and (d) post-processing the classification results with an MLP model that estimates the prevalence of the ICD codes given a text set of clinical records while capturing label correlations. Aspects (c) and (d) from the previous enumeration deserved particular attention since we considered not only individual instance classification but also the problem of text quantification, which concerns estimating the prevalence (i.e., the relative frequency) of ICD codes in a given set of instances, in a single unified well calibrated model. Properly calibrated models are significant for applications like text quantification since these methods can explore accurate per-class probability estimates and existing associations between the different ICD codes. Experimental results on different splits of the MIMIC-III dataset show that the proposed approach achieves state-of-the-art classification results while outputting well-calibrated probability estimates that can support accurate text quantification.

Keywords: Automatic ICD Coding, Hospital Discharge Summaries, Deep Learning, Natural Language Processing, Multi-Label Classification, Pre-trained Language Models,Transformers, Quantification, MIMIC-III, Model Calibration

Dedication

I dedicate this dissertation to my beloved grandmother, who recently passed away.
Forever in my heart, *Maria Dolores Refacho Lascas Cavaco*.

Psalm [91:1]

Whoever dwells in the shelter of the Most High
will rest in the shadow of the Almighty.

Psalm [77:19]

Your path led through the sea,
your way through the mighty waters,
though your footprints were not seen.

Footprints in the Sand

One night I dreamed a dream.
As I was walking along the beach with my Lord.
Across the dark sky flashed scenes from my life.
For each scene, I noticed two sets of footprints in the sand,
One belonging to me and one to my Lord.
After the last scene of my life flashed before me,
I looked back at the footprints in the sand.
I noticed that at many times along the path of my life,
especially at the very lowest and saddest times,
there was only one set of footprints.
This really troubled me, so I asked the Lord about it.
"Lord, you said once I decided to follow you,
You'd walk with me all the way.
But I noticed that during the saddest and most troublesome times of my life,
there was only one set of footprints.
I don't understand why, when I needed You the most, You would leave me."
He whispered, "My precious child, I love you and will never leave you
Never, ever, during your trials and testings.
When you saw only one set of footprints,
It was then that I carried you."

Contents

List of Tables

List of Figures

Acronyms

ADAM Adaptive Moment Estimation. 10, 35

AE Absolute Error. 33

AI Artificial Intelligence. 10

AUC Area Under the Curve. 32

BERT Bidirecional Encoder Representations from Transformers. 12, 14

BM Base Model. 35, 36, 37, 38, 41, 43, 45

BOW Bag-Of-Words. 13

CAML Convolutional Attention for Multi-Label. 17, 18

CBOW Continuous Bag-Of-Words. 13

CC Classify and Count. 3, 14, 41

CLQ Jointly Classify & Quantify Model. 35, 36, 37, 38, 41, 43, 45, 46

CM Clinical Modification. xiii, xv, 3, 5, 6, 7, 8, 29, 30, 31, 38, 43, 45, 46

CNN Convolutional Neural Network. 17, 18

EHR Electronic Health Record. 1

EMR Electronic Medical Record. 19

GPT Generative Pre-trained Transformer. 14

GRU Gated Recurrent Unit. 16

ICD International Classification of Diseases. xiii, xv, 1, 2, 3, 4, 5, 6, 7, 8, 14, 15, 16, 17, 18, 19, 20, 22, 23, 24, 27, 29, 30, 31, 32, 33, 37, 38, 40, 41, 42, 43, 44, 45, 46

LAAT A Label Attention Model for ICD Coding. 18

LSTM Long-Short Term Memory. 13, 16, 18, 19

MAE Mean Absolute Error. 33

MECE Mean Expected Calibration Error. xiii, 33, 37, 43, 45

MIMIC Medical Information Mart for Intensive Care. xiii, xv, 3, 4, 5, 6, 17, 18, 19, 20, 22, 29, 30, 31, 32, 34, 35, 36, 37, 41, 42, 43, 45, 46

MLP Multi-Layer Perceptron. 5, 9, 31, 34, 41, 44, 45, 46

MRAE Mean Relative Absolute Error. 33

MSAM Multiple Synonyms Attention Mechanism. 35, 36, 37, 38, 41, 43, 44, 45

MSMN Multiple Synonyms Matching Network. 19, 20

NCHS National Center for Health Statistics. 6

NLP Natural Language Processing. 3, 11, 12, 13, 14, 15, 16, 19, 44

NMF Non-negative Matrix Factorization. 17

PCC Probabilistic Classify and Count. 14, 27, 41, 43, 45, 46

PLM Pre-trained Language Model. 5, 19, 20, 37

RAE Relative Absolute Error. 33

RNN Recurrent Neural Network. 11, 13, 18

SGD Stochastic Gradient Descent. 9, 10

SVM Support Vector Machine. 15

UMLS Unified Medical Language System. 4, 19, 27

WHO World Health Organization. 1, 6

Chapter 1

Introduction

In the ever-evolving healthcare landscape, the widespread adoption of Electronic Health Record (EHR) has transformed how patient information is recorded and managed (Mosby, 2006). These digital repositories store a wealth of data comprising structured information, such as laboratory results and diagnoses, and unstructured clinical narratives, including progress notes and hospital discharge summaries. This abundance of information has opened up new avenues for analyzing patient data and leveraging it to enhance clinical decision support systems.

Central to the effective organization and analysis of healthcare data is the ICD[1] coding system. The ICD, proposed by World Health Organization (WHO), has emerged as a widely adopted standard by physicians and other healthcare providers, for accurately documenting diagnoses and procedures in the medical domain. By adhering to a consistent framework of ICD codes, healthcare providers can facilitate essential processes such as epidemiological studies, billing and predictive modeling of patient conditions (O'malley et al., 2005). Figure 1.1 shows an example for ICD coding, in which a clinical note is associated with a set of ICD codes.

The manual assignment of ICD codes to clinical text is a challenging task often time-consuming, error-prone and extremely expensive for a healthcare facility. Medical coders face several hurdles during this process due to rigid guidelines and conventions. (O'malley et al., 2005; Nguyen et al., 2018).

As illustrated in Figure 1.2, the ICD codes follow a hierarchical arrangement, wherein the upper-tier codes encompass broad disease categories, while the lower-tier codes pertain to more precise ailments. It is a prevalent occurrence for medical coders to either choose inaccurate sub-types for a specific disease due to the subtle distinctions between these sub-types or to assign an overly generalized ICD code instead of a more detailed alternative (a practice known as undercoding). Additionally, clinical notes incorporate abbreviations and synonyms, leading to ambiguities and misinterpretations during coders' assignment of ICD codes. Lastly, it is essential to note that not all diagnosis descriptions possess a direct one-to-one correspondence with specific ICD codes. Multiple closely related diagnosis descriptions must be linked to a single ICD code in many instances, potentially leading to an unbundling error if physicians code each disease separately.

The limitations mentioned above have driven the exploration of automated approaches using machine learning and deep learning algorithms to streamline the assignment of ICD codes to clinical text. However, despite the advancements in the field, the task of automating the ICD coding still presents several challenges up to date. In terms of medical note representation, clinical notes consist of extensive text narratives encompassing a vast medical vocabulary, posing difficulties for neural network models in effectively encoding and extracting critical information. Additionally, the medical coding system poses its own set of challenges. The label space for codes is highly dimensional and spare, with many avail-

[1] https://www.who.int/standards/classifications/classification-of-diseases

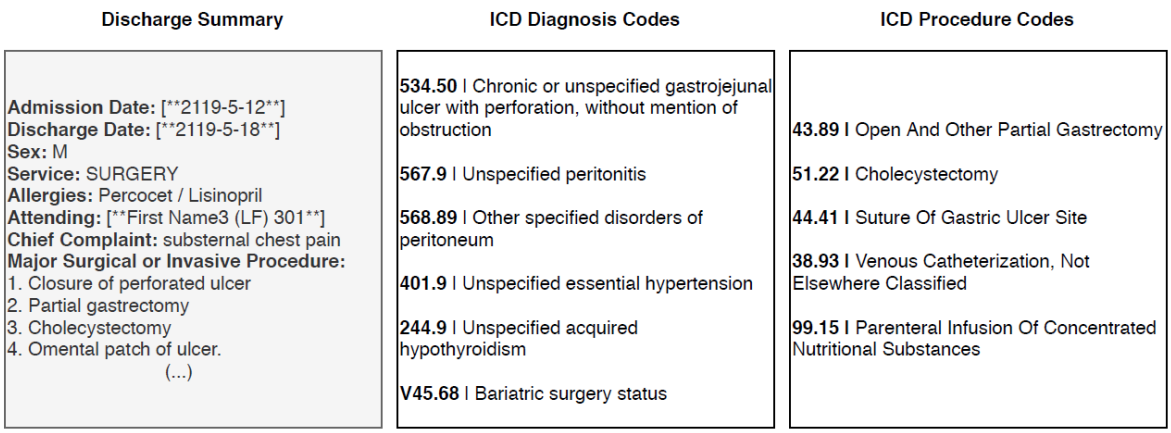


Figure 1.1: An example for ICD coding. Given a hospital discharge summary, a set of ICD codes are assigned. Descriptions of the ICD codes are also provided to make it more comprehensible.

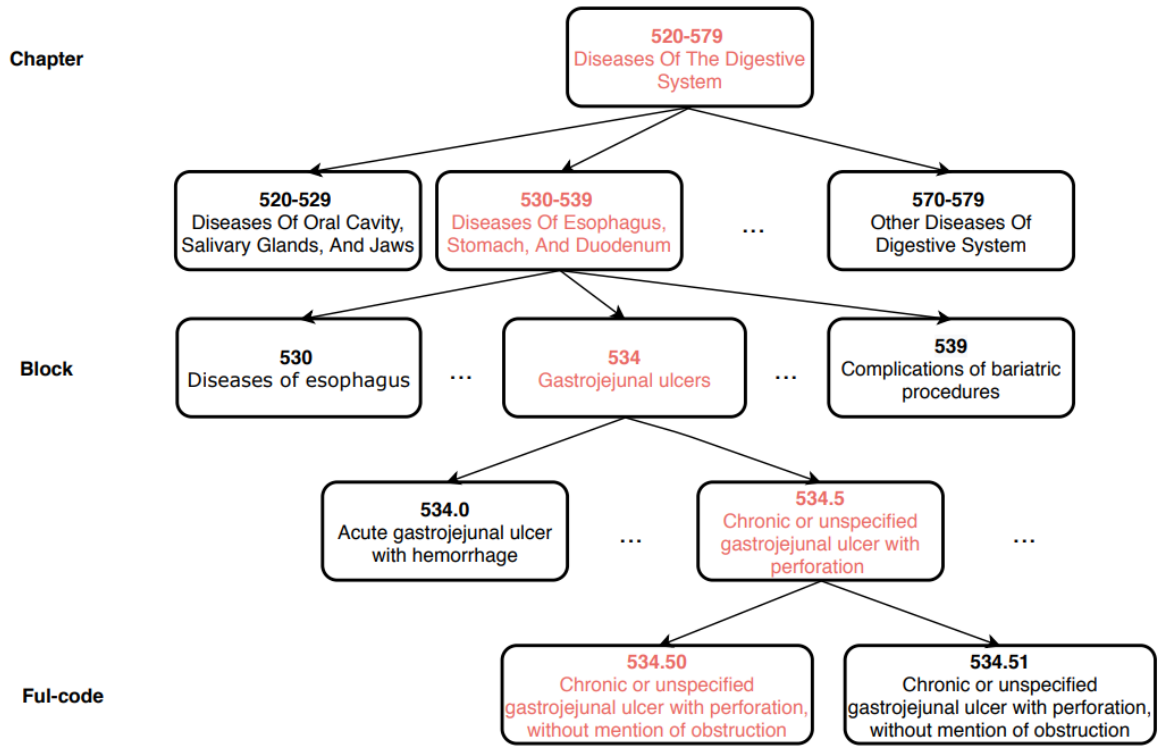


Figure 1.2: Illustration of a medical taxonomy. The tree is based on the hierarchy of ICD codes. The bottom-level codes are usually annotated as the labels of a clinical note by medical coders.

able codes, such as over 17000 in ICD-9[2] and 180000 in ICD-10[3]. Moreover, the distribution of these codes is highly imbalanced, with most codes occurring infrequently, leading to few-shot learning problems, while a small subset of codes appears substantially more often. Finally, the automated models for ICD coding should be computationally efficient and widespread across languages, avoiding the need for pre-training huge models on large amounts of texts from the clinical domain.

Despite the numerous hurdles tied to the automation of ICD coding, the use of deep learning algorithms unquestionably holds the potential to enhance the ICD coding procedure. Sophisticated models

[2] https://www.cdc.gov/nchs/ICD/ICD9cm.htm
[3] https://www.cdc.gov/nchs/ICD/ICD10cmpcs.htm

can be seamlessly incorporated into current coding frameworks, yielding high-priority prediction lists that warrant subsequent manual review. By adopting this approach, the involvement of medical coders in ICD coding can be reduced, decreasing the expenses associated with this endeavor and increasing the coding consistency.

In addition to accurately classifying individual documents, estimating the relative frequency of ICD codes within a dataset is crucial. In fact, in most practical applications around healthcare, we want to estimate the prevalence of an ICD code (or group of codes) in a dataset, rather than simply estimating codes for individual documents. For instance, epidemiologists are often interested in monitoring the prevalence of specific diseases, and this may be done through clinical Natural Language Processing (NLP). Still, more than classifying documents associated to specific individuals (e.g., death certificates, hospital discharge summaries, etc.), this requires the analysis of sets of documents representing a population under a period of analysis. This form of prevalence estimation, also known as text quantification (Schumacher et al., 2021; Moreo et al., 2022), requires properly calibrated text classification models, and provides valuable insights into the distribution of diseases and plays a pivotal role in various healthcare domains, including clinical document coding and disease prevalence estimation.

Prior studies have already introduced various techniques for text quantification that surpass the conventional Classify and Count (CC) approach Levin and Roitman (2017); Moreo and Sebastiani (2021). Nevertheless, situations involving multi-label classifiers or extensive label sets have been relatively overlooked. One conceivable approach to address the challenge of multi-label quantification could involve reformulating the problem as a series of distinct binary quantification tasks Moreo et al. (2022). While straightforward, this approach lacks total adequacy, particularly when the presumption of independence among the target labels is not met — a scenario frequently encountered in ICD coding due to prevalent comorbidities or the inherent correlation between diagnoses and procedures.

1.1 Thesis Proposal

Methods for automatic ICD coding, using a supervised machine learning approach and specifically relying on specialized deep learning neural networks and pre-trained language models, were already developed worldwide. This dissertation describes a novel approach for ICD coding, combining several ideas from previous works. We split long clinical documents into chunks, and use a strong Transformer-based model (Yang et al., 2022a) for processing each of the text chunks independently. The resulting representations are processed with a max-pooling operation, and combined with a label embedding mechanism inspired by that of Yuan et al. (2022), that explores diverse ICD code synonyms. Additionally, taking inspiration on the MLP-based quantification approach from Coutinho and Martins (2023), we explored a training setup in which multi-label classification and text quantification are jointly addressed. This additional step was explored as an approach to potentially improve model calibration.

Our approach aims to achieve well-calibrated state-of-the-art performance in the multi-label classification task, effectively addressing the challenges unique to the ICD coding task. By effectively capturing associations between codes and taking advantage of this information, our goal is to reduce errors in estimating posterior probability and improve the accuracy of predictions and prevalence estimates.

Following prior work, the proposed deep neural network models were evaluated on the publicly available MIMIC-III dataset (Johnson et al., 2016), specifically analyzing two subsets of hospital discharge summaries. The first one is a recently new dataset split, named MIMIC-III-clean (Edin et al., 2023). This subset considers $3,681$ unique ICD-9-CM codes, thus representing a more challenging classification problem. It contains $52,712$ hospital discharge summaries, in which $38,401$, $5,577$, and $8,734$ documents are used as training, validation, and test sets, respectively. The second one yields a subset containing

Table 1.1: Comparison of the MIMIC-III dataset splits.

	MIMIC-III-50	**MIMIC-III-clean**
Number of unique codes	50	3,681
Average number of codes per document	5	14
Number of training documents	8,066	38,401
Number of validation documents	1,573	5,577
Number of test documents	1,729	8,734

the fifty most frequent ICD codes within MIMIC-III (MIMIC-III-50) (Mullenbach et al., 2018). This subset consists of $11,368$ hospital discharge summaries, in which $8,066$, $1,573$, and $1,729$ documents are used as training, validation, and test sets, respectively. Table 1.1 presents a comparison of the two MIMIC-III dataset splits considered in the experiment

1.2 Results and Contributions

The main results and contributions of this dissertation can be summarized as follows:

- A simple version of the proposed classification model without the multi-synonym attention, was evaluated on the publicly available MIMIC-III dataset (Johnson et al., 2016). Two experiments were conducted on this dataset, using the full-clean-label setting (a total of $52,712$ discharge summaries) (Edin et al., 2023) and the top-50 most frequent codes (a total of $11,368$ discharge summaries) (Mullenbach et al., 2018).

- The complete extension of the proposed classification model with a label embedding mechanism that explores diverse ICD code synonyms, was evaluated on the publicly available MIMIC-III dataset (Johnson et al., 2016). The synonyms were extracted from Unified Medical Language System (UMLS), wikidata and wikipedia, and then carefully selected by solving an optimization problem called Maximum Diversity Problem, in order to choose a subset of M synonyms that maximized the sum of the pair-wise embedding distances between the synonyms within each selected synonym group of ICD codes. Two experiments were conducted on this dataset, using the full-clean-label setting (a total of $52,712$ discharge summaries) and the top-50 most frequent codes (a total of $11,368$ discharge summaries). The proposed pre-trained model based, together with the multi-synonym attention, and passing to a two-stage training strategy using the binary cross-entropy objective, proved to be very effective and well-calibrated classifier regarding ICD coding.

- Additionally, taking inspiration on the MLP-based quantification approach from Coutinho and Martins (2023), we explored a training setup in which multi-label classification and text quantification are jointly addressed. This additional step was explored as an approach to potentially improve model calibration. This comprehensive analysis was performed under both previously mention MIMIC III splits (MIMIC-III-clean (Edin et al., 2023), and MIMIC-III-50 (Mullenbach et al., 2018))

- Regarding classification under the MIMIC-III-50 subset, the proposed model achieved macro and micro average F1 scores of 70.3% and 73.6%, respectively, while the previously best reported classifier under this domain (KEPTLongFormer (Yang et al., 2022b)), achieved macro and micro average F1 scores of 68.9% and 72.9%. This shows a improvement in both evaluation metric scores when compared to the latest best recorded model. Under a more challenging subset, the MIMIC-III-clean, the proposed model achieved macro and micro average F1 scores of 31.9% and 73.3%,

respectively, while the previously best reported classifier under this domain (Pre-trained Language Model (PLM)-ICD (Huang et al., 2022)), achieved macro and micro average F1 scores of 26.6% and 72.1%, respectively. It is also noteworthy that this latter model, underwent an adjustment using the validation splits, as the authors reported on model performance after optimizing the decision boundary values through a grid search mechanism to maximize F1 scores in the validation splits. In contrast, our results do not involve any such adjustment, and still surpassed the best reported models to date, establishing a new state-of-the-art approach with a default decision boundary set at 0.5. This shows a significant improvement across all evaluation metric scores in both evaluated dataset splits when compared to the latest best recorded models.

