

<http://researchcommons.waikato.ac.nz/>

Research Commons at the University of Waikato

Copyright Statement:

The digital copy of this thesis is protected by the Copyright Act 1994 (New Zealand).

The thesis may be consulted by you, provided you comply with the provisions of the Act and the following conditions of use:

- Any use you make of these documents or images must be for research or private study purposes only, and you may not make them available to any other person.
- Authors control the copyright of their thesis. You will recognise the author's right to be identified as the author of the thesis, and due acknowledgement will be made to the author where appropriate.
- You will obtain the author's permission before publishing any material from the thesis.

Domain-specific Language Models for Multi-label Classification of Medical Text

A thesis

submitted in fulfilment

of the requirements for the Degree

of

Doctor of Philosophy

at

The University of Waikato

by

Vithya Yogarajan



THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

2022

Abstract

Recent advancements in machine learning-based medical text multi-label classifications can be used to enhance the understanding of the human body and aid the need for patient care. This research considers predicting medical codes from electronic health records (EHRs) as multi-label problems, where the number of labels ranged from 15 to 923. It is motivated by the advancements in domain-specific language models to better understand and represent electronic health records and improve the predictive accuracy of medical codes.

The thesis presents an extensive empirical study of language models for binary and multi-label medical text classifications. Domain-specific multi-sourced fastText pre-trained embeddings are introduced. Experimental results show considerable improvements to predictive accuracy when such embeddings are used to represent medical text. It is shown that using domain-specific transformer models outperforms results for multi-label problems with fixed sequence length. If processing time is not an issue for a long medical text, then TransformerXL will be the best model to use. Experimental results show significant improvements over other models, including state-of-the-art results, when TransformerXL is used for down-streaming tasks such as predicting medical codes.

The thesis considers concatenated language models to handle long medical documents and text data from multiple sources of EHRs. Experimental results show improvements in overall micro and macro F1 scores, and such improvements are achieved with fewer resources. In addition, it is shown that concatenated domain-specific transformers improve F1 scores of infrequent labels across several multi-label problems, especially with long-tail labels.

Acknowledgements

My PhD has been an exciting journey. It gives me great pleasure to thank everyone who made it possible for me.

First and foremost, my chief supervisor, Professor Bernhard Pfahringer, without whom none of this would have been possible. It took a lot for Bernhard to take me on as a PhD student, given I was a returning student with young children and from a non-computer science background. I will always be grateful for that and for believing in me. He had great faith in my abilities. It is incredible to have known someone so knowledgeable as well. Despite having a dozen PhD students under his supervision, Bernhard would never forget what each of us was doing and the details of our research. I have learnt so much from him, and it makes me proud to have been one of his PhD students.

I was very fortunate with my co-supervisors, Dr Tony Smith and Dr Jacob Montiel. Tony is probably the most caring and lovely person that I have ever known. He is there for you as great emotional support and is very down to earth. Thanks a lot, Tony, for being on my supervision team. Although Jacob only joined the supervision team later in the journey, he made it a point to ensure he was always on top of my research. He was always there, not just as a supervisor, but also as a friend. As someone who has only recently finished his PhD, he understood my stress level. He was happy to share his experience and help me be a better student.

Associate Professor Annika Hinze, the Head of School, is probably the best thing that happened in our school, during my time at Waikato University. Annika went out of her way to ensure I got the best experience possible with my PhD. Her door was always open no matter the time of day or how silly the issue was. She is always happy to listen to you and actively made sure any issues were sorted out promptly. She made having a GPU computer possible for me, making a big difference in my PhD research direction. She also got me a separate office room to work from, for which I still feel privileged. I will forever be thankful for all that Annika has done.

There have been many others who have been there for me in many ways and have always been very encouraging and supportive. Professor Eibe Frank, Professor David Bainbridge, Dr Mike Mayo, Dr Heitor Gomes, Dr Henry Gouk,

Dr Felipe Bravo, Dr Alvin Yeo, Mr Peter Reutemann, Mr Phil Treweek and Mr Tim Elphick are some of the amazing people who made my experience at the University of Waikato unique. I am thankful for each of them. I am thankful for all the technical support I got from TSG and the school administrators who were always there for all graduate students. A special thanks to Eibe for providing me access to ML servers and looking the other way when I ran multiple experiments.