- Regarding quantification under the MIMIC-III-50 subset, the proposed model despite surpassing more simplistic baseline models, we find that the joint optimization does not improve performance over the separate training of an Multi-Layer Perceptron (MLP) for quantification, as previously proposed by Coutinho and Martins (2023). A possible explanation relates to the fact that MIMIC-III-50 does not feature severe class imbalance issues. With a sufficient amount of data for all ICD codes, the multi-synonym attention mechanism is effective in producing well-calibrated classification outputs, leading to good quantification performance. On what regards results over the MIMIC-III-clean split, which features more ICD codes and more severe class imbalance issues, the proposed model outperforms all the baseline approaches by a significant margin, including the use of an MLP that was separately trained for quantification. These results aligned with the observations regarding classifier calibration, highlighting the correlation between the two.

1.3 Thesis Outline

This dissertation is organized as follows:

- Chapter 2 describes fundamental concepts regarding the ICD-9-CM classification system, artificial neural networks, transformers, pre-trained models and the representation of textual data, as well as previous related work focusing on automatic ICD coding of clinical text and the quantification of clinical text.
- Chapter 3 presents the architectural framework that forms the foundation of our research. This section outlines the deep neural network architecture tailored to address ICD coding as a supervised classification task. Additionally, it is elucidated the model architecture and strategy employed for text quantification. Every aspect is detailed to understand the model's structure and the methods for both the text classification and quantification task.
- Chapter 4 reports the outcomes of our rigorous experimental evaluation for both classification and quantification tasks. This includes providing dataset statistics derived from the widely recognized MIMIC-III dataset, along with the evaluation metrics employed. We compare different variants of our comprehensive model against state-of-the-art works in ICD coding, establishing a solid basis for performance analysis. Additionally, we also compared the capabilities of our proposed model to perform ICD code's prevalence estimations against strong baseline methods of text quantification, establishing a solid basis for performance analysis.
- Finally, Chapter 5 offers a concise summary of the main conclusions derived from our study. We highlight the key insights and contributions of our research. Additionally, we present potential avenues for future work in this dynamic field.

Chapter 2

Concepts and Related Work

This chapter aims to make fundamental concepts more accessible while providing a succinct review of related work. We begin by unraveling the intricacies of the ICD-9-CM classification system in Section 2.1. In Section 2.2, we introduce artificial neural networks and their associated gradient-based algorithms for loss function minimization during training. Our exploration continues in Section 2.3, where we delve into the world of advanced deep learning models, particularly transformers. These models have reshaped natural language processing, and we will see how they excel at handling textual data, especially when pre-trained. Section 2.4 introduces the concept of text quantification, including simple yet highly effective baselines. We also take a moment to explore previous works in this area. In Section 2.5, we undertake a comprehensive review of prior research that addresses the automatic coding of clinical text, shedding light on what has been accomplished in this field. To tie it all together, Section **??** summarizes the main topics presented throughout this chapter.

2.1 The International Classification of Diseases System

Responsible for crafting and examining the ICD classification system, the WHO has established the global norm for documenting diseases and health conditions.

This classification structure organizes diagnoses and procedures hierarchically, facilitating convenient storage, retrieval, and analysis of health information to support evidence-based decision-making. Additionally, it enables the sharing and comparing health information among hospitals, regions, settings, and countries. Moreover, it permits data comparisons within a single location across various time frames.

The periodic revision of the ICD system is a standard practice. Currently, hospitals and health facilities are encouraged to adopt ICD-10. The 11th revision, released on June 18, 2018, is already available. However, utilization of ICD-11 for reporting purposes should commence only from January 1, 2022.

This dissertation's experiments utilized the publicly accessible MIMIC-III dataset (Johnson et al., 2016), which adheres to the ICD-9-CM revision. This version is an adaptation introduced by the United States National Center for Health Statistics (NCHS). Although rooted in ICD-9, ICD-9-CM offers more comprehensive morbidity information. It encompasses:

- Volume 1: A tabular list presenting disease code numbers in tabular format.
- Volume 2: An alphabetical index to disease entries.
- Volume 3: A classification system for surgical, diagnostic, and therapeutic procedures, including an alphabetic index and a tabular list (adapted from (Johnson et al., 2016)).

Table 2.1: Titles of the ICD-9-CM (volumes 1 and 2) diagnosis chapters and corresponding range of blocks.

Chapter	Blocks	Title
I	001 – 139	*Infectious And Parasitic Diseases*
II	140 – 239	*Neoplasms*
III	240 – 279	*Endocrine, Nutritional And Metabolic Diseases, And Immunity Disorders*
IV	280 – 289	*Diseases Of The Blood And Blood-Forming Organs*
V	290 – 319	*Mental Disorders*
VI	320 – 389	*Diseases Of The Nervous System And Sense Organs*
VII	390 – 459	*Diseases Of The Circulatory System*
VIII	460 – 519	*Diseases Of The Respiratory System*
IX	520 – 579	*Diseases Of The Digestive System*
X	580 – 629	*Diseases Of The Genitourinary System*
XI	630 – 679	*Complications Of Pregnancy, Childbirth, And The Puerperium*
XII	680 – 709	*Diseases Of The Skin And Subcutaneous Tissue*
XIII	710 – 739	*Diseases Of The Musculoskeletal System And Connective Tissue*
XIV	740 – 759	*Congenital Anomalies*
XV	760 – 779	*Certain Conditions Originating In The Perinatal Period*
XVI	780 – 799	*Symptoms, Signs, And Ill-Defined Conditions*
XVII	800 – 999	*Injury And Poisoning*
	V01 – V89	*Supplementary Classification Of Factors Influencing Health Status And Contact With Health Services*
	E0000 – E999	*Supplementary Classification Of External Causes Of Injury And Poisoning*

The diagnostic[1] codes within ICD-9-CM possess either an alphabetic or numeric initial digit, while the subsequent digits are numeric. These codes consist of a minimum of three and a maximum of five digits. The three-digit variant designates the block, with the fourth and fifth digits detailing the anatomic site or manifestations. This classification comprises 17 chapters, each corresponding to a range encompassing two blocks. Table 2.1 presents an overview of this structure.

For example, let us examine the diagnosis code *403.90* in volumes 1 and 2, described as *"Hypertensive chronic kidney disease, unspecified, with chronic kidney disease stage I through stage IV, or unspecified"* This code pertains to Block 403, titled *"Hypertensive chronic kidney disease"*, positioned within section 401-405, denoted as *"Hypertensive Disease"* within Chapter VII (blocks 390-459), named *"Diseases Of The Circulatory System"*. Within the complete code, the fourth digit characterizes the manifestation *"Unspecified hypertensive renal disease"* the fifth digit specifies the stage of the disease, site *"chronic kidney disease stage I through stage IV, or unspecified"*.

The procedure[2] codes in ICD-9-CM are exclusively numeric, featuring a minimum of two and a maximum of four digits. Here, a block corresponds to the two-digit version. This schema encompasses 18 blocks, as illustrated in Table 2.2.

[1] http://www.icd9data.com/2015/Volume1/default.htm
[2] http://www.icd9data.com/2012/Volume3/default.htm

Table 2.2: Titles of the ICD-9-CM (volume 3) procedure chapters and corresponding range of blocks.

Chapter	Blocks	Title
I	00 – 00	*Procedures And Interventions , Not Elsewhere Classified*
II	01 – 05	*Operations On The Nervous System*
III	06 – 07	*Operations On The Endocrine System*
IV	08 – 16	*Operations On The Eye*
V	17 – 17	*Other Miscellaneous Diagnostic And Therapeutic Procedures*
VI	18 – 20	*Operations On The Ear*
VII	21 – 29	*Operations On The Nose, Mouth, And Pharynx*
VIII	30 – 34	*Operations On The Respiratory System*
IX	35 – 39	*Operations On The Cardiovascular System*
X	40 – 41	*Operations On The Hemic And Lymphatic System*
XI	42 – 54	*Operations On The Digestive System*
XII	55 – 59	*Operations On The Urinary System*
XIII	60 – 64	*Operations On The Male Genital Organs*
XIV	65 – 71	*Operations On The Female Genital Organs*
XV	72 – 75	*Obstetrical Procedures*
XVI	76 – 84	*Operations On The Musculoskeletal System*
XVII	85 – 86	*Operations On The Musculoskeletal System*
XVIII	87 – 99	*Miscellaneous Diagnostic And Therapeutic Procedures*

2.2 Introduction to Neural Networks

The inception of Neural Networks dates back to the mid-20th century (1957), conceived as a computational model inspired by the interconnectedness of biological brain cells. Their nascent form materialized with the creation of the perceptron, a seminal innovation in the late 1950s that simulated a single artificial neuron. The perceptron showcased its practical prowess through the ”Mark 1 Perceptron,” an electromechanical device capable of rudimentary image recognition tasks. The Cornell University psychologist Frank Rosenblatt demonstrated it as the first trainable neural network. However, it was only in the digital age burgeoned that Neural Networks underwent a transformative evolution. Neural networks drove the wave of computational advancements, migrated from their mechanical origins to digital implementations, catalyzed by the emergence of electronic computers.

2.2.1 Artificial neuron

Drawing an analogy from neurobiology, neural networks mirror the intricate network of brain cells, or neurons, which transmit and process information. The perceptron, a linchpin of this architecture, mirrors its biological counterpart by receiving scalar inputs and generating an output akin to a neuron’s firing mechanism. The artificial neuron multiplies each input by its weight and then sums the results, afterwards applying a non-linear function to the result, and using this to produce the output. More formally, we have:

$$y = \varphi\left(\sum_{i=1}^{n} w_i \times x_i + b\right) = \varphi(w^T \cdot x + b). \tag{2.1}$$

In the previous expression, y refers to the output, $x =< x_1, ..., x_n >$ corresponds to the vector of inputs, w denotes the vector of weights, b is a bias term, and $\varphi(.)$ is an activation function that helps the neuron achieving non-linear capabilities.(e.g., a sigmoid, hyperbolic tangent function, ReLU, ect).

2.2.2 Multi-Layer Perceptron

Extending this notion, the MLP emerges, showcasing its prowess in unveiling abstract representations from raw input data. The idea consist of a stacking sets of artificial neural nodes forming the input, hidden and output layers. This more complex formulation gives more complex capabilities to the network, enabling it to perform more complex tasks. A feed-forward network with a single hidden layer can be represented as follows:

$$y = \varphi(C \times \varphi'(B \times \varphi''(A \times x + a) + b)) + c. \tag{2.2}$$

In this case, x corresponds to a vector of inputs and y to a vector of outputs. The matrix A represents the weights of the input layer, and a is the bias vector of the input layer, B and b are, respectively, the weight matrix and the bias vector of the hidden layer, and finally, C and c are, respectively, the weight matrix and the bias vector of the output layer. The functions φ'', φ' and φ denote the non-linear function applied, respectively, to the nodes in the input, hidden and output layers.

With this capability comes a new era of machine learning - deep learning. Distinct from conventional methodologies that rely on manual feature engineering, deep learning embraces automation, as data pre-processing gives way to autonomous machine-driven feature extraction.

2.2.3 Optimization Algorithms

Neural network training uses gradient-based optimization to minimize a loss function across a training dataset. The standard algorithm for training neural networks is the Stochastic Gradient Descent (SGD) method, a general optimization algorithm. Unlike traditional gradient descent, which processes the entire training dataset in each iteration, SGD randomly selects a subset (mini-batch) of data for each iteration. It calculates the gradient of the loss function concerning the parameters using only the selected mini-batch. Then, it updates the parameters in the opposite direction of the gradient to reduce the loss. More formally, it can be mathematically describe as follows:

$$\underset{\mathbf{W}}{\operatorname{argmin}} \sum_{t=1}^{B} \mathcal{L}(\mathbf{W}; (x_t, y_t)). \tag{2.3}$$

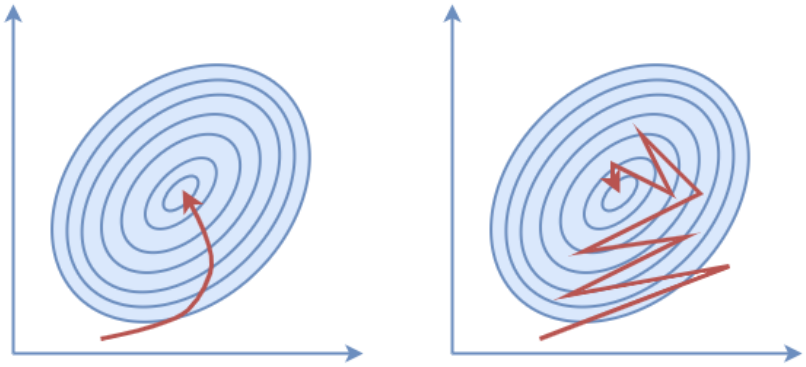
And computationally this optimization problem is iteratively solved using the following expression:

$$W^{k+1} = W^k - \eta \lambda w \left(\sum_{t=1}^{B} \mathcal{L}(W; (x_t, y_t))\right). \tag{2.4}$$

In this case, B is a mini-batch size of the entire dataset length, W are the weights, η is a suitable step size, $\mathcal{L}$ is the loss function to be minimized, and $\lambda_{\mathbf{W}}\mathcal{L}(\mathbf{W})$ is the gradient of the established loss function. W^0 can be initialized in various ways, but a neural network is typically randomly initialized.

This process is not only faster and less computationally intensive (since it only computes the gradients for a subset of the data) but also introduces randomness and noise, which can help escape local

minima. In Figure 2.1 is present a visual representation of the Stochastic Gradient Descent compared with traditional Gradient Descent.



(a) Traditional Gradient Descent (b) Stochastic Gradient Descent

Figure 2.1: Visual representation of (a) Traditional Gradient Descent and (b) Stochastic Gradient Descent.

While the SGD algorithm often achieves acceptable results, the optimization field offers more advanced methods (Ruder, 2016). One prominent example is the Adaptive Moment Estimation (ADAM) algorithm (Adam et al., 2014), a popular technique extensively used to train deep neural networks. With ADAM, updates to the model's parameters are guided by an evolving average of past gradients, coupled with adaptable learning rates for each parameter. This unique feature ensures that infrequently encountered parameters receive substantial updates while frequently encountered ones undergo more subtle adjustments, enhancing the algorithm's effectiveness. Furthermore, a variant of ADAM called ADAMW has emerged, aiming to enhance optimization by disentangling weight decay from adaptive learning rates. This refinement ultimately improves the stability and performance of neural network training.

A crucial step for solving this optimization algorithms, is the gradient computation (Ruder, 2016). This problem can be solved using the backpropagation algorithm (Rumelhart et al., 1986), which is a methodology for computing the derivatives of a complex expression using the chain-rule. The backpropagation algorithm involves two steps: in a forward pass, the predicted outputs corresponding to the given inputs are evaluated; then, in a backward pass, partial derivatives of a given loss function with respect to the different parameters are propagated back through the network.

2.3 Advanced Neural Networks

This section represents the recent technologies in deep learning that are leading to massive breakthroughs in the field, pushing the boundaries of what machines can achieve in understanding and generating human-like information. Moreover, we delve into two pivotal domains that have revolutionized the field: Transformers and pre-trained language models. These neural network models, very recent in their inception, have irrevocably transformed natural language processing and understanding. Within the subsection on Transformers, we explore the architecture that has become the cornerstone of many state-of-the-art models, allowing them to capture intricate contextual relationships. Simultaneously, the subsection on pre-trained language models unveils the power of leveraging vast text corpora to create competent Artificial Intelligence (AI) systems.

2.3.1 Transformers

In artificial intelligence, a revolutionary advancement emerged when the concept of transformers was first proposed. This groundbreaking architecture, initially introduced by Vaswani et al. (2017) in 2017, bore transformative implications for various fields, especially in NLP.

Prior to transformers, sequential models, like Recurrent Neural Network (RNN)-type models, were considered the state of the art in NLP, yet they struggled with capturing long-range dependencies due to their sequential nature. However, the innovative architecture of transformers overcame these limitations by introducing parallelism and attention mechanisms that revolutionized how machines understood and generated human language.

At the heart of this paradigm shift was the introduction of the transformer architecture. Instead of relying solely on sequential processing, transformers harnessed the power of self-attention mechanisms to weigh the significance of different words in a sentence, allowing for a holistic understanding of context. To illustrate, consider a given NLP task (i.e language translation). Traditional models would laboriously process input sentences word by word, often losing track of overarching context. On the other hand, transformers could simultaneously consider all words and their interdependencies, leading to improved linguistic tasks with coherent and contextually accurate results.

The structural brilliance of transformers is built upon several essential components.

The input embedding model translates words into high-dimensional vectors, embedding them into a continuous space where their semantic relationships are preserved. This embeddings represent a token in a d-dimensional space where tokens with similar meaning are closer to one another. However, due to parallelized processing the embedding module, this latter lack the capabilities of encoding the relative position of token inputs, so to address this, the Positional Encoding model comes into play, infusing each token embedding with its position within the sequence.

Central to the Transformers' power is the Encoder model. It consists of layers that iteratively refine the input information.

At the heart of these layers lies the Scaled Dot Product Attention mechanism. Equation 2.5 present a mathematical formulation of this attention mechanism

$$Attention(Q, K, V) = softmax_k\left(\frac{QK^T}{\sqrt{d_k}}\right)V. \tag{2.5}$$

This attention mechanism allows each token to focus on relevant parts of the input sequence, offering a contextual understanding of its surroundings.

This module starts by taking the input embeddings and performs a matrix product between three matrices of trained weights (W_q,W_k,W_v), forming respectively the query matrix (Q), with q_i being a single query vector associated with a single input word, the key matrix (K), with k_i being a single query vector associated with a single input word, and the value matrix (V), with v_i being a single query vector associated with a single input word. Then, after having these three matrices, the module proceeds to calculate the dot product between query and key vectors. This dot product is used to compute the attention score, a similarity score between the query and key vectors.

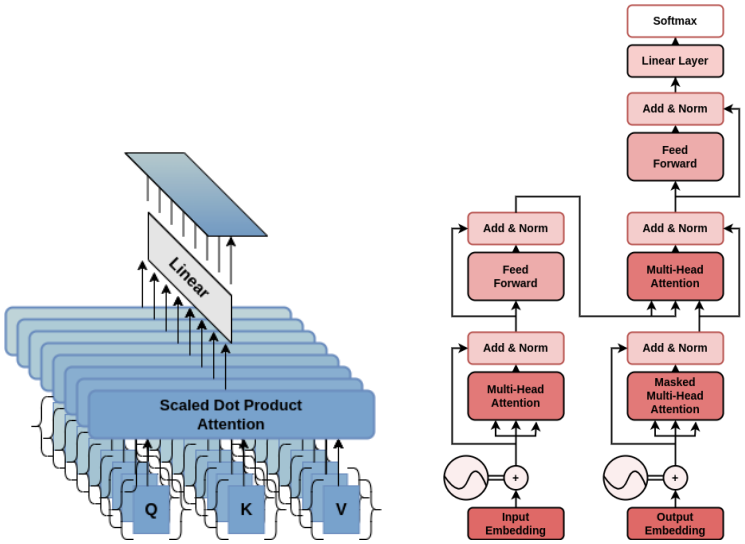
The names of the three matrices are not randomly assigned, and the intention is to behave similarly to what is done in information retrieval. In analogy, in question answering, usually, given a query, we want to retrieve the closest sentence in meaning among all possible answers, so if we pay attention to what is done in equation 2.5, closer query and key vectors will have higher dot products, then by applying the softmax will normalize the dot product between 0 and 1, and finally by multiplying the softmax results to the value vectors will push down close to zero all value vectors for words that had a low dot product score (distant vectors) between query and key vector.

The Multi-Head Attention mechanism, is an ingenious extension of the previously mention attention mechanism. It blends multiple such attention outputs, enabling the model to simultaneously capture various types of relationships and features. This combination of attention mechanisms is the crux of the Transformers, granting it unparalleled sequence comprehension. Figure 2.2 shows a visual representation of Multi-Head Attention mechanism.

The Decoder, sibling to the Encoder, is integral for tasks requiring sequence generation. It utilizes the same fundamental components but introduces a masked attention mechanism during self-attention to prevent future tokens from being seen. This safeguard ensures that the model generates sequences autoregressively, where each token depends only on preceding tokens.

The process mirrors the Encoders for the Decoder Input Embeddings and Positional Encoding. Tokens are embedded and combined with positional encodings, allowing the Decoder to understand the sequence and structure. This sets the stage for the Decoder's autoregressive generation process, where it predicts one token at a time while paying heed to the preceding ones.

Figure 2.2 shows a graphical representation for the complete architecture of the transformer model described above and proposed by Vaswani et al. (2017).



(a) Multi-Head Attention mechanism

(b) Transformer's model architecture

Figure 2.2: Graphical representations of (a) Multi-Head Attention mechanism and (b) Transformer's model architecture

Note that while this description follows the Encoder-Decoder paradigm established in the seminal paper "Attention is All You Need" proposed by Vaswani et al. (2017), subsequent extensions and adaptations of transformers have emerged. For instance, models like Bidirecional Encoder Representations from Transformers (BERT), which only involve encoders, have gained prominence for their capability to generate text embeddings, forming the foundation for various text classification tasks. Moreover, models like chatGPT, built upon decoder-only architectures, have demonstrated remarkable text-generative capacities, showcasing the versatility and impact of transformers across diverse NLP applications.

2.3.2 Pre-trained Models

A pre-trained model in deep learning refers to a model trained on a large dataset before being fine-tuned to a specific downstream task. The idea behind pre-trained models is to leverage the knowledge learned from one task and dataset to improve performance on another related task. This approach has

been particularly successful in transfer learning, where a model pre-trained on a source task can be fine-tuned on a target task with a smaller dataset, leading to improved results and faster convergence.

With the proposal of transformer models, pre-training is preferred using this latter model rather than its predecessor RNN-type models. The primary reason for the absence of pre-trained RNN models is that RNNs need help with vanishing and exploding gradient problems, making it harder to propagate information effectively across long sequences. This results in difficulties in retaining valuable knowledge during pre-training. Additionally, RNNs process sequences sequentially, limiting parallelization and making training slower. These issues make it challenging to develop a widely applicable and efficient pre-training approach for RNNs, leading to the focus on architectures like transformers that can more effectively capture long-range dependencies and parallelize computations.

Utilizing pre-trained models in deep learning offers several compelling advantages. These models, previously trained on extensive and diverse datasets, encode valuable knowledge about various features, patterns, and representations within the data. Leveraging these pre-trained weights as initializations for new tasks accelerates convergence during training, reducing the need for massive labeled datasets and extensive computational resources. This facilitates efficient transfer learning, enabling models to adapt to specific tasks even with limited task-specific data. Moreover, pre-trained models embody learned hierarchical structures, enhancing their ability to generalize and capture intricate patterns in novel datasets. As a result, pre-trained models serve as powerful tools for researchers and practitioners, fostering rapid development, improved performance, and resource-efficient deployment of deep learning solutions across a wide range of applications.

One of the primary uses of pre-trained language models is to represent text data. In natural language, words and sentences are the basic building blocks, but representing words in isolation can fail to capture their relationships effectively.

The breakthrough idea of representing words and subwords as real-value embedding vectors has revolutionized NLP. This approach enables language models to efficiently grasp linguistic patterns and semantic similarities among words. Imagine words as points in a high-dimensional space where similar words are grouped closely together. This concept empowers NLP models to comprehend word relationships, paving the way for advanced language understanding.

Early NLP approaches included the Bag-Of-Words (BOW) model, which treated text as a collection of individual words without considering word order or structure. While BOW had its simplicity, it could not capture the contextual nuances of language.

Models like Skip-Gram and Continuous Bag-Of-Words (CBOW) emerged to address these limitations. They used neural networks to generate word embeddings based on word co-occurrence statistics. Skip-Gram aimed to predict neighboring words, while CBOW predicted the central word from its context. However, they still struggled to capture complex semantic relationships.

Word2Vec and GloVe, introduced by Mikolov et al. (2013) and Pennington et al. (2014), respectively, improved upon earlier models by training on extensive datasets. While they were significant advancements, they provided context-independent word representations, limiting their usefulness for context-aware tasks.

Recognizing the need for context-dependent representations, researchers explored new directions. Horn (2017) proposed Context Encoders, which used trained Word2Vec embeddings to create context-dependent representations based on local contexts. Melamud et al. (2016) introduced Context2Vec, and Peters et al. (2018) introduced ELMo (Embeddings from Language Models), both utilizing bi-Long-Short Term Memory (LSTM) to extract context-sensitive features. These were pioneering innovations in capturing the subtleties of language, as they were the first models to consider word context effectively.

There has been a monumental shift in NLP since introducing the Transformer architecture and pre-trained models in recent years. These groundbreaking innovations have entirely transformed how we

encode textual data into embedding vectors. The Transformer architecture, introduced by Vaswani et al. (2017), introduced self-attention mechanisms, enabling models to capture long-range dependencies in text efficiently. Furthermore, the emergence of pre-trained models like BERT and Generative Pre-trained Transformer (Generative Pre-trained Transformer (GPT)) has taken the field to unprecedented heights. These models, trained on massive text corpora, exhibit an unmatched ability to understand and generate human-like text. They excel in encoding textual data into rich, context-aware embeddings and have become the cornerstone for a wide range of NLP applications, setting the standard for state-of-the-art performance in tasks ranging from machine translation to sentiment analysis.

2.4 Text Quantification

The estimation of the relative frequency (prevalence) of each label within a classification problem has been a highly requested line of work in a variety of domains specially in healthcare. This task is often called quantification (González et al., 2017; Levin and Roitman, 2017). In fact in most practical applications, we want to estimate the distribution of a class (or group of classes) in a dataset, rather than simply estimating the labels of an individual data point. Within the healthcare domain, epidemiologists are often interested in monitoring the prevalence of specific diseases, and this may be done through clinical NLP. Still, more than classifying documents associated to specific individuals (e.g., death certificates, hospital discharge summaries, etc.), this requires the analysis of sets of documents representing a population under a period of analysis.

CC is perhaps the simplest quantification method, which we use as a baseline in our study. Each class i is handled independently, and the method consists of simply counting the number of documents classified as i and dividing by the total number of documents in a sample. Given a classifier c and a sample of documents ϵ, CC for each ICD code i is defined as follows:

$$\hat{p}_{\epsilon}^{CC}(i) = \frac{|\{x \in \epsilon | c_i(x) = 1\}|}{|\epsilon|}. \tag{2.6}$$

A variant that corresponds to a stronger baseline is Probabilistic Classify and Count (Probabilistic Classify and Count (PCC)). Instead of counting the number documents classified as i, we can use the posterior probabilities returned by the classifier, as follows:

$$\hat{p}_{\epsilon}^{PCC}(i) = \frac{1}{|\epsilon|} \sum_{\{x \in \epsilon} p_i(x). \tag{2.7}$$

Despite their conceptual simplicity, both CC and PCC methods are robust baselines, often outperforming more complex quantification methods (Moreo and Sebastiani, 2021).

A variety of different algorithms has been proposed in recent years to deal with text quantification (Schumacher et al., 2021). Previous work has explored the application to different domains, including sentiment quantification over tweets or in the analysis of product reviews (Esuli et al., 2018). Still, few previous studies have specifically considered multi-label settings (Moreo et al., 2022).

One possible solution is simply to recast the problem as a set of independent binary quantification problems. The main problem with approaches such as the last one, and with other previous methods suggested for handling the measurement of text in both binary and multi-class situations, is that they treat labels independently. This becomes problematic when trying to estimate prevalence in a multi-label context where there are strong correlations between labels, leading to high co-occurrence. Moreo et al. (2022) proposed the first truly multi-label quantification methods, leveraging the dependencies among the classes when inferring class prevalence. The proposed methods take inspiration on approaches for

adapting multi-label classification into simpler multi-class problems, e.g. considering sets of ICD codes as labels. The authors methods take inspiration on approaches for adapting multi-label classification into simpler multi-class problems, e.g. considering sets of ICD codes as labels.

More recently, Coutinho and Martins (2023) explored the use of a Multi-Layer Perceptron (MLP) model, inspired on under-complete denoising auto-encoders. The MLP was trained to refine estimates provided by the probabilistic classify and count method, considering label correlations. Experiments with MIMIC-III datasets showed that the proposed method could outperform baseline approaches such as Classify and Count (CC) and Probabilistic Classify and Count (PCC).

In this dissertation, and taking inspiration from the MLP-based quantification approach from Coutinho and Martins (2023), we explored a training setup in which multi-label classification and text quantification are jointly addressed. This step was explored to improve the classes' posterior probability estimation (model calibration), which automatically leads to improvement in text quantification results.

2.5 Related Work on Automatic ICD Coding

NLP has witnessed substantial advancements in recent decades, leading to the enduring challenge of ICD coding in healthcare for over two decades (Larkey and Croft, 1996; de Lima et al., 1998). To automate this task, researchers have explored machine and deep learning approaches. This article examines various strategies in three subsections: firstly, it delves into feature-based machine learning methods (Subsection 2.5.1); secondly, it explores deep learning models for ICD coding (Subsection 2.5.2); and finally, it discusses the latest state-of-the-art models for ICD coding designed to overcome conventional deep learning method limitations (Subsection **??**).

2.5.1 Feature-Based Machine Learning Models

In the early days of assigning ICD codes to clinical text, Perotte et al. (2014) explored two different methods based on Support Vector Machine (SVM)s to tackle this task. The first method, which the authors called the "flat SVM," treated each ICD-9 code independently. It used a separate SVM classifier for each possible ICD-9 code. So, if there were 100 different codes, they would have 100 separate classifiers. In this approach, when training these classifiers, all the documents in the training set labeled with a particular ICD-9 code were considered positive examples, while all others were considered negative examples. When testing, if a document was assigned a specific ICD-9 code, its ancestors in the code hierarchy also had to be positive, and any descendants had to be negative. The second method, the "hierarchy-based SVM," was more sophisticated. It took into account the hierarchical structure of the ICD-9 codes. Imagine ICD-9 codes as a big tree with branches and sub-branches, as it is graphically shown in Figure 1.2. In this approach, they used the hierarchy to create an augmented label set for each document. Then, they trained multiple SVM classifiers, one for each code, but these classifiers were linked according to the hierarchy. This means a classifier associated with a specific code would only be applied if its parent code was also classified as positive. This method's downside is that it makes the dataset more imbalanced since some classifiers could have fewer training documents. The authors hypothesized that since the documents are all relevant to the parent's given code, they will be more informative than all the documents from the flat setting. Both strategies were compared through various metrics, showing that when the hierarchical nature of the codes is leveraged, the modeling is improved.

Koopman et al. (2015) introduced a method to streamline the process of assigning ICD-10 codes to death certificates in cancer-related cases using a two-level SVM-based architecture. The main objective of their research was to tackle two critical questions when dealing with death certificates: firstly, to

determine whether cancer played a role in the cause of death, and secondly, if cancer was indeed a factor, to pinpoint the specific type of cancer involved. To achieve this, they devised a clever system involving two distinct classifiers. The first classifier, the ”binary filter classifier,” is responsible for the initial assessment. It is like a gatekeeper, deciding whether cancer is relevant to the case. If it flags a positive result, indicating that cancer is involved, the second tier comes into play. In the second tier, a series of specialized classifiers, each designed for a specific type of cancer, takes over. They assign the precise ICD-10 code corresponding to the specific type of cancer mentioned in the death certificate.

2.5.2 Deep Learning Models

Following the path of breakthroughs obtained by neural network models in several NLP problems (Goldberg, 2016), deep learning approaches took the lead previously held by machine learning techniques, in handling ICD coding task.

Prakash et al. (2017) introduced condensed memory neural networks (C-MemNNs). This model works by iteratively condensing memory representations while preserving the hierarchy of stored features. To better understand how this works, let us first delve into the concept of a ”memory neural network” proposed by Weston et al. (2015). All relevant information is stored in an external memory in such a network. Each piece of information in this memory has an associated relevance probability, and the network reads these memory slots by taking a weighted sum of their contents. Prakash et al. (2017) stands out by introducing a memory state to the model, which the authors called ”condensed memory state”. This state is created through iterative concatenation, which involves progressively reducing the dimensionality of the input memory state. Cleverly, this process integrates external clinical knowledge with the free-text clinical notes, leveraging the learning capabilities of memory networks to deduce the most likely diagnosis accurately.

Shi et al. (2017) introduced an innovative approach for automating the assignment of ICD (International Classification of Diseases) diagnosis codes based on written medical diagnoses. Their hierarchical deep learning model incorporated an attention mechanism, using character-level LSTM and word-level LSTM networks to represent diagnosis descriptions effectively. This choice was informed by the prevalence of medical terms with shared suffixes denoting similar diseases. The model also featured a two-level LSTM architecture for ICD code representations, with separate parameters for the encoder networks. An attention mechanism was employed to handle the potential mismatch between the number of diagnosis descriptions and ICD codes, offering two variants: ”hard-selection” and ”soft-attention.” The former selected each code’s most influential diagnosis description based on maximum attention scores. At the same time, the latter utilized a softmax function to normalize attention scores across descriptions, recognizing the significance of multiple relevant descriptions. This approach outperformed others using character-unaware encoding and lacking attention mechanisms, with soft attention proving more effective than hard selection. Shi et al. (2017) work significantly advanced in automating ICD code assignment from medical diagnoses.

A study conducted by Duarte et al. (2018) addressed the assignment of ICD-10 codes for causes of death, by analyzing free-text descriptions in death certificates, autopsy reports, and clinical bulletins from the Portuguese Ministry of Health. They leveraged a deep neural network that combines word embeddings, a hierarchical arrangement of recurrent units, neural attention, and mechanisms for initializing the weights of the final nodes of the network. The neural network explores the hierarchical nature of the input data, i.e., words from different fields (word-level) and the fields from different documents (field-level), using bi-Gated Recurrent Unit (GRU)s and an attention mechanism at both levels. The representation produced as the output of the field-level attention mechanism is then concatenated with the average of the embeddings for all words in the input fields, as previously suggested by Joulin et al. (2017). Fur-

thermore, the authors proposed the initialization of the output nodes with the result of a Non-negative Matrix Factorization (NMF), applied to a matrix that encodes label co-occurrences in the training data. Experimental results attested to the contribution of the different neural network components, such as the attention mechanism and the NMF initialization. The model proposed by Duarte et al. (2018) is, in fact, the main source of inspiration for the approach described in this dissertation.

Mullenbach et al. (2018) presented Convolutional Attention for Multi-Label (CAML), i.e., a Convolutional Neural Network (CNN)-based method for automatic ICD code assignment. The authors employed a label-wise attention mechanism in ICD coding, which allows the model to learn distinct document representations for each label. The neural network passes the text through a convolutional layer to compute a base representation of each document's text, making binary classification decisions for each ICD code. Rather than a pooling operation, they apply an attention mechanism to select the most relevant parts of the document for each possible code. As already mentioned, the label space is very high dimensional; thereby, many codes are rarely observed in the labeled data. To improve the performance on rare codes, the authors used text descriptions of each code, building a secondary module in the network that learns to embed them as vectors. A regularization component encourages each code's parameters to be similar to those of codes with similar textual descriptions (DR-CAML). The code embedding consists of a max pooling CNN architecture. Although CAML outperforms previous models on all metrics, DR-CAML performs worse on most metrics than CAML. The experiments were conducted on the MIMIC datasets (Lee et al., 2011; Johnson et al., 2016), and the splits of datasets were publicly available, becoming a milestone for reproducibility in terms of methods for automated ICD coding.

Models such as the one proposed by Mullenbach et al. (2018) employ flat and fixed-length convolutional architectures. There are evident drawbacks in this type of approaches for multi-label clinical classification, which clearly requires variable-size features (such as texts fragments about diseases or procedures) for better representation, since the length and grammar vary significantly in different documents (Xie et al., 2019; Li and Yu, 2020).

Xie et al. (2019) improved the convolutional attention model by leveraging a densely connected CNN together with multi-scale feature attention. Their CNN consists of several stacked convolution blocks via dense connections, producing variable n-gram features layer per layer. Compared to traditional sequentially stacked convolutional blocks, densely connected convolution computes upstream features (larger-scale n-grams) by considering downstream features. Thus, smaller n-grams will be fully used to get larger n-grams, resulting in the flexibility of extracting multi-scale features. After that, the authors apply an attention layer consisting of two components, namely an innovative multi-scale feature attention and, as proposed by Mullenbach et al. (2018), a label-dependent attention. This way, the authors first adaptively select the most informative n-gram features for each word according to the neighborhood. Then, they attend to the most relevant parts of the input for each code. Finally, the authors incorporate a graph CNN to capture both hierarchical relationships among medical codes and the semantics of each code. To get initial node features, they first feed every code description into the description embedding module, consisting of an embedding layer, a convolutional layer, and a max pooling layer. Differently from Mullenbach et al. (2018) in the sense of using code representations V as a regularization, Xie et al. (2019) directly use V to select the most relevant phrase in a label-dependent attention. The graph convolutional neural network captures code relationships and correlations for learning better code classifiers, and every node updates its information by combining the information from its children and parents, in every step. The proposed model, named MSATT-KG, outperforms the CAML method by a considerable margin.

Li and Yu (2020) proposed a novel CNN architecture, combining multi-filter CNN and residual CNN. To capture the patterns with different lengths, the authors leverage the multi-filter CNN, where each filter has a different window kernel size. On top of each filter, there is a residual convolutional layer, which

consists of several residual blocks. Each of these blocks consists of three convolutional filters. Similarly to Mullenbach et al. (2018), the authors also employed a per-label attention mechanism to make each ICD code attend to different parts of the document representation. The advantages of this method regard two different aspects: MultiResCNN not only captures various text patterns with different lengths via the multi-filter CNN, but it also enlarges the receptive field, via the residual CNN. Experiments showed that MultiResCNN performs better than CAML in most evaluation metrics.