My most treasured achievement in the last three years as a PhD student is the lifetime friendships that I have developed. Our coffee group - Jemma, Cameron, David, Andreea and Abigail - have been there for me as a very tight group, and I will forever be thankful for all that they have done. My PhD will not have been possible without the support from them. I do not think I have words to express how I feel, but I know they all know how much they mean to me. Over the last few years, Attaullah and I have become very close. As a fellow PhD student in ML with Bernhard who is finishing his PhD at the same time, we have shared a lot. Attaullah has also helped me a lot with my PhD research. I will miss talking to him once he returns to his home.

I would also like to take this opportunity to thank both my examiners, Prof Myra Spiliopoulou and Prof Gill Dobbie. They were both very positive and provided me with very valuable comments. Thanks a lot for taking the time to read my thesis.

Lastly, a huge hug to my hubby Rajan, who took my dream to do a PhD as his and ensured I completed one. Rajan has been there for me throughout the journey and sacrificed a lot for me to complete a PhD within three years. I would have walked out of this PhD if it were not for him, and I love him for it. Our twins, Ishan and Varan, have always been happy to video call me to say good night and sleep with Dad while I worked late nights at uni. They have also been very understanding when I spend my weekends at uni. I am grateful for such amazing kids. Huge thanks to my parents, especially my mum, who has always been there for me and her grandkids. I am lucky to have her living in New Zealand. My in-laws have always supported my career goals, and I am also thankful for them.

Contents

Abstract	iii
Acknowledgements	v
List of Figures	xi
List of Tables	xvii
1 Introduction	1
1.1 Aim	2
1.2 Challenges	3
1.2.1 Electronic Health Records Challenges	3
1.2.2 Multi-Label Classifications of EHRs and label distributions	4
1.2.3 Adaptation of NLP techniques in Healthcare	5
1.3 Contributions	6
1.4 List of Publications	7
1.5 Organisation of the Thesis	8
2 Data and Labels	10
2.1 MIMIC-III	10
2.2 eICU	11
2.3 TREC 2017	13
2.4 Medical Text Document Length	13
2.5 Pre-processing Text	14
2.6 Overview: Medical Codes	16
2.7 Predicting ICD-9 codes	17
2.8 Overview Case studies	20
2.8.1 Cardiovascular Disease	20
2.8.2 COVID-19 patient shielding	22
2.8.3 Systemic Fungal or Bacterial Infection	24
2.9 Statistics: Multi-label Problems	25

3	Background and Related Work	27
3.1	Text representations	28
3.1.1	Bag-of-Words	28
3.1.2	Word Embeddings	29
3.2	Classifiers	31
3.2.1	Traditional Classifiers	31
3.2.2	Word Embeddings-based Neural Networks	34
3.2.3	Transformers	39
3.3	Experimental Methodology	45
3.4	Evaluation Measures	46
3.4.1	F-measure	47
3.4.2	Hamming Loss	47
3.4.3	Ranking Loss	48
3.4.4	Jaccard Similarity	48
3.4.5	Statistical assessment of differences	48
3.5	Related Work	48
3.5.1	Multi-label Learning	49
3.5.2	Natural Language Processing related to Health care	50
3.5.3	Predicting Medical Codes	54
4	Domain-specific Embeddings and Binary Classification	57
4.1	Bag-of-Words (BOW)	58
4.1.1	MIMIC-III	60
4.1.2	eICU	61
4.2	Word Embeddings	61
4.2.1	Training Word Embeddings	62
4.2.2	Visualising Embeddings	63
4.2.3	Binary Classification using Embeddings	67
4.2.3.1	MIMIC-III	67
4.2.3.2	eICU	70
4.3	Transformers	71
4.3.1	MIMIC-III	71
4.3.2	eICU	72
4.4	Discussion	73
5	Word Embeddings for Multi-label Classification of Medical Text	77
5.1	ICD-9 Level 1	79
5.1.1	Comparing Multi-label Classifiers	80
5.1.2	FastText Parameter Choices	85
5.1.3	Concatenating Embeddings	87