2.5.3 Evaluation Split

In their research to enhance automatic ICD coding models, the scientists delved into evaluation using the well-respected MIMIC datasets comprising hospital discharge summaries. Over the past few decades, two primary dataset splits have gained popularity. The first, and introduced by Mullenbach et al. (2018), simplifies matters by focusing solely on the 50 most common ICD codes within MIMIC-III. On the other hand, the second split unleashes the full complexity of MIMIC-III, boasting a daunting $8,929$ distinct ICD codes.

More recently, Edin et al. (2023) introduced a new dataset split, dubbed MIMIC-III-clean. Their work unearthed a crucial revelation: many models struggled due to weak configurations, poorly designed train-test splits, and inadequate evaluation procedures. Not stopping at uncovering these shortcomings, they recalibrated the performance of cutting-edge models on the original MIMIC-III dataset, exposing critical flaws in the evaluation methods and proposing amendments that led to a two-fold increase in macro F1 scores. Their exploration also pinpointed a significant issue with the initial MIMIC-III split, which introduced biases by omitting certain classes in the test set. To rectify this, they engineered a novel split using stratified sampling to ensure a complete representation of all classes, birthing the MIMIC-III-clean dataset. In this new playing field, they conducted a comprehensive model comparison, employing consistent training, evaluation, and experimental setups across the board. Astonishingly, models that previously languished in performance surged to new heights, underlining the pivotal role of hyperparameters and decision boundary fine-tuning. As their journey continued, they reported pioneering results by subjecting state-of-the-art models to the recently released MIMIC-IV dataset, validating their prior findings in this new context. Through meticulous error analysis, they unearthed empirical evidence that shed light on various model vulnerabilities. Perhaps most intriguingly, they challenged a previous assertion, revealing that rare codes posed a considerable performance challenge. At the same time, contrary to prior beliefs, longer documents had only a minimal impact on performance.

2.5.4 State-of-the-Art Models

Vu et al. (2020) introduced A Label Attention Model for ICD Coding (LAAT) and JoinLAAT, models that combine an RNN-based encoder with a new label attention mechanism for ICD coding. The LAAT model employs a Bi-LSTM encoder, which acts as a sort of language understanding engine, capturing the nuanced context within clinical notes. Building on this foundation, the researchers introduced a novel label attention mechanism inpired by Li and Yu (2020)'s structured self-attention concept. This mechanism learns label-specific vectors, essentially identifying crucial clinical text fragments relevant to specific medical labels, and then employs these vectors to construct binary classifiers for each label, making the coding process more precise. Going even further, the JointLAAT model expands upon the label attention model by introducing a hierarchical joint learning mechanism, addressing the challenge of highly imbalanced data. This new approach capitalizes on the hierarchical structure of ICD codes to improve coding accuracy. To put their models to the test, the researchers conducted evaluations using three widely recognized benchmark MIMIC datasets (Lee et al., 2011; Johnson et al., 2016), which

have been extensively used in automatic ICD coding research over the years. This research represents a significant step forward in improving the accuracy and efficiency of ICD coding in clinical settings (Perotte et al., 2014; Prakash et al., 2017; Mullenbach et al., 2018; Xie et al., 2019; Li and Yu, 2020).

Yuan et al. (2022) put forth the Multiple Synonyms Matching Network (MSMN) as an alternative approach to ICD coding. Rather than relying on the code hierarchy, the authors leveraged synonyms to enhance code representation learning and improve coding performance. The MSMN model first applies a shared LSTM to encode Electronic Medical Record (EMR) texts and each synonym. Then, we propose a novel multi-synonyms attention mechanism inspired by the multi-head attention (Vaswani et al., 2017), which considers synonyms as attention queries to extract different code-related text snippets for codewise representations. Finally, we propose using a biaffine-based similarity of codewise text and code representations for classification. To test the models, the authors conducted experiments on the MIMIC-III dataset (Lee et al., 2011; Johnson et al., 2016) with two settings: full codes and top-50 codes (Mullenbach et al., 2018).

In recent years, research has shifted towards Transformer-based language models. Dai et al. (2022) compared Transformer-based models for extended document classification, focusing on mitigating computation overheads associated with encoding extensive text, and in their experiments, the authors observed a clear benefit from being able to process longer text.

To address the challenge of memory consumption due to the full self-attention mechanism, Longformer and BigBird introduced sparse attention, extending the input sequence length and improving long-term dependency modeling. Inspired by their success, Li et al. (2022) presented domain-enriched models, ClinicalLongformer and Clinical-BigBird, pre-trained on large-scale clinical data. In their work, the authors pre-train and publicly release two transformer models for long clinical sequences, namely ClinicalLongformer and Clinical-BigBird, using large-scale clinical notes. The Clinical-Longformer and Clinical-BigBird models significantly improve the performance of a variety of downstream clinical NLP tasks including question answering, named entity recognition and document classification.

Huang et al. (2022) investigated pre-trained language models' limitations, and ultimatelly propose a framework that effectively handles these challenges and achieves superior results on the MIMIC dataset. First, the authors conducted a preliminary experiments to verify and investigate the challenges of using a pre-trained language model framework. The main challenges that were identified were: (1) The length of clinical notes exceeds the maximum length of PLMs; (2) The regular fine-tuning scheme where we add a linear layer on top of the PLMs does not perform well for multi-label classification problems with a large label set. (3) PLMs are usually pretrained on general domain corpora, while clinical notes are very medical-specific and the language usage is different. The author's proposed mechanisms are: 1) segment pooling for the long input sequence problem, 2) label attention for the large label set problem, and 3) domain-specific pretraining for the domain mismatch problem, 2) segment pooling for the long input sequence problem. By integrating these techniques together, we propose PLM-ICD, a framework specifically designed for automatic ICD coding with PLMs. The effectiveness of PLM-ICD is verified through experiments on the benchmark MIMIC-III and MIMIC-II datasets (Saeed et al., 2011; Johnson et al., 2016). Further, Edin et al. (2023) evaluated this model in the MIMIC-III-clean with the proper choice of hyperparameters, revealing the remarkable performance of this model under this dataset split.

Yang et al. (2022b) investigates the high-dimensional space and long-tail adversities concerning the task of automating the ICD coding. To mitigate the data sparsity problem, additional structured knowledge could be applied. ICD codes are organized with an ontological/hierarchical structure where a text description is associated to each code. For the long-tail challenge, this paper present an adapting prompt-based fine-tuning technique with label semantics, which has been shown to be effective under few-shot setting. First the authors pretrain a Longformer Language model on MIMIC-III dataset. Then, pretrained on structured medical knowledge UMLS using self-alignment learning with contrastive loss to

Table 2.3: Summary of the related work on automatic ICD coding using hospital discharge summaries.

Author	Dataset	Results
Vu et al. (2020)	MIMIC-II with 22,815 samples and 5,031 unique codes MIMIC-III with 52,712 samples and 8,929 unique ICD codes. MIMIC-III-50 with 11,368 samples and the 50 more frequent ICD codes.	For LAAT model: MIMIC-II: 48.6% micro-F1 and 5.9% macro-F1 ; MIMIC-III: 57.5% micro-F1 and 9.9% macro-F1; MIMIC-III-50: 71.5% micro-F1 and 66.6% macro-F1. For JointLAAT model: MIMIC-II: 49.1% micro-F1 and 6.8% macro-F1 ; MIMIC-III: 57.5% micro-F1 and 10.7% macro-F1; MIMIC-III-50: 71.6% micro-F1 and 66.1% macro-F1.
Yuan et al. (2022)	MIMIC-III with 52,712 samples and 8,929 unique ICD codes. MIMIC-III-50 with 11,368 samples and the 50 more frequent ICD codes.	For MSMN model: MIMIC-III: 58.4% micro-F1 and 10.3% macro-F1; MIMIC-III-50: 75.2% micro-F1 and 68.3% macro-F1;
Huang et al. (2022)	MIMIC-II with 22,815 samples and 5,031 unique codes MIMIC-III with 52,712 samples and 8,929 unique ICD codes.	For PLM-ICD model: MIMIC-II: 50.4% micro-F1 and 6.1% macro-F1 ; MIMIC-III: 59.8% micro-F1 and 10.4% macro-F1;
Yang et al. (2022b)	MIMIC-III-50 with 11,368 samples and the 50 more frequent ICD codes.	For KEPT-Longformer model: MIMIC-III-50: 72.85% micro-F1 and 68.91% macro-F1 ;

inject medical knowledge into pretrained Language model. For the downstream ICD code assignment fine-tuning, we add a sequence of ICD code descriptions (label semantics) as prompts in addition to each clinical note as KEPT Language Model input. This allows early fusion of code descriptions and the input note. Experiments on common disease coding (MIMIC-III-50).

Chapter 3

Jointly Addressed Classification

In this chapter, we unveil a novel approach to ICD coding that not only strives for top-tier classification performance but also delivers meticulously calibrated outputs, serving as a valuable resource for downstream applications such as text quantification.

Our pursuit of this objective has led us to divide our approach into three distinct sections, each meticulously addressing pivotal aspects of the ICD coding classification process:

1. In the inaugural section (Section 3.1), we explored a chunk-base modeling of clinical texts. Here, we take long clinical documents and break them into manageable segments. Each chunk is processed independently by a robust Transformer-based model. To consolidate the results, we employ a max-pooling operation. This segmentation strategy ensures we can use compact models to process long-length texts effectively.

2. In Section 3.2, we introduce a label embedding mechanism, taking inspiration from the pioneering work of Yuan et al. (2022). This mechanism delves into the world of ICD code synonyms, providing an expansive and insightful perspective on the classification process. By considering the diverse terminologies and synonyms associated with ICD codes, we enhance the breadth and accuracy of our classification system.

3. In the final section, we draw inspiration from the MLP-based quantification approach presented by Coutinho and Martins (2023). Here, we explore a novel training setup that simultaneously addresses multi-label classification and text quantification. This dual approach represents a significant innovation in our methodology, with the potential to enhance model calibration.

Together, these three sections result in a robust and cohesive framework designed to dramatically enhance the precision, efficiency, and calibration of ICD coding and text quantification in the context of hospital discharge summaries.

The implications of this framework are vast, with far-reaching consequences for healthcare data analysis and decision-making. By improving the accuracy of ICD coding and the quantification of clinical text, we pave the way for more robust and insightful healthcare insights, ultimately benefiting medical professionals and patients.

3.1 Chunk-Based Modeling of Clinical Text

Inspired by the intrinsic connection between hospital discharge summary expressions and their ICD code, we devised an approach using segmented classifications and a max pooling layer. The key aspect in this approach is the assumption that if an ICD code is identified in a single segment (i.e., a chunk) of the input document, then that code should be assigned when classifying the document as a whole.

By carefully attending to the ICD codes in each chunk, and employing max-pooling to take into account the contribution of all chunks, we can effectively leverage the capabilities of a standard Transformer encoder, limited to a maximum of T tokens (in our case, $T = 512$), to analyze long clinical documents. To mitigate the loss of information from abruptly breaking interconnected pieces of text, we adopted a smooth partitioning scheme that considers large overlaps between chunks, as shown in Figure 3.1.

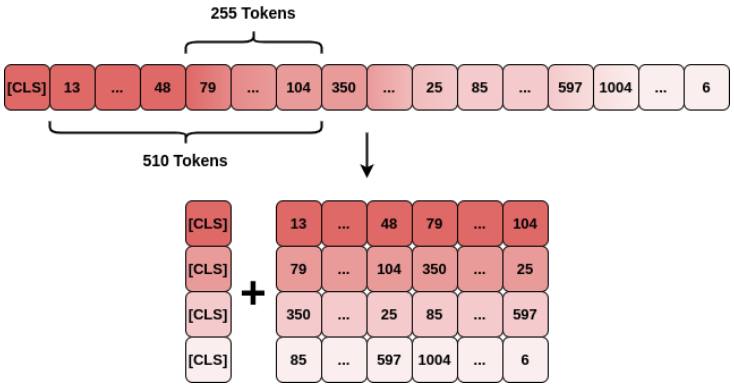


Figure 3.1: Document smooth segmentation with token overlaping

We used a Megatron BERT model (i.e., GatorTron, described by Yang et al. (2022a)) pre-trained on the healthcare domain, consisting of data from clinical notes from the University of Florida Health System, PubMed CC0, WikiText, and MIMIC-III itself Yang et al. (2022a). This model is publicly available in the NVIDIA[1] NGC Catalog and in association with the HuggingFace[2] Transformers library.

Figure 3.2 illustrates the chunk-based classification architecture, where C refers to the number of chunks, T corresponds to the number of tokens within each chunk, H corresponds to the dimensionality of the vectors representing each token, and L denotes the number of ICD classes.

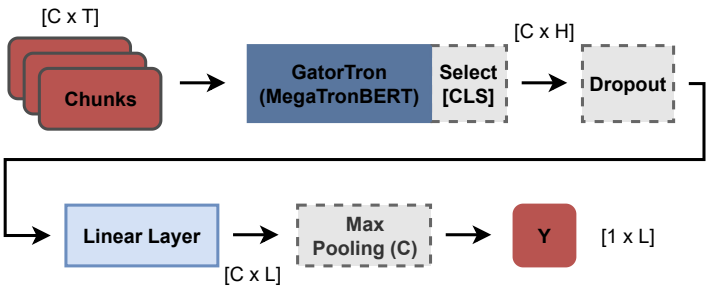


Figure 3.2: Segmented Megatron architecture

[1] https://catalog.ngc.nvidia.com/
[2] https://huggingface.co/UFNLP/gatortron-base

3.2 Multi-Synonym Attention

Inspired by Yuan et al. (2022), we enhanced our classification model through the integration of a multi-synonyms attention mechanism. The primary objective was to explore the intricate relationships between specific mentions to ICD codes, within chunks of the hospital discharge summaries, and the textual descriptions for ICD codes. This integration aimed to leverage synonyms to improve code representation learning (i.e., label embeddings), ultimately aiding in code classification.

We started by extending the ICD-9-CM code descriptions with synonyms obtained from a large medical knowledge base, specifically the UMLS metathesaurus. By aligning ICD codes with UMLS Concept Unique Identifiers (CUIs), we selected corresponding synonyms for English terms sharing the same CUIs. Additionally, we considered synonym variants by removing special characters, allowing only hyphens and brackets, and removing the coordinating conjunctions "or" and "and".

While extending code descriptions, we noticed that the number of synonyms associated with each code were not evenly distributed. This imbalance posed a notable risk of introducing bias in classification, since excessive use of padding values within the representation of each code with fewer synonyms could affect the overall meaning of the synonym cluster. Moreover, this uneven distribution would lead to inefficient memory usage. To tackle this problem, we took steps to rectify it. We enhanced the synonyms linked to the codes by searching for alternative code descriptions from ICD-10. Additionally, to improve diversity, we gathered more synonyms from Wikidata and Wikipedia using query services. Nevertheless, there were codes that still had insufficient synonyms. For those we employed a strategy of duplicating them until we reached a satisfactory threshold of synonyms for each code, denoted as M. These synonyms were first represented as vectors through the same GatorTron model used to represent the text chunks (i.e, taking the `[CLS]` token representation for each synonym). Then, M vectors were selected for each ICD code through the application of the Gurobi optimizer[3] as a way to address the Maximum Diversity Problem[4], which can be formulated as follows:

$$\text{Maximize} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} d_{ij} x_i x_j; \tag{3.1}$$

$$\text{Subject to} \sum_{i=1}^{n} x_i = M; \tag{3.2}$$

$$x_i = \{0,1\}, \quad 1 \leq i \leq n; \tag{3.3}$$

In the previous equations, d_{ij} is a distance metric between synonym representations i and j (i.e., the cosine distance between the vectors), and x_i takes the value 1 if element i is selected and 0 otherwise. Through this optimization problem, we selected a small subset of synonyms (M) that effectively represents the broader embedding space for each ICD code.

Here we denote by Q_l a matrix where rows correspond to the representations for the M synonyms associated to ICD code l, with each code synonym jl composed of tokens $\{s_i^j l\}_{i=1}^{S_j l}$:

$$Q_l = \{\text{GatorEnc}(s_1^{jl}, ..., s_{T_{jl}}^{jl})_{\texttt{[CLS]}}\}_{j=1}^{M}. \tag{3.4}$$

Note that the token representations within each chunk of text c are similarly produced with the GatorTron model, given by:

$$K^c = \text{GatorEnc}(x_1^c, ..., x_T^c). \tag{3.5}$$

[3] `https://www.gurobi.com`

[4] https://grafo.etsii.urjc.es/optsicom/mdp.html

To integrate the text representations from each chunk with the multiple synonym representations, we use an approach inspired by the multi-synonyms attention method proposed by Yuan et al. (2022), which in turn draws inspiration from the multi-head attention mechanism of the Transformer architecture (Vaswani et al., 2017).

We specifically split K^c into Z heads, setting this value to be equal to the maximum number of synonyms per code, i.e. $Z = M$:

$$K^c = K^c_1, ..., K^c_Z. \tag{3.6}$$

The code synonyms $\{Q_l\}_{l=1}^{L}$ are used to query K^c, and by calculating attention scores α_l over K^c, we identify the parts from the chunk's text that are more related to code's synonym l:

$$\alpha_l = \{\text{Softmax}(W_Q Q_l \,.\, \tanh(W_K K^c))\}_{c=1}^{C}; \tag{3.7}$$

We then use avg-pooling of $\tanh(K)\alpha_l$ assuming our intention is to create code-wise text representations R by averaging the contributions from synonyms:

$$R = \{\text{AvgPool}(\tanh(K)\alpha_l)\}_{l=1}^{L}. \tag{3.8}$$

To assess whether the text of a chunk c contained code l, we evaluate the similarity between the code-wise text representation R_c and code's embeddings V. We aggregate the code synonym representations Q to form a code representation V through avg-pooling, resulting in a matrix with each row depicting a global representation of each code. To measure the similarity for classification, we apply a bi-affine transformation. Finally, after carefully attending to the ICD codes in each chunk using synonyms to enhance the classification, we employ max pooling to consolidate the results:

$$V = \text{AvgPool}(Q^1, Q^2, ..., Q^M), \tag{3.9}$$

$$Y = \sigma(\text{MaxPool}(\text{Diag}(R_1^T WV), ..., \text{Diag}(R_C^T WV))). \tag{3.10}$$

Unlike previous approaches that perform classification using code-dependent parameters, which can be challenging to define for rare codes, our bi-affine function uses code-independent parameters WV. This approach simplifies the learning process, at the same time making it more effective.

Figure 3.3 illustrates the process behind the chunk-based classification method that considers the multi-synonyms attention mechanism.

In exploring loss functions to tackle the challenges posed by the binary multilabel characteristics of the classification task, we adopted the widely-used Binary Cross-Entropy (BCE) Loss. Binary cross-entropy loss treats each class prediction independently, assessing the model's ability to discern the presence or absence of each label for a given input. By optimizing the model's parameters using this loss function, it learns to assign probability scores to each label, effectively capturing the complex interactions between them. The resulting probabilities indicate the likelihood of an instance belonging to each class, making the binary cross-entropy loss a versatile tool for multilabel binary classification. This loss is formally described as follows:

$$\mathcal{L}_{BCE} = \sum_{l \in L} -y_l \log(\hat{y}_l) - (1 - y_l)\log(1 - \hat{y}_l).$$

The variable $y_l \in \{0, 1\}$ is the ground truth for an instance l, $\hat{y}_l$ is the probability of the label y_l being true given by the classifier.

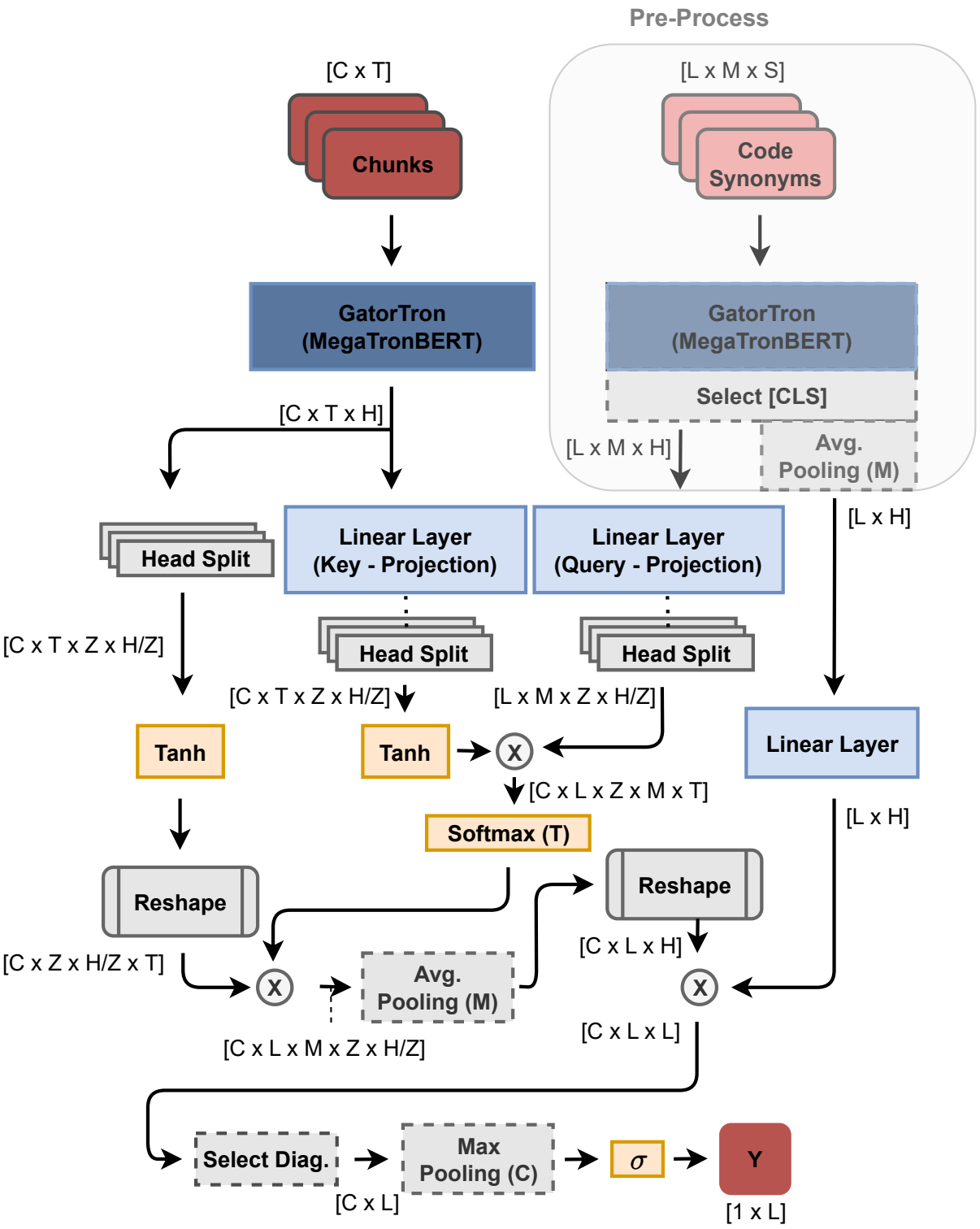


Figure 3.3: Segmented Megatron with Multi-Sysonyms attention mechanism

3.3 Jointly Classify & Quantify Model

Following previous work by Coutinho and Martins (2023), we considered the use of an under-complete denoising auto-encoder to quantify the prevalence of ICD codes within a set of documents, accounting with label associations. We integrated this quantification module, implemented as a three-layer MLP, together with the classifier, performing end-to-end training of the resulting model. We hypothesise that the classification and the quantification objectives can naturally complement each other, contributing to improved model calibration.

Notice that classification operates at the level of individual instances, while quantification operates over groups of instances. To integrate both objectives within end-to-end training, we follow the steps:

1. **Shuffling and setting a limit:** We shuffle the training dataset at the start of each training epoch. We also establish a limit that simulates the maximum number of instances that will be considered for quantification.

2. **Iterative data collection:** We process the instances individually as we progress through the training set. For each instance that is processed, we collect the classification results until we hit the previously defined maximum limit. This creates a new group of instances for each new instance that is processed, consisting of the ones we have processed thus far, plus the latest instance. The processing of each instance is made as follows:

   (a) **Computation of classification loss:** When processing each new instance, we apply our classification model and calculate the classification loss associated to that instance.

   (b) **Computation of quantification loss:** We take the classification output and add it to the previous classification outputs. This combination allows us to compute a probabilistic classify and count vector, denoting the estimated relative frequency of each class label within the group of instances. We then process this vector using the aforementioned MLP, which refines the probabilistic classify and count estimates. We finally calculate the quantification loss with the refined estimates.

   (c) **Aggregation of results:** The losses computed in the previous steps are aggregated and used to update the model parameters for each batch of instances that is processed.

3. **Repeat and reset:** We follow the iterative process (steps (a) to (c)) until we reach the maximum number of instances designated for the quantification set. Once this limit is reached, we reset the quantification group and establish a new maximum limit for the instances to be quantified, continuing with model training until a stopping criteria is meet.

Our combined loss function can be formally described by the following equation, where λ is an hyper-parameter controlling the relative influence of the quantification loss:

$$\mathcal{L} = \mathcal{L}_C + \lambda \mathcal{L}_Q. \tag{3.11}$$

The classification loss ($\mathcal{L}_C$) is the BCE formally described in Equation 3.2, while the quantification loss ($\mathcal{L}_Q$) uses the MSE, given by:

$$\mathcal{L}_Q(\hat{p}_\epsilon^{\text{MLP}}, p_\epsilon) = \sum_{l=1}^{L} |\hat{p}_\epsilon^{\text{MLP}}(l) - p_\epsilon(l)|^2, \tag{3.12}$$

where p_ϵ is the ground-truth quantification result (i.e., the relative class frequency within the set of instances) for each of the L class labels.

The MSE loss was preferred over other regression-type losses, such as the MAE, because it provides a smoother optimization landscape, leading to more stable and accurate results.

3.4 Summary

In this chapter, we introduce an innovative approach to ICD coding. Our method not only aims for top-notch classification performance but also provides carefully calibrated results, which are valuable for various downstream applications like text quantification. We have divided our approach into three key sections, each meticulously addressing crucial aspects of the ICD coding classification process.

In the first section (Section 3.1), we detail our initial approach. Here, we use a base model to classify high-length documents in smaller chunks. The idea is that if a class is correctly identified in a document segment (chunk), it is also correctly classified when considering the entire document. We pay close attention to the ICD codes in each chunk and use max pooling to combine the final results. This strategy allows us to effectively use a more compact model, which is limited to a maximum number of tokens (in our case, 512), to analyze documents that exceed the model's token input limit.

Moving on to Section 3.2, inspired by previous work of Yuan et al. (2022), we describe an approach that involves encoding labels using multiple synonyms for ICD codes. This enhancement enriches the model with code synonyms, expanding its knowledge of ICD code expressions. A plausible analogy for this strategy is that the model uses a "cheat sheet" of code synonyms to aid in classifying clinical documents with ICD codes. This module leverages the model's learning capabilities while using synonyms that may appear in the document alongside their respective ICD codes without augmenting the training data with sentences containing synonyms for the model to learn these associations. We gathered ICD code synonyms from various clinical sources (such as UMLS and ICD-10), Wikipedia, and Wikidata. We carefully selected these synonyms by solving an optimization problem known as the Maximum Diversity Problem. Our primary goal in this selection was to enhance diversity. This was achieved by considering the vector representations of synonyms, aiming to maximize the diversity of the code cluster while normalizing the number of synonyms for each code to a reasonable count (say, M synonyms per code).

Finally, in Section 3.3, we introduce a novel training setup inspired by a quantification approach presented by Coutinho and Martins (2023). This approach simultaneously tackles multi-label classification and text quantification by using a joint loss function that combines information from both tasks to train the two processes simultaneously. This dual approach represents a significant innovation in our methodology and can potentially enhance model calibration. The quantification module takes a weighted sum of the posterior probabilities returned by the classifier, the PCC vector, as input. It processes it to minimize the error distribution of the classes, taking into account potential label correlations.

Together, these three sections form a robust and cohesive framework designed to significantly improve the precision, efficiency, and calibration of ICD coding and text quantification in the context of hospital discharge summaries. The implications of this framework are vast and have far-reaching consequences for healthcare data analysis and decision-making. By enhancing the accuracy of ICD coding and clinical text quantification, we pave the way for more reliable and insightful healthcare insights, ultimately benefiting both medical professionals and patients.

Chapter 4

Experimental Evaluation

In this chapter, we embark on an in-depth exploration of the experimental evaluation of our novel method. Our objective is to provide a comprehensive understanding of our approach by comparing it to previously reported results in the field.

To begin, Section 4.1 will offer a meticulous statistical breakdown of the dataset employed in our experiments. This dataset will serve as the foundation for our testing conditions, encompassing classification and quantification problems. In addition to presenting the dataset's statistics, we will also shed light on the primary challenges associated with it, enhancing our understanding of the context in which our method operates.

Moving forward, Section 4.2 will elucidate the intricacies of the evaluation criteria we've adopted. These criteria are essential for rigorously analyzing and comparing the results obtained in classification and quantification tasks. This thorough evaluation ensures the robustness of our models and validates their performance against industry standards.

Our journey into the inner workings of our method continues in Section 4.3. Here, we delve deeper into the hyperparameters utilized in our classification and quantification models. By doing so, we provide valuable insights into the fine-tuning process that underlies the success of our experimental endeavors. This section is instrumental in understanding the technical aspects contributing to our method's efficacy.

In the subsequent Section 4.4, we engage in insightful comparisons between our proposed approach and the cutting-edge models and strategies for both ICD coding and ICD quantification. Through this comparative analysis, we aim to highlight the strengths and advantages of our method, showcasing its contributions to the field.

Finally, in Section 4.5, we draw the curtains on this chapter by summarizing the results we've obtained. This summary encapsulates the essential findings and takeaways from our experimental evaluation, providing a comprehensive overview of the achievements and contributions of our research.

4.1 Datasets

Experiments were conducted using publicly available MIMIC-III data (Johnson et al., 2016). This dataset contains comprehensive insights about patients under critical care, and the access to it was granted through PhysioNet[1], after completing the ethical training by the Collaborative Institutional Training Initiative program. Following the line of previous works, our experiments took place using well established and publicly available MIMIC-III splits:

- **MIMIC-III-50:** This split comprises the top-50 most frequent codes present in MIMIC-III dataset. This split was first introduced by Mullenbach et al. (2018) and has been vastly adopted to train, validate and test further experiments that came up regarding ICD coding hospital discharge summaries, including the one reported in this dissertation.
- **MIMIC-III-clean:** This split corresponds to a cleaned dataset version of the entire MIMIC-III complaining $3,681$ unique ICD-9-CM codes, thus representing a more challenging classification problem (MIMIC-III-clean Edin et al. (2023)).

One of the main challenges in the ICD coding classification task is the highly imbalanced label distribution. Most codes appear very seldom, while only some occur several orders of magnitude more than others. $5,418$ from $8,921$ possible codes occur less than ten times in the entire MIMIC-III training dataset, which complains of few-shot training tasks for most ICD codes. This disparity was one of the main reasons that led to Edin et al. (2023) to propose a new split to test a more complex scenario than the MIMIC-III-50 but within a more plausible training, validation, and test environments. In this new split, the authors call it MIMIC-III-clean, only 401 from $3,681$ possible codes occur less than ten times in the training set. Table 4.1 summarizes the statistics of each set for the two experiments.

Table 4.1: Statistics for training, validation and test sets of MIMIC-III-clean and MIMIC-III-50 datasets. The column "Words pr. Doc" corresponds to the average and maximum number of words per hospital discharge summary. The column "Tokens pr. Doc" corresponds to the average and maximum number of tokens per hospital discharge summary. The column "Unique codes" corresponds to the unique number of ICD codes in the respective split. The column named "Type of codes" has two sub-columns "Diag." and "Proc." correspond to the number of unique diagnosis and procedure codes only, respectively.

	Split	Samples	Words pr. Doc.		Tokens pr. Doc.		Codes pr. Doc.		Unique	Type of Codes	
			Avg.	Max.	Avg.	Max.	Avg.	Max.	Codes	Diag.	Proc.
MIMIC-III-50	Train	8,066	1,642	7,989	2,830	20,297	5.4	18	50	33	17
	Val	1,573	1,932	6,658	3,410	16,566	5.9	21	50	33	17
	Test	1,729	1,964	6,470	3,465	11,871	6.0	20	50	33	17
MIMIC-III-clean	Train	38,401	1,514	10,500	1,651	11,758	14.0	57	3,681	2,849	832
	Val	5,577	1,552	6,393	1,694	6,897	15.9	60	3,676	2,844	832
	Test	8,734	1,485	7,858	1,619	8,299	14.8	56	3,681	2,849	832

An important aspect of both dataset splits is the information about the frequency of ICD codes present in the both training and test sets for both MIMIC-III splits. This information is divided in three relevant percentiles. Here is a breakdown of what these percentiles represent: Low Percentile (Low Pth): This corresponds to the interval of codes for the 10% of medical codes with the lowest frequency rates in

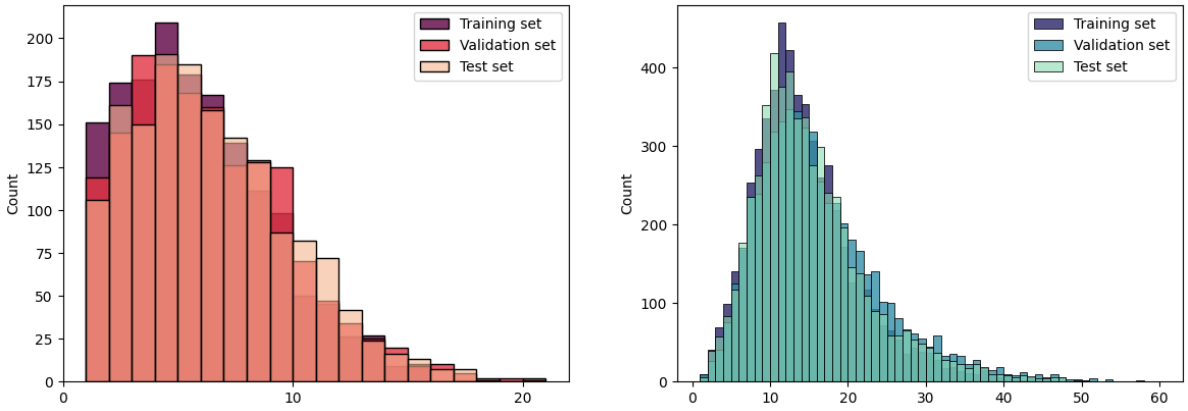
[1] https://physionet.org/content/mimiciii/

the training set of the respective MIMIC-III split. Medium Percentile (Medium Pth): This represents the interval of codes for the 10% of medical codes with medium frequency rates, falling within the 55% to 65% range in the respective MIMIC-III split training set. High Percentile (High Pth): This indicates the interval of codes for the 10% of medical codes with the highest frequency rates in the training set of the respective MIMIC-III split. In Table 4.2 is present this absolute frequency information of code occurrences in the above mentioned percentiles.

Table 4.2: Interval of code occurrences in a specific percentile of code frequency.

Dataset	Split	Low Pth: [0-10%]	Medium Pth: [55%-65%]	High Pth: [90%-100%]
MIMIC-III-50	**Train**	397-449	759-914	1615-3233
	Test	60-127	148-247	402-470
MIMIC-III-clean	**Train**	4-9	36-56	308-14,598
	Test	1-4	6-27	55-2228

Besides the highly imbalanced and sparse feature space, another challenge concerning the multi-label scenario is the high number of ICD codes assigned to each discharge summary. Figure 4.1 presents the frequency of discharge summaries associated with the number of ICD-9-CM codes per discharge summary for training, validations and test set for both MIMIC-III-50 and MIMIC-III-clean. Although the average number of unique labels per instance in training set is 14 and 5.4 for MIMIC-III-clean and MIMIC-III-50, it is possible to find up to 57 and 18 codes, respectfully for both splits, associated with a single discharge summary.

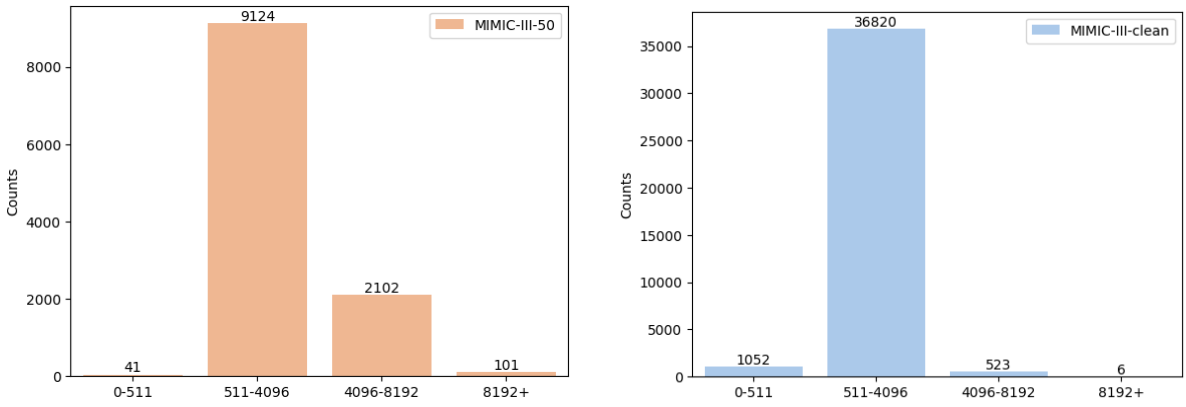


(a) MIMIC-III-50

(b) MIMIC-III-clean

Figure 4.1: Number of ICD-9-CM codes per instance of the dataset.

Furthermore, another critical challenge associated with ICD coding centers on the considerable length of hospital discharge summaries. While the average token length per sentence in the text is $2,829$ for MIMIC-III-50 and $1,651$ for MIMIC-III-clean, it is worth noting, as highlighted in the statistical breakdown of the MIMIC-III splits presented in Table 4.1, that both splits (MIMIC-III-clean and MIMIC-III-50) contain a substantial number of sentences that exceed these averages by a significant margin. This presents a significant challenge to effectively incorporating all tokens within such lengthy sentences into our deep learning models. For a visual representation of the distribution of hospital discharge summaries in terms of input token length, please refer to Figure 4.2, which categorizes the data into four relevant bins.



(a) MIMIC-III-50 (b) MIMIC-III-clean

Figure 4.2: Barplot of four principal bins of the input token length of discharge summaries per dataset.

Figure 4.3 shows the distribution for the 50 most common ICD-9-CM codes in the dataset.

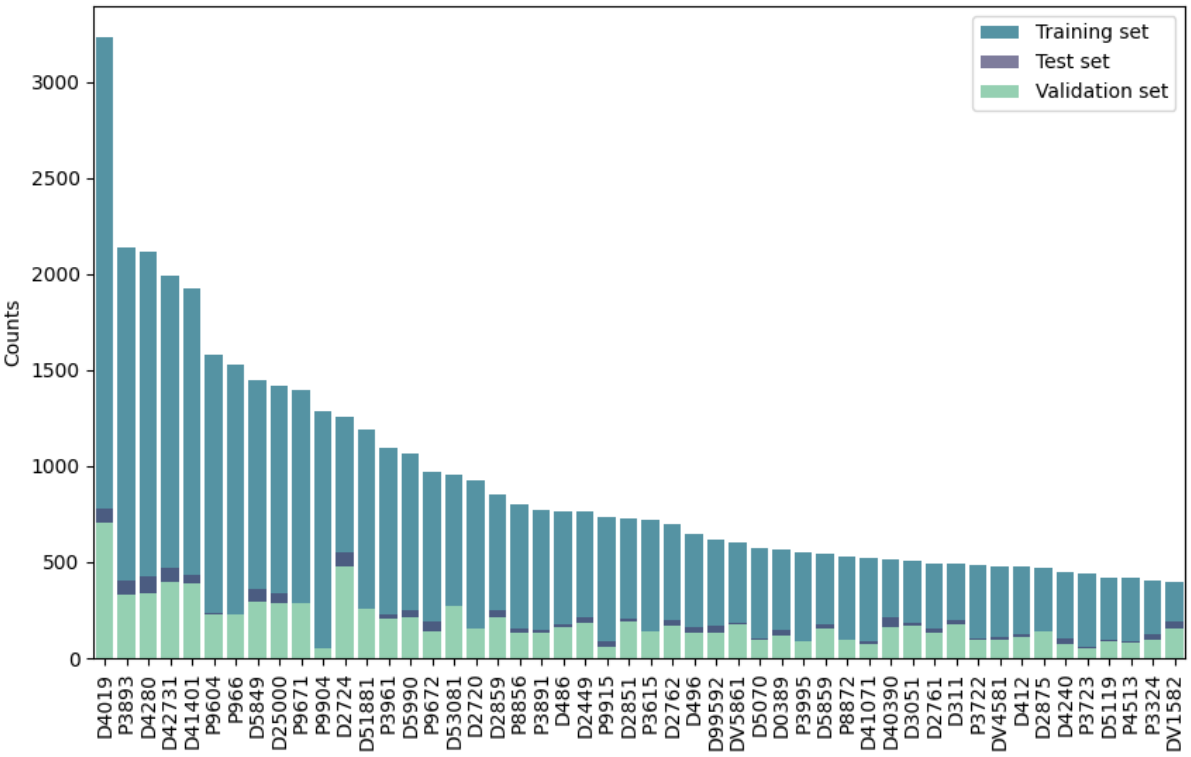


Figure 4.3: Number of occurrences of the 50 most common ICD-9-CM codes in the dataset.

The quantification experiments also used MIMIC-III-50 and MIMIC-III-clean, following the general methodology and guidelines from Coutinho and Martins (2023). Specifically, for assessing result quality, we sampled documents from the validation set order to form $5,000$ quantification groups of different sizes, with the size parameter varying between one and the number of documents in the set. A separate set of $1,000$ groups was also created by sampling documents from the test split. These were used for (pre-)training the MLP quantification model and test the different quantification experiments.

4.2 Performance Metrics

This section serves as the yardstick by which we assess the effectiveness and accuracy of our proposed models. This subsection has two subsubsections, each tailored to a specific task. In the Classification subsubsection (Subsubsection: **??**), we meticulously define and discuss the evaluation metrics chosen to gauge our models' classification performance and calibration, ensuring a comprehensive analysis of their ability to categorize data accurately. Meanwhile, in the Quantification subsubsection (Subsubsection: **??**), we introduce a distinct set of metrics tailored to assess the proper tools to measure our proposed models' precision to estimate the class prevalences, enabling us to compare them to baseline methods. Together, these subsections provide a holistic view of our models' performance across varied tasks, facilitating a well-rounded evaluation of their capabilities.

4.2.1 Classification

To establish a fair comparison of performance with prior work, the results of the proposed model are reported on a variety of metrics, focusing on micro-averaged and macro-averaged F1, Area Under the Curve (Area Under the Curve (AUC)) and precision at n (Mullenbach et al., 2018). Micro-averaged values treat each pair of text codes as a separate prediction. On the other hand, macro-averaged values are computed by averaging metrics computed per label. Opitz and Burst (2019) looked closely at standard macro F1 formulas used in multi-class and multi-label classification, and they found a flaw in the traditional approach. This uncovered blemish using the harmonic mean overly favored biased classifiers when dealing with imbalanced datasets. Therefore, as recommended by the authors and following the guidance of Edin et al. (2023), we decided to compute the F1 macro score using the arithmetic mean for each class as it is shown in equation 4.7.

For the case of precision, the metrics can be distinguished as follows:

$$Micro - Precision = \frac{\sum_{\ell=1}^{|\mathcal{L}|} TP_\ell}{\sum_{\ell=1}^{|\mathcal{L}|} TP_\ell + FP_\ell}; \tag{4.1}$$

$$Macro - Precision = \frac{1}{|\mathcal{L}|} \sum_{\ell=1}^{|\mathcal{L}|} \frac{TP_\ell}{TP_\ell + FP_\ell}. \tag{4.2}$$

In the previous expression, TP denotes true positives (i.e., examples that the model correctly predicts a ICD code that was assigned to a discharge summary) and FP false positives examples (i.e., examples that the model incorrectly predicts a ICD code that was not assigned to a discharge summary). A variation of the traditional precision also explored in our evaluation metric is the precision at n. $P@n$, gauges the precision of the top n labels with the highest scores, verifying their presence in the actual data. Four our evaluation on the MIMIC-III-50 dataset, we selected $n = 5$, roughly aligning with the average code count. Additionally, for experiments conducted on MIMIC-III-clean, we employed $n = 8$ and $n = 15$, reflecting varying degrees of label precision complexity.

Similarly, recall metrics are represented as follows:

$$Micro - Recall = \frac{\sum_{\ell=1}^{|\mathcal{L}|} TP_\ell}{\sum_{\ell=1}^{|\mathcal{L}|} TP_\ell + FN_\ell}; \tag{4.3}$$

$$Macro - Recall = \frac{1}{|\mathcal{L}|} \sum_{\ell=1}^{|\mathcal{L}|} \frac{TP_\ell}{TP_\ell + FN_\ell}. \tag{4.4}$$

In the previous expression, FN denotes false negatives examples (i.e., examples that the model does

not predict a ICD code that was assigned to a discharge summary).

The F1 metrics correspond to the harmonic mean of the precision and recall, as follows:

$$F1 = 2 \times \frac{precision \times recall}{precision + recall}; \tag{4.5}$$

$$F1 - micro = \frac{\sum_{\ell=1}^{|\mathcal{L}|} TP_\ell}{\sum_{\ell=1}^{|\mathcal{L}|} TP_\ell + \frac{1}{2} \cdot \left(\sum_{\ell=1}^{|\mathcal{L}|} FP_\ell + \sum_{\ell=1}^{|\mathcal{L}|} FN_\ell \right)}; \tag{4.6}$$

$$F1 - macro = \frac{\sum_{\ell=1}^{|\mathcal{L}|} (F1)}{Numberofclasses}. \tag{4.7}$$

A well-calibrated classifier in deep learning is paramount for classification and quantification accuracy. It ensures that the predicted probabilities assigned to each class accurately represent the actual likelihood of those classes, enabling an accurate understanding of the model's confidence in its predictions and informs decision-making. In classification tasks, well-calibrated probabilities allow for setting appropriate decision thresholds, improving interpretability, and enhancing trustworthiness. Moreover, well-calibrated probabilities are essential for quantification, helping to estimate class relative frequencies more accurately, which is vital in applications within the healthcare domain. To measure the calibration performance of our models, we used the MECE which is formally written as follows:

$$MECE = \frac{1}{L} \sum_{i=1}^{L} \sum_{j=i}^{N} b_j ||(p_{ij} - c_{ij})||. \tag{4.8}$$

Where p_{ij} is the top-1 prediction accuracy in bin j for class i, c_{ij} is the average confidence of predictions in bin j for class i, and b_j is the fraction of data points in bin j.

4.2.2 Quantification

Following previous work (Moreo et al., 2022; Sebastiani, 2020), we use the Absolute Error (AE) and the Relative Absolute Error (RAE) as evaluation metrics for quantification. We additionally divide the error by the number of samples, obtaining the Mean Absolute Error (MAE) and the Mean Relative Absolute Error (MRAE):

$$MAE(p, \hat{p}) = \frac{1}{N} \sum_{j=1}^{N} \left(\frac{1}{L} \sum_{i=1}^{L} |p_{e_j}(i) - \hat{p}_{e_j}(i)| \right); \tag{4.9}$$

$$MRAE(p, \hat{p}) = \frac{1}{N} \sum_{j=1}^{N} \left[\frac{1}{2L} \sum_{i=1}^{L} \left(\frac{|p_{e_j}(i) - \hat{p}_{e_j}(i)|}{p_{e_j}(i)} + \frac{|(1 - p_{e_j}(i)) - (1 - \hat{p}_{e_j}(i))|}{(1 - p_{e_j}(i))} \right) \right]; \tag{4.10}$$

where p is the ground-truth, $\hat{p}$ is a quantification method, p, $\hat{p} \in \mathbb{R}^{NxL}$, and N is the number of samples in the evaluation dataset. Since the MRAE is undefined when $p_{e_j}(i) = 0$ or $p_{e_j}(i) = 1$, we smooth the probability distributions p_{e_j} and $\hat{p}_{e_j}$ via additive smoothing, as follows:

$$s(p_{e_j}) = \frac{\gamma + p_{e_j}}{2\gamma + 1}; \tag{4.11}$$

with $\gamma = (2|e_j|)^{-1}$ as the smoothing factor.

4.3 Implementation Details

This section delves deeper into the pertinent hyperparameters and training strategies which enabled us to deal with both classification and quantification tasks.

Table 4.3 presents the training hyper-parameters considered in our experiments.

Table 4.3: Hyper-parameters used for model training in the MIMIC-III-50 and MIMIC-III-clean settings. The *max number of epcochs* values are related to the classification and quantification modules.

Parameters	**MIMIC-III-50**	**MIMIC-III-clean**
Maximum token input length	$7,142$	$6,122$
Token overlapping window	255	255
GatorTron hidden size	$1,024$	$1,024$
Synonyms per ICD code (M)	4	4
Number of heads (Z)	4	4
Maximum number of epochs	300	300
Early stopping patience	5	5
Effective batch size	16	16
Adam e	$1e$-8	$1e$-8
Starting learning rate	$2e$-5/$2e$-7	$2e$-5/$2e$-7
Ending learning rate	0	0
MLP hidden size	32	$3,072$
Quantification coefficient (λ)	100	100
Learning rate scheduler	linear	linear

Since the proposed model processes the input text in chunks, the maximum allowable token length is limited only by hardware constraints. During training, we had to cap the maximum input token length due to restrictions in the available GPU memory. However, we could further raise this limit in the test environment, up to $20,000$ tokens.

We trained our classifiers in two stages. The first stage uses a learning rate starting at $2e$-5 and proceeds until we reach the early stopping criteria. We then perform a second training stage, with a learning rate starting at $2e$-7. The quantifier model (MLP) was first trained individually following the guidelines of Coutinho and Martins (2023), using a constant learning rate schedule starting at $2e$-5 and proceeds until we reach the early stopping criteria.

The model that integrates the quantification objective was initialized with pre-trained classification and quantification components, obtained through the first stage of training. Thus, these components should already perform each task with reasonable competence, prior to their combination.

4.4 Experiments and Results

The experimental results section of this study marks the culmination of our rigorous research efforts, as we present a comprehensive account of our previously proposed models obtained through systematic experimentation across the previously defined metrics, comparing them against ablated model versions. Our investigation is structured into two subsections: Classification 4.4.1 and Quantification 4.4.2. In the first subsection, Classification 4.4.1, we unveil the results of our efforts to categorize and distinguish elements within our datasets, providing valuable insights into the emerging patterns and relationships, including the calibration evaluation of the classifier models. Following that, in the second subsection, Quantification 4.4.2, we delve into finding error associated with the prevalence estimations of the quantification models, offering a detailed analysis of the measured variables and their implications.

4.4.1 Classification

The implementation of the model relied mostly on Pytorch[2] and Huggingface[3] deep learning library. Other machine learning such as scikit-learn[4] were also used for specific tasks. The models were trained using the backpropagation algorithm (Rumelhart et al., 1986) in conjunction with the ADAMW optimization method (Loshchilov and Hutter, 2017), a variation of ADAM algorithm. An early stopping mechanism was also used, in which the training was stopped if there was no improvement in the f1-micro score in the validation set, in five continuous epochs. It was also used a linear learning rate scheduler. This creates a schedule with a learning rate that decreases linearly from the initial learning rate set in the optimizer to 0, but since we wanted a nearly constant learning rate with a slight decrease around $2e$-5/$2e$-7, we set the maximum number of epochs to a high number (300) in both training stages.

In order to gauge the importance of each proposed features and variants of the models previously outlined in Chapter 3, we preformed a critical evaluation using the appropriate performing metrics. Tables 4.4 and 4.5 presents the results for the following distinct models:

1. **Base Model (BM)**: the base architecture previously described in subsection 3.1. This model consists in a deep learning model that classify a given input text by chunks and applying a max pooling layer to consolidate the results.

2. **BM + Multiple Synonyms Attention Mechanism (MSAM)**: the segmented model described above, including the multi-synonym attention mechanism;

3. **BM + MSAM + Jointly Classify & Quantify Model (CLQ)**: the segmented model described above with the multi-synonym attention mechanism, and trained with a jointly loss function combining both classification and quantification objectives.

When it comes to the impact of the multi-synonym attention mechanism, it is clear that this extra module played a crucial role in significantly boosting performance across all metrics. Moreover, this module demonstrated its ability to excel in few-shot learning scenarios, as evident in Tables 4.4 and 4.5. Notably, the training steps required to meet the Early Stopping criterion saw a substantial decrease when this module was applied to all models across both MIMIC-III data splits, specially in the case of MIMIC-III-clean. This decrease in training steps is particularly noteworthy because it indicates the potential of few-shot learning, a key element in achieving higher results in this specific context.

[2]https://pytorch.org/
[3]https://huggingface.co/
[4]http://scikit-learn.org

Table 4.4: Results for the different classification methods on the MIMIC-III-50 test set. Results for methods marked with * were taken directly from Edin et al. (2023). Results for methods marked with † were taken directly from the corresponding paper.

Model	MIMIC-III-50									
	Stopping	AUC		Recall		Precision		F1		P@n
	Epochs	Macro	Micro	Macro	Micro	Macro	Micro	Macro	Micro	5
CNN * (Mullenbach et al., 2018)	–	89.2	91.9	–	–	–	–	58.0	64.9	62.6
Bi-GRU * (Mullenbach et al., 2018)	–	85.2	89.3	–	–	–	–	43.1	56.1	57.9
CAML * (Mullenbach et al., 2018)	–	87.5	91.1	–	–	–	–	51.0	60.6	61.1
MSATT-KG† (Xie et al., 2019)	–	91.4	93.6	–	–	–	–	63.8	68.4	64.4
MultiResCNN * (Li and Yu, 2020)	–	89.7	92.4	–	–	–	–	61.1	67.3	63.4
HiperCore† (Cao et al., 2020)	–	89.5	92.9	–	–	–	–	60.6	67.0	63.2
LAAT * (Vu et al., 2020)	–	90.5	92.8	–	–	–	–	59.2	66.8	64.0
MSMN† (Yuan et al., 2022)	–	92.8	94.7	–	–	–	–	68.3	72.5	68.0
KEPTLongformer† (Yang et al., 2022b)	–	92.6	94.8	–	–	–	–	68.9	72.9	67.3
PLM-ICD * (Huang et al., 2022)	–	91.7	93.8	–	–	–	–	65.4	70.5	65.7
BM	10(+0)	91.2	93.4	62.4	66.3	69.0	74.1	65.5	70.0	66.1
BM + MSAM	4(+10)	**93.7**	**95.4**	**71.1**	**74.4**	69.6	73.4	**70.4**	73.9	68.8
BM + MSAM + CLQ	4(+4)	**93.7**	**95.4**	**71.1**	74.1	**69.8**	**73.8**	**70.4**	**74.0**	**68.9**

Table 4.5: Results for the different classification methods on the MIMIC-III-clean test set. Results for methods marked with * were taken from Edin et al. (2023).

Model	MIMIC-III-clean										
	Stopping	AUC		Recall		Precision		F1		P@n	
	Epochs	Macro	Micro	Macro	Micro	Macro	Micro	Macro	Micro	8	15
CNN * (Mullenbach et al., 2018)	–	88.1	97.1	–	–	–	–	9.9	48.0	61.6	46.6
Bi-GRU * (Mullenbach et al., 2018)	–	91.1	97.8	–	–	–	–	12.2	49.7	62.8	47.6
CAML *Mullenbach et al. (2018)	–	91.4	98.2	–	–	–	–	20.4	55.4	67.7	52.8
MultiResCNN *Li and Yu (2020)	–	93.1	98.5	–	–	–	–	22.9	56.4	68.5	53.5
LAAT *Vu et al. (2020)	–	94.0	98.6	–	–	–	–	22.6	57.8	70.1	54.8
PLM-ICD * Huang et al. (2022)	–	95.9	98.9	–	–	–	–	26.6	59.6	72.1	56.5
BM	68(+0)	91.7	96.1	13.0	42.8	24.3	66.5	16.9	52.1	66.1	50.6
BM + MSAM	8(+5)	96.3	**98.9**	**34.5**	**66.4**	27.4	55.3	30.5	60.3	73.3	**57.5**
BM + MSAM + CLQ	8(+6)	**96.4**	**98.9**	34.3	65.0	**28.6**	**56.5**	**31.2**	**60.5**	**73.3**	57.4

In turn, although the best results were achieved with the model variant that includes the multi-synonym attention mechanism (BM+MSAM+CLQ), jointly training the classification and quantification objectives had, in fact, a negligible impact on classification accuracy.

When compared to latter proposals, our approach outperformed the previously best-performing models reported for both splits under analysis. The models reported by Edin et al. (2023) underwent an adjustment using the validation splits, as the authors reported on model performance after optimizing the decision boundary values through a grid search mechanism to maximize F1 scores in the validation splits. In contrast, our results do not involve any such adjustment, and still surpassed the best reported models to date, establishing a new state-of-the-art approach with a default decision boundary set at 0.5.

For the MIMIC-III-50 setup, the proposed approach outperforms the best reported model to date (i.e., KEPTLongFormer) across all metrics securing leading scores of 93.7 (+1.1), 95.4 (+0.6), 70.4 (+1.6), 74.0

(+1.1), and 68.9 (+1.6) in terms of macro-AUC, micro-AUC, macro-F1, micro-F1, and P@5, respectively. For the MIMIC-III-clean setup, the proposed approach outperforms the best reported model to date (i.e., PLM-ICD) across all metrics, securing leading scores of 96.4 (+0.5), 99.0 (+0.1), 31.2 (+4.6), 60.4 (+0.8), 73.3 (+1.2) and 57.4 (+0.9) regarding macro-AUC, micro-AUC, macro-F1, micro-F1, P@8, and P@15.

To explore the influence of using a different number of synonyms, we considered the BM+MSAM+CLQ model and varied M between 2, 4, or 8 synonyms over the MIMIC-III-50 dataset. Similarly to Yuan et al. (2022), our experiments showed that $M = 4$ lead to the best results, as can be observed in Table 4.6.

Table 4.6: Results of different synonyms counts (M) on MIMIC-III 50 dataset.

	AUC		**F1**		**Prec@N**
	Macro	**Micro**	**Macro**	**Micro**	**P@5**
$M = 1$	93.5	95.2	69.3	72.5	68.0
$M = 2$	93.6	95.3	69.8	73.4	68.3
$M = 4$	**93.7**	**95.4**	**70.4**	**73.9**	**68.8**
$M = 8$	93.4	95.1	69.2	72.9	68.0

We also analyzed the proposed approach in terms of calibration performance. In Table 4.7, we explicitly examine the calibration error over different sets of ICD codes: Low percentile (Low Pth) corresponds to the average value of the calibration error calculated for the 10% of ICD codes with the lowest frequency rates in the training set of the respective MIMIC-III split. In turn, medium percentile (Medium Pth) represents the average value of the calibration error for the 10% of ICD codes with medium frequency rates, falling within the 55% to 65% range in the respective MIMIC-III split training set; Finally, high percentile (High Pth) indicates the average value of the calibration error for the 10% of medical codes with the highest frequency of occurrence in the training set of the respective MIMIC-III split.

Table 4.7: Calibration metric MECE across all proposed classification models on different percentiles of MIMIC-III splits.

Dataset	**Classifier**	**Mean**	**Low Pth**	**Medium Pth**	**High Pth**
	BM	3.5e-2	2.1e-2	3.0e-2	5.1e-2
MIMIC-III-50	**BM+MSAM**	**2.7e-2**	**2.0e-2**	**2.5e-2**	**3.6e-2**
	BM+MSAM+CLQ	2.9e-2	2.1e-2	2.6e-2	3.7e-2
	BM	2.4e-3	**1.1e-4**	**8.5e-4**	16.0e-3
MIMIC-III-clean	**BM+MSAM**	**1.6e-3**	1.9e-4	**8.5e-4**	**7.7e-3**
	BM+MSAM+CLQ	**1.6e-3**	2.0e-4	8.8e-4	8.0e-3

The results show that the the label embedding mechanism that explores multiple-synonyms also offers notable benefits in terms of model calibration. The joint optimization of classification and quantification objectives failed to further improve classification performance on both MIMIC-III splits.

Besides presenting overall classification results, we also analyzed model performance across different chapters of ICD codes, using the MIMIC-III-clean split. Tables 4.8 and 4.9 provide additional insights

into our model's performance, specifically considering results with the BM+MSAM+CLQ model for codes within different ICD-9-CM diagnosis and procedure chapters.

Table 4.8: Number of instances and performance metrics for each of the ICD-9-CM diagnosis chapters. The column named "Percentage" corresponds to the percentage of the diagnosis codes under consideration over the MIMIC-III-clean test dataset.

	Occurrences				Performance metrics	
Chapter	Train	Validation	Test	Percentage	Macro-F1	Micro-F1
I	152,465	21,978	35,168	26.302%	39.97	68.96
II	9,200	1,401	2,076	1.590%	36.15	57.65
III	49,135	7,356	11,008	8.470%	33.28	60.37
IV	17,882	2,657	4,106	3.092%	30.71	41.33
V	17,392	2,562	3,740	2.973%	22.10	48.24
VI	15,811	2,433	3,397	2.715%	29.69	54.62
VII	99,076	14,729	22,526	17.107%	29.43	67.38
VIII	31,613	4,703	7,113	5.449%	36.12	59.90
IX	27,061	3,967	6,022	4.649%	31.27	56.47
X	22,940	3,438	5,260	3.970%	29.96	62.08
XI	151	24	33	0.026%	33.79	43.64
XII	6,056	888	1,371	1.043%	29.07	47.67
XIII	9,098	1,360	1,944	1.556%	28.52	51.07
XIV	2,228	328	471	0.380%	51.54	62.20
XV	12,656	1,740	2,565	2.128%	31.50	60.40
XVI	20,692	3,154	4,550	3.563%	15.96	39.75
XVII	87,280	13,018	19,131	14.986%	24.36	51.11

Chapter I (i.e., infectious and parasitic diseases) in the ICD-9-CM diagnosis codes accounts for a substantial portion of the dataset, representing 26.302% of all codes. This chapter shows impressive performance, achieving macro and micro averaged F1 scores of 39.97% and 68.96%, respectively.

Conversely, Chapter XI (i.e., complications of pregnancy, childbirth, and the puerperium) is the least frequent chapter of ICD codes, and it also corresponds to the lowest performance metrics. With a prevalence of only 0.026% in the dataset, this chapter yields macro and micro-averaged F1 scores of 33.79% and 43.64%, respectively. These scores highlight the negative impact of infrequent ICD code occurrences on the model's effectiveness.

Furthermore, we observe an interesting phenomenon in Chapter XIV (i.e., congenital anomalies). Despite representing a relatively small percentage (0.380%) of the overall dataset, the model performs

performs remarkably well in this chapter. It attains macro and micro-averaged F1 scores of 51.54% and 62.20%, respectively, empirically showing the model's ability to perform few-shot learning when dealing with seldom-seen codes.

Table 4.9: Number of instances and performance metrics for each of the ICD-9-CM procedure chapters. The column named "Percentage" corresponds to the percentage of the procedure codes under consideration over the MIMIC-III-clean test dataset.

	Occurrences				Performance metrics	
Chapter	Train	Validation	Test	Percentage	Macro-F1	Micro-F1
I	5,508	855	1,347	3.589%	36.28	65.21
II	4,852	733	1,148	3.134%	41.74	66.70
III	91	13	17	0.056%	63.07	66.67
IV	102	15	23	0.065%	57.16	60.87
V	0	0	0	0%	0.0	0.0
VI	21	3	4	0.013%	40.00	40.00
VII	501	75	104	0.317%	26.96	39.29
VIII	9,590	1,480	2,164	6.161%	37.62	63.98
IX	47,762	6,895	10,813	30.478%	45.93	76.26
X	897	127	217	0.578%	49.96	71.83
XI	15,302	2,267	3,555	9.834%	39.15	66.59
XII	1,045	152	230	0.664%	54.48	74.77
XIII	641	102	127	0.405%	74.50	69.43
XIV	201	27	43	0.126%	64.40	68.24
XV	20	3	4	0.013%	88.89	88.89
XVI	5,990	924	1,307	3.827%	44.69	60.23
XVII	2,308	318	539	1.473%	32.01	49.90
XVIII	61,329	8,568	14,455	39.267%	26.39	66.81

By examining the overall distribution of procedure codes, we see that the dataset is characterized by a generally low density of procedure codes, with two notable exceptions in Chapter IX (i.e., operations on the cardiovascular system) and Chapter XVIII (i.e., miscellaneous diagnostic and therapeutic procedures), which encompass almost 70% of the dataset. However, despite the relatively low frequency of procedures in the other chapters, our model performs exceptionally well in them. For instance, Chapters VI and XV achieve performance values of 40% and 88.89% respectively in both metrics, even though these codes have a minuscule 0.013% representation within the dataset. These results underscore the model's capacity to learn even from infrequent instances, emphasizing its few-shot learning capabilities.

Chapter XVIII in the ICD-9-CM procedure codes, which covers "miscellaneous diagnostic and therapeutic procedures," stands out as the most frequently occurring chapter in the dataset, accounting for a substantial 39.267% of the total. We achieve 26.39% for macro-averaged F1 in this chapter, and 66.81% for micro-averaged F1.

We also conducted experiments testing the classification performance of our best-performing model concerning specific ICD codes relevant in the clinical domain.

Considering the top-10 most frequent ICD-9-CM codes, Table 4.10 presents the results per code, using our best performing model. We obtained a mean precision of 75.56%, a recall of 79.34%, and an F1 score of 77.39%, i.e. results which we believe that can attest to the usefulness of our approach. In turn, Table 4.11 presents performance metrics for some relevant chronic diseases, representing some of the main focuses of health care investigation. These results again attest to the usefulness of the proposed classification method.

Table 4.10: Results for the 10 most frequent ICD-9-CM codes in the MIMIC-III-clean test dataset.

Code	Description	Precision	Recall	F1
401.9	*Unspecified essential hypertension*	75.82	86.26	80.71
38.93	*Venous Catheterization, Not Elsewhere Classified*	68.84	72.40	70.58
428.0	*Heart failure*	80.68	82.97	81.81
427.31	*Atrial fibrillation*	90.38	92.06	91.21
414.01	*Coronary atherosclerosis of native coronary artery*	81.52	86.15	83.77
96.04	*Insertion Of Endotracheal Tube*	78.36	82.13	80.20
96.6	*Enteral Infusion Of Concentrated Nutritional Substances*	69.76	78.32	73.80
99.04	*Transfusion Of Packed Cells*	65.72	59.59	62.50
584.9	*Acute kidney failure, unspecified*	72.58	69.76	71.15
250.00	*Diabetes mellitus without mention of complication type II or unspecified type, not stated as uncontrolled*	71.95	83.72	77.39
	Average	75.56	79.34	77.39

Table 4.11: Results for some relevant chronic diseases. The columns named "Unique Codes" and "Percentage" refer to the number of unique codes of the respective block within the MIMIC-III-clean test dataset, and to the corresponding percentage of occurrences.

Block	Chronic Disease	Unique codes (Present)	Percentage	Performance metrics	
				Macro-F1	Micro-F1
250	*Diabetes mellitus*	33	1.943%	29.71	65.21
401-405	*Hypertensive Disease*	14	3.303%	29.38	76.78
410-414	*Ischemic Heart Disease*	32	3.279%	31.11	68.99
428	*Heart Failure*	15	2.471%	37.19	71.53
585;403-404	*Renal Failure*	16	1.600%	35.19	58.89
490-496	*Pulmonary Disease*	16	1.209%	48.16	67.32

4.4.2 Quantification

Tables 4.12 and 4.13 show quantification test results, using both MIMIC-III splits. The results correspond to the standard Classify and Count (CC) and Probabilistic Classify and Count (PCC) methods, as well as to the use of an MLP separately trained for quantification, following the experimental setup from Coutinho and Martins (2023). In the case of BM+MSAM+CLQ, the MLP trained jointly with the classifier was used for quantification.

Analysing Table 4.12 regarding MIMIC-III-50 split, we observe that the PCC method performs less when using the model results that jointly optimize classification and quantification objectives. These results aligned with the calibration performance reported in the previous section. Additionally, we find that the joint optimization does not improve performance over the separate training of an MLP for quantification, as previously proposed by Coutinho and Martins (2023). A possible explanation relates to the fact that MIMIC-III-50 does not feature severe class imbalance issues. With a sufficient amount of data for all ICD codes, the multi-synonym attention mechanism is effective in producing well-calibrated classification outputs, leading to good quantification performance.

Regarding MIMIC-III-clean complaining a more challenge scenario (i.e more imbalanced and higher feature space), Table 4.13 shows that BM+MSAM+CLQ model outperforms all reported baselines.

Table 4.12: Results for different quantification methods, using the results from different classification models on the MIMIC-III-50 test dataset split.

Model	**CC**		**PCC**		**MLP/CLQ**	
	MAE	**MRAE**	**MAE**	**MRAE**	**MAE**	**MRAE**
BM	2.11e-02	1.08e-01	1.50e-02	9.67e-02	1.14e-02	6.83e-02
BM+MSAM	1.72e-02	9.31e-02	1.38e-02	9.31e-02	**1.09e-02**	**6.64e-02**
BM+MSAM+CLQ	1.91e-02	9.90e-02	1.69e-02	10.9e-02	1.14e-02	6.83e-02

Table 4.13: Results for different quantification methods, using the results from different classification models on the MIMIC-III-clean test dataset split.

Model	**CC**		**PCC**		**MLP/CLQ**	
	MAE	**MRAE**	**MAE**	**MRAE**	**MAE**	**MRAE**
BM	1.41e-03	3.15e-01	1.24e-03	5.59e-01	8.62e-04	5.98e-01
BM+MSAM	1.41e-03	3.32e-01	1.24e-03	5.97e-01	8.62e-4	6.43e-1
BM+MSAM+CLQ	1.41e-03	3.31e-01	1.24e-03	5.64e-01	**7.03e-04**	**4.47e-01**

Figure 4.4 shows that CLQ method outperforms PCC for nearly all ICD codes when it comes to accurately grasping the prevalence of each ICD code. For instance, ICD code 401.9, which is the most frequent in the test set, presents a high disparity in Absolute Error between PCC and CLQ results. Further investigation, revealed that despite having a high F1 score (81%), ICD code 401.9 has a notable difference between its precision score (75.8%) and recall score (86.3%). This suggests that the model tends to overestimate this class due to its high frequency, resulting in inaccurate posterior probabilities with the PCC approach. CLQ method appears to recognize this behavior and corrects it. Figure 4.5 aligns with the previous analysis.

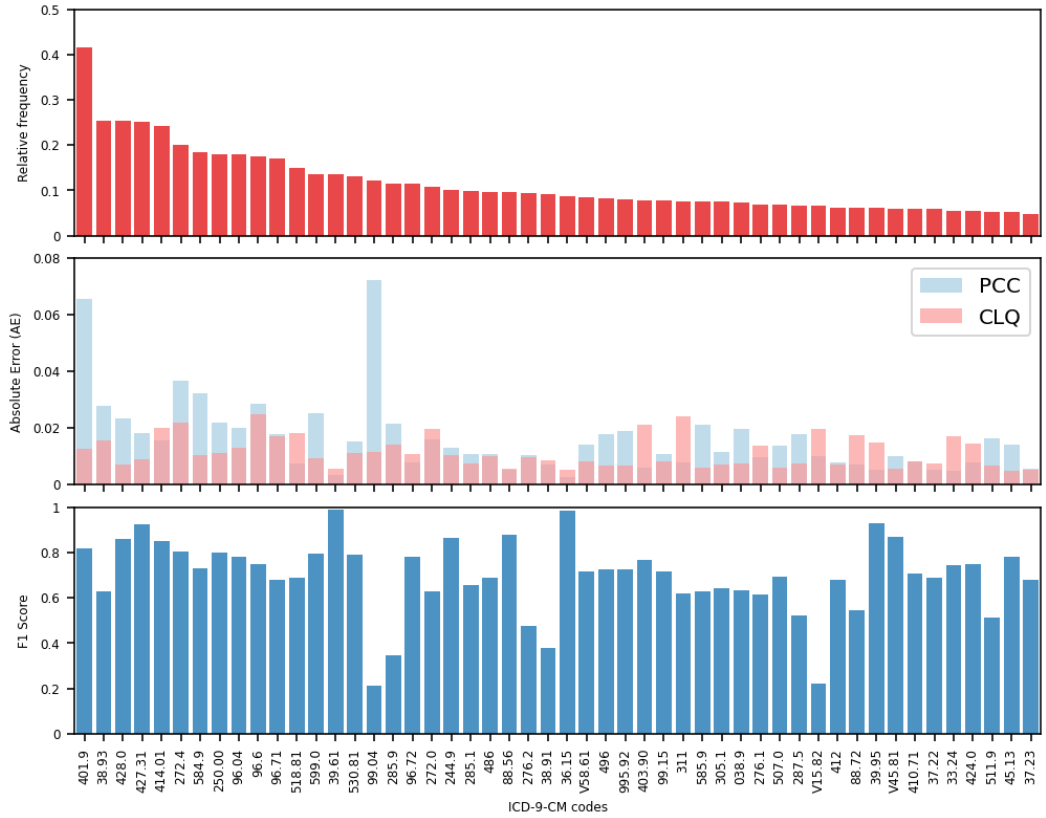


Figure 4.4: Relative frequency, Absolute Error, and F1-score for each ICD code over MIMIC-III-50.

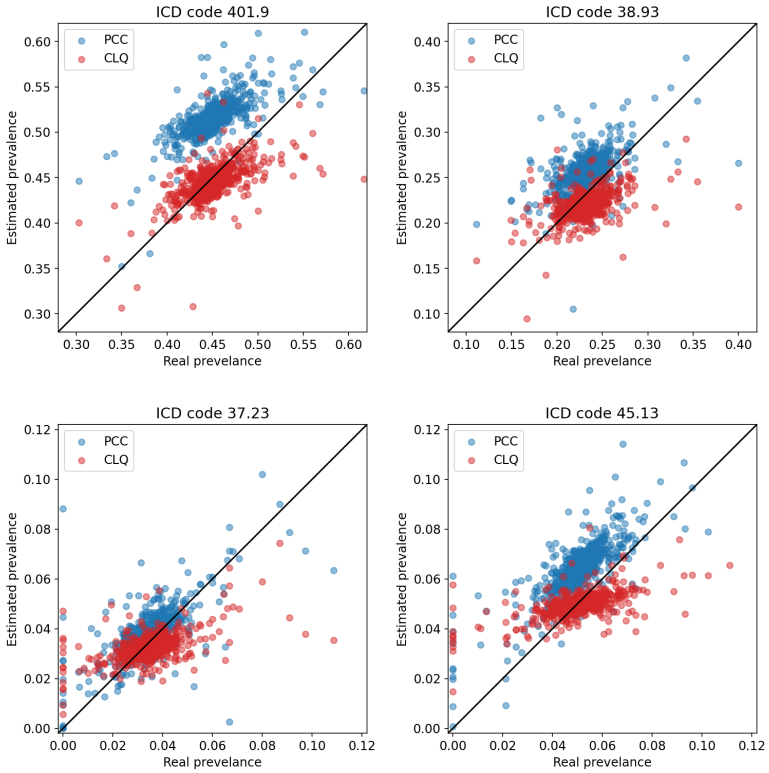


Figure 4.5: Estimated versus real prevalence for the two most frequent (top) and rarest (bottom) ICD codes in the MIMIC-III-50 dataset.

4.5 Summary

In this chapter, we dive into the experimental evaluation of our proposed approach from various angles. Firstly we introduce the dataset and its relevant splits in Section 4.1. Here, we outline the dataset splits used across training and evaluation, and provide a statistical overview of the dataset. We also tackled the primary challenges when dealing with ICD coding of hospital discharge summaries. After that, the Performance Metrics subsection (Section 4.2) elucidates the evaluation criteria adopted to rigorously analyze and compare results for both classification and quantification tasks, ensuring the robustness of our models. Lastly, the Implementation Details subsection (Section 4.3) delves deeper into the hyperparameters used in our classification and quantification models, offering insight into the fine-tuning process that underpins the success of our experimental endeavors.

Transitioning to Section 4.4, we discuss the results of the evaluation metrics employed to assess the model's ICD code predictive and relative frequency estimate capabilities.

Subsection 4.4.1 present different variants of our whole model, highlighting the significance of the components within our deep learning architecture. The standout model emerged as a fusion of different approaches such as segmentation classification and a unique label embedding mechanism based on the ICD code's synonym association with the document expressions, while wrapping all up in a unified model combining both classification and quantification tasks with a joint loss that enables training both tasks simultaneously.

Furthermore, we gauge the calibration level of our proposed model using an appropriate calibration metric known as the Mean Average Calibration Error (MECE). Our findings indicate that the BM+ MSAM model is the best-calibrated model, with the joint training process of combining both classification and quantification objectives failing in improving model's calibration.

Despite the failure in improving the model capabilities of estimating the posterior probabilities of the ICD codes, the BM+ MSAM + CLQ model showed minor improvements regarding classification task when compared to the BM+ MSAM model across nearly all evaluated metrics.

We also conduct a comparative analysis of our top-performing models against recent state-of-the-art deep learning models. This evaluation encompasses both the full-clean-label setting (MIMIC-III-clean) and a focus on the fifty most frequent codes (MIMIC-III-50), offering a comprehensive overview of the strengths and contributions of our approach. This comparative evaluation empirically establishes our best and second-best performing models as surpassing the state-of-the-art models to date in both the MIMIC-III-50 and MIMIC-III-clean scenarios, in the realm of ICD coding.

Lastly, in Subsection 4.4.1, we assessed our top-performing model's performance in predicting specific ICD code diagnoses and procedure chapters. We also examined its ability to predict the top-10 most frequent ICD-9-CM codes within the MIMIC-III dataset, as well as its in-depth predictive capabilities for certain chronic diseases within the healthcare domain.

Subsection 4.4.2 elucidates our quantification proposed model's prevalence (relative frequency) estimate capabilities, comparing its performance to established baselines using the appropriate metrics. Our model demonstrates remarkable proficiency in estimating class distributions within clinical texts, outperforming the baseline models across all evaluated metrics in MIMIC-III-clean dataset. Further assessments reveal that these results stem from the classifier's well-calibration, resulting in accurate posterior probabilities for the classes. Additionally, the CLQ method outperforms PCC for nearly all ICD codes when it comes to accurately grasping the prevalence of each ICD code. Further investigation revealed that PCC strategy tends to overestimate class posterior probabilities, and CLQ method appears to recognize this behavior and correct it. Additional plots are shown to align with the previous analysis.

Chapter 5

Conclusions and Future Work

This dissertation presented an innovative deep learning model capable of simultaneously outperforming the current state-of-the-art models for automatically assigning ICD codes to clinical text while refining the posterior probabilities outputted by the well-calibrated classifier to estimate the ICD code distribution for a group of clinical documents accounting for labeling correlations. This chapter overviews the main contributions and presents possibilities for future work.

5.1 Contributions

The revision of related work was essential to learn about state-of-the-art NLP studies, particularly concerning about automatic ICD coding, and text quantification, which served as inspiration for the present work.

The main contribution of this dissertation starts in chapter 3 by describing iteratively the line of thinking until we reach the full deep learning architecture capable of outputting remarkable results concerning clinical text classification and quantification. This chapter started by exploring the idea of segmented classification, which helps leverage the classification capabilities of compact large language models to classify large length documents that surpass by far its maximum number of input tokens. The crucial aspects that make this approach effective in our scenario is our goal to classify all codes in the document. Therefore, if a class is correctly identified in a segment (chunk) of the document, it is also correctly classified when considering the document as a whole. Moreover, we delve into the idea of integrating a label embedding mechanism (MSAM) that takes inspiration from the previous work of Yuan et al. (2022). The fundamental aspects that make this approach work in our scenario (but we believe that could be applied to almost every classification task), is the fact that each ICD code could be referred by a set of different words (synonyms). So, think of this module as a helpful ”cheat sheet” for the model during its classification task. It is like having a list of handpicked synonyms at its disposal. When the model reads a document, it can refer to this cheat sheet to assist in classifying the content. This way, the model learns the strong connections between synonyms and their corresponding ICD codes without relying on additional techniques like data augmentation. Finally, we introduced an innovative approach of integrating an auto-encoder MLP module into the classifier we previously described. This integration creates a unified model capable of simultaneously handling text classification and quantification tasks. We train this unified model using a combined loss function that addresses both tasks. The core idea behind this approach is to connect both tasks within a single model and share a joint loss function. This way, each task can gain insights from the other, improving performance across both tasks.

Chapter 4 describes and reveals the experimental evaluation of the proposed methods while com-

paring them to current state-of-the-art models. This chapter started by analyzing the structure of the dataset, mainly focusing on two well-established splits concerning the fifty most frequent ICD codes of the MIMIC-III datasets (MIMIC-III-50) and a clean version of the entire dataset (MIMIC-III-clean). After that, we described the evaluation metrics used to measure the proposed models' classification, calibration, and quantification performance, enabling further comparison to baselines and current state-of-the-art models.

Moreover, we reach the experimental results of this study, which marks the culmination of our rigorous research efforts.

Starting with the classification results, we presented the results by performing an in-depth analysis across the different variants of our whole model, highlighting the significance of the components within our proposed deep learning architecture. The standout model emerged as the previously mentioned fusion of approaches such as segmentation classification, a label embedding mechanism and a unified training process combining both classification and quantification tasks. The evaluation results revealed and confirmed previous speculations across mostly all integrated modules. The segmented classification strategy was very effective in dealing with large documents using a compact large language model. The multi-synonym attention mechanism (MSAM) that grasp the intricacies of ICD code synonyms with expressions present in clinical documents, showed already primary results surpassing the best performing state-of-the-art models across all evaluated metrics for both dataset splits. Additionally, the MSAM module revealed significant few-shot learning capabilities, which is fundamental in this study case since many classes appear very seldom across the training set.

The final model combining classification and quantification tasks into a single unified model was the best-performing proposed model across nearly all evaluated metrics and datasets. Further comparative evaluation empirically establishes our best and second-best performing proposed models, surpassing the state-of-the-art models to date in both the MIMIC-III-50 and MIMIC-III-clean scenarios. Additionally, we gauge the calibration level of our proposed models using an appropriate calibration metric known as the Mean Average Calibration Error (MECE). Our findings indicate that the second best-performing model in classification (BM+ MSAM) is the best-calibrated model, which indicates that the integration of the quantification information during the training process of the classifier though the jointly loss function failed in improving model's calibration.

Lastly still in the context of ICD code classification, we assessed our top-performing model's performance in predicting specific ICD code diagnoses and procedure chapters. We also examined its ability to predict the top-10 most frequent ICD-9-CM codes within the MIMIC-III dataset, as well as its in-depth predictive capabilities for certain chronic diseases within the healthcare domain.

Regarding text quantification, the best performing model revealed to demonstrates remarkable proficiency in estimating class distributions within clinical texts, outperforming the baseline models across all evaluated metrics in MIMIC-III-clean dataset. Further investigation revealed that PCC strategy tends to overestimate class posterior probabilities, and MLP-type strategies such as CLQ method appears to recognize this behavior and correct it.

In this dissertation, we successfully achieved our primary goals. We developed a highly accurate and well-calibrated model for assigning ICD codes to clinical documents, capable of surpassing the best-performing models reported up to date. Our model not only excelled in its accuracy regarding ICD coding but also demonstrated the ability to accurately estimate the relative frequency of ICD codes within a collection of hospital discharge summaries, accounting for label correlations. In essence, our work represents a significant advancement in clinical document coding, providing a reliable and precise tool for healthcare professionals.

5.2 Future Work

In order to improve the model performance regarding ICD coding and ICD text quantification, some options can be taken into account for future work.

One major difficulty when developing deep learning methods for large-scale multi-label text classification problems, particularly for automatic ICD coding, is predicting infrequent or unseen labels. For example, in the MIMIC-III dataset, among the 17,000 unique ICD-9-CM codes, more than 50% of them never occur in the training data. Despite the few-shot learning capabilities revealed by our proposed model, we believe that few-shot or zero-shot learning is still an under-explored area in our research, with plenty of room for improvement in future studies.

For instance, when it comes to ICD coding, we can easily see (and as stated previously in the introductory chapter (1) that the ICD coding system has a hierarchical structure nature. Therefore, including a classifier architecture that could benefit and take advantage of such structure could benefit major ICD coding improvements. A possible solution would be having two classification heads, each performing different classification tasks (one classifying the father code and the other classifying the son code). This strategy could lead to a more challenging scenario of grasping the proper posterior probability of the entire ICD code, leading to a decrease in performance when using this classifier in association with standard quantification methods such as PCC. However, we believe such an issue would be mitigated when using the Auto-Encoder MLP module of the proposed CLQ architecture to estimate the prevalence of the classes.

When considering the application of automatic ICD coding in a decision support system, explainability comes up as an import model feature. The system should be able to explain what parts of the clinical note are more relevant for each one of the assigned codes. This has the potential to increase the users' trust in automated models and also to help to identify missed and erroneous coding. These considerations are usually related with visualizations associated to the attention mechanisms, which assign importance values to specific parts in the input document and provide explanations for the assigned codes (Duarte et al., 2018; Mullenbach et al., 2018). Despite being a topic under-explored throughout this dissertation, such feature could be easily derived from the proposed model of this dissertation, due to the segmented nature of its classification module.

As pointed out recently by Searle et al. (2020), undercoding is a very common practice. In particular, these authors showed that the most frequently assigned codes in the MIMIC-III dataset are undercoded up to 35%. Thus, it is worth to adapt the current algorithms to capture missing labels.

Finally, regarding ICD text quantification, although we have seen some promising results, there are still plenty of opportunities for enhancement. One primary improvement avenue is to explore alternative methods to enhance the calibration of our classifiers. Secondly, we could also explore a different approach to enhance the training of the Multi-Layer Perceptron in the context of the joint classification and quantification training process previously described in this dissertation. This new approach involves incorporating concepts from scheduled learning and teacher forcing within the quantification loss calculation for each data instance such as: A) We will maintain a running sum of the classification predictions made up to the current point in time. B) Simultaneously, we will keep a vector containing the actual prevalence values derived from the ground truth. C) We will update both vectors by adding the estimated probabilities for the current instance and dividing the values by the total number of instances processed. D) The quantification loss will be computed using these updated vectors. Notably, the process described in step B aligns with the concept of teacher forcing. If, in step B, we combine both actual and estimated prevalence values, it resembles the idea of "scheduled sampling." The idea behind this new approach is to mitigate the adverse effects of accumulating classification errors when training the quantification module.

Bibliography

I. Mosby. *Mosby's medical dictionary*. 2006.

K. J. O'malley, K. F. Cook, M. D. Price, K. R. Wildes, J. F. Hurdle, and C. M. Ashton. Measuring diagnoses: ICD code accuracy. *Health Services Research*, 40:1620–1639, 2005.

A. N. Nguyen, D. Truran, M. Kemp, B. Koopman, D. Conlan, J. O'Dwyer, M. Zhang, S. Karimi, H. Hassanzadeh, M. J. Lawley, et al. Computer-assisted diagnostic coding: effectiveness of an NLP-based approach using SNOMED CT to ICD-10 mappings. In *AMIA Annual Symposium Proceedings*, 2018.

T. Schumacher, M. Strohmaier, and F. Lemmerich. A comparative evaluation of quantification methods. *arXiv preprint arXiv:2103.03223*, 2021.

A. Moreo, M. Francisco, and F. Sebastiani. Multi-label quantification. *arXiv preprint arXiv:2211.08063*, 2022.

R. Levin and H. Roitman. Enhanced probabilistic classify and count methods for multi-label text quantification. In *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval*, 2017.

A. Moreo and F. Sebastiani. Re-assessing the "classify and count" quantification method. In *Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28–April 1, 2021, Proceedings, Part II 43*, 2021.

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

Z. Yuan, C. Tan, and S. Huang. Code synonyms do matter: multiple synonyms matching network for automatic ICD coding. *arXiv preprint arXiv:2203.01515*, 2022.

I. Coutinho and B. Martins. Exploring label correlations for quantification of ICD codes. In *Proceedings of the International Conference on Discovery Science*, 2023.

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

J. Edin, A. Junge, J. D. Havtorn, L. Borgholt, M. Maistro, T. Ruotsalo, and L. Maaløe. Automated Medical Coding on MIMIC-III and MIMIC-IV: A Critical Review and Replicability Study. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Taipei, Taiwan, 2023.

J. Mullenbach, S. Wiegreffe, J. Duke, J. Sun, and J. Eisenstein. Explainable prediction of medical codes from clinical text. *arXiv preprint arXiv:1802.05695*, 2018.

Z. Yang, S. Wang, B. P. S. Rawat, A. Mitra, and H. Yu. Knowledge injected prompt based fine-tuning for multi-label few-shot ICD coding. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, volume 2022, 2022b.

C.-W. Huang, S.-C. Tsai, and Y.-N. Chen. PLM-ICD: automatic ICD coding with pretrained language models. *arXiv preprint arXiv:2207.05289*, 2022.

S. Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

K. D. B. J. Adam et al. A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *nature*, 323:533–536, 1986.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Proceedings of the Annual Conference on Advances in Neural Information Processing Systems*, 2017.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

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

F. Horn. Context encoders as a simple but powerful extension of word2vec. *arXiv preprint arXiv:1706.02496*, 2017.

O. Melamud, J. Goldberger, and I. Dagan. context2vec: Learning generic context embedding with bidirectional lstm. In *Proceedings of the 20th SIGNLL conference on computational natural language learning*, 2016.

M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. arxiv. *arXiv preprint arXiv:1802.05365*, 2018.

P. González, A. Castaño, N. V. Chawla, and J. J. D. Coz. A review on quantification learning. *ACM Computing Surveys (CSUR)*, 50:1–40, 2017.

A. Esuli, A. Moreo Fernández, and F. Sebastiani. A recurrent neural network for sentiment quantification. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018.

L. S. Larkey and W. B. Croft. Combining classifiers in text categorization. In *Proceeding of Special Interest Group on Information Retrieval*, 1996.

L. R. S. de Lima, A. H. F. Laender, and B. A. Ribeiro-Neto. A hierarchical approach to the automatic categorization of medical documents. In *Proceedings of Conference on Information and Knowledge Management*, 1998.

A. Perotte, R. Pivovarov, K. Natarajan, N. Weiskopf, F. Wood, and N. Elhadad. Diagnosis code assignment: models and evaluation metrics. *Journal of the American Medical Informatics Association*, 21: 231–237, 2014.

B. Koopman, G. Zuccon, A. Nguyen, A. Bergheim, and N. Grayson. Automatic ICD-10 classification of cancers from free-text death certificates. *International journal of medical informatics*, 84:956–965, 2015.

Y. Goldberg. A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, 57:345–420, 2016.

A. Prakash, S. Zhao, S. Hasan, V. Datla, K. Lee, A. Qadir, J. Liu, and O. Farri. Condensed memory networks for clinical diagnostic inferencing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.

J. Weston, S. Chopra, and A. Bordes. Memory Networks. In *Proceedings of Computing Research Repository*, 2015.

H. Shi, P. Xie, Z. Hu, M. Zhang, and E. P. Xing. Towards automated ICD coding using deep learning. *arXiv preprint arXiv:1711.04075*, 2017.

F. Duarte, B. Martins, C. S. Pinto, and M. J. Silva. Deep neural models for ICD-10 coding of death certificates and autopsy reports in free-text. *Journal of biomedical informatics*, 80:64–77, 2018.

A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov. Bag of tricks for efficient text classification. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*, 2017.

J. Lee, D. J. Scott, M. Villarroel, G. D. Clifford, M. Saeed, and R. G. Mark. Open-access MIMIC-II database for intensive care research. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2011.

X. Xie, Y. Xiong, P. S. Yu, and Y. Zhu. EHR coding with multi-scale feature attention and structured knowledge graph propagation. In *Proceedings of the 28th ACM international conference on information and knowledge management*, 2019.

F. Li and H. Yu. ICD coding from clinical text using multi-filter residual convolutional neural network. In *proceedings of the AAAI conference on artificial intelligence*, 2020.

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

X. Dai, I. Chalkidis, S. Darkner, and D. Elliott. Revisiting transformer-based models for long document classification. *arXiv preprint arXiv:2204.06683*, 2022.

Y. Li, R. M. Wehbe, F. S. Ahmad, H. Wang, and Y. Luo. Clinical-longformer and clinical-bigbird: Transformers for long clinical sequences. *arXiv preprint arXiv:2201.11838*, 2022.

M. Saeed, M. Villarroel, A. T. Reisner, G. Clifford, L.-W. Lehman, G. Moody, T. Heldt, T. H. Kyaw, B. Moody, and R. G. Mark. Multiparameter intelligent monitoring in intensive care II (MIMIC-II): a public-access intensive care unit database. *Critical care medicine*, 39:952, 2011.

J. Opitz and S. Burst. Macro f1 and macro f1. *arXiv preprint arXiv:1911.03347*, 2019.

F. Sebastiani. Evaluation measures for quantification: An axiomatic approach. *Information Retrieval Journal*, 23:255–288, 2020.

I. Loshchilov and F. Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

P. Cao, Y. Chen, K. Liu, J. Zhao, S. Liu, and W. Chong. Hypercore: Hyperbolic and co-graph representation for automatic ICD coding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

T. Searle, Z. Ibrahim, and R. J. B. Dobson. Experimental Evaluation and Development of a Silver-Standard for the MIMIC-III Clinical Coding Dataset. In *Proceedings of the Biomedical Natural Language Processing Workshop*, pages 76–85, 2020.