5.1.4	Tagging Words	89
5.1.5	Comparison between Embeddings and Classifiers . . .	91
5.1.6	Summary	96
5.2	ICD-9 Level 2 and 3	97
5.2.1	Summary	103
5.3	Case Studies	103
5.3.1	Cardiovascular Disease	105
5.3.2	COVID-19 Patient Shielding	106
5.3.3	Systemic Fungal or Bacterial Infection	110
5.4	Discussion	110
6	Transformers for Multi-label Classification of Medical Text	112
6.1	ICD-9 Level 1	113
6.1.1	Medical Text Input Sequence Selections	113
6.1.2	Comparison between Transformers	115
6.1.3	Computational Time Requirements	119
6.2	ICD-9 Levels 2 and 3	122
6.3	Case Studies	127
6.3.1	Cardiovascular Disease	128
6.3.2	COVID-19 Patient Shielding	128
6.3.3	Systemic Fungal or Bacterial Infection	132
6.4	Discussion	132
7	Language Model Variations	134
7.1	Multi-BioMed-Transformers	136
7.2	Multi-CNNText	137
7.3	CNNText with Transformers	138
7.4	Input Text Data	139
7.5	Overall Performances of Language Models	140
7.5.1	Cardiovascular Disease	142
7.5.2	Systemic Fungal or Bacterial Infection	145
7.5.3	COVID-19 Patient Shielding	148
7.5.4	Levels 2 and 3 ICD-9 codes	151
7.5.5	Summary	154
7.6	Tail-end Labels	155
7.6.1	Levels 2 and 3 ICD-9 codes	156
7.6.2	Case Studies	159
7.7	Discussions	166

8	Conclusions	168
8.1	Future work	169
8.1.1	Domain-specific continuous training of TransformerXL and Longformer	170
8.1.2	Hierarchical Multi-label Classifications	170
8.1.3	HAN with Transformers and Word Embeddings	170
8.2	Final Remarks	171
A	Additional Data Information	172
B	Additional Results for Binary Classification	173
C	Additional Results for Multi-label Classification	176
D	Additional Results for Multi-label Classification using Trans- formers	180
E	Additional Results for Language Model Variations	184
	Appendices	172
	Bibliography	186

List of Figures

2.1	Sample data MIMIC-III and eICU	12
2.2	Number of tokens in MIMIC-III and eICU data.	14
2.3	ICD-9 Hierarchy with MIMIC-III and eICU label details. . . .	16
2.4	ICD-9 code frequency	17
2.5	Percentage of occurrence of levels 1,2, and 3 of ICD-9 codes for MIMIC-III	18
2.6	Percentage of occurrence of levels 1,2, and 3 of ICD-9 codes for eICU	19
2.7	Percentage frequency of labels for case studies for MIMIC-III data.	21
2.8	Percentage frequency of labels for case studies for eICU data. .	22
2.9	Cardiovascular disease medical codes.	23
2.10	COVID-19 patient shielding medical codes.	23
2.11	Systemic fungal or bacterial infection medical codes.	24
3.1	2-D Visual representation of word embeddings	28
3.2	An example of context window for the word ‘dizzy’.	29
3.3	Gated Recurrent Unit (GRU) architecture.	36
3.4	CNNText [1] architecture.	37
3.5	CAML [2] architecture.	38
3.6	HAN [3] architecture.	39
3.7	BERT-base [4] Architecture. Details of a single encoder is shown on the right hand side of the figure.	41
3.8	Example of BERT-base model for multi-label classification. . .	41

3.9	Sigmoid Function	46
4.1	Flow chart of using bag-of-words for predictions.	58
4.2	Coefficient of features for LR with BOW for category preg for MIMIC-III.	60
4.3	Visualising the most similar words obtained using health domain- specific text pre-trained word embeddings.	65
4.4	Visualising the most similar words obtained using available text pre-trained word embeddings.	66
4.5	Flow chart of using word embeddings for predictions.	67
4.6	Best F1 scores of binary classification for MIMIC-III.	74
4.7	Best F1 scores of binary classification for eICU.	75
5.1	Critical difference plots.	81
5.2	Comparison of F1 scores for the 18 top level ICD-9 groups is presented for various multi-label neural networks.	84
5.3	Critical difference plots.	84
5.4	Comparison of F1 scores for top level ICD-9 groups for embed- dings obtained by varying fastText parameters.	86
5.5	Critical difference plots.	86
5.6	A comparison of F1 scores for top-level ICD-9 groups with vari- ations of concatenations are presented.	89
5.7	Sample text split of MIMIC-III data before and after tagging.	91
5.8	A comparison of F1 scores for top-level ICD-9 groups between discharge summaries with and without a POS tagger and with text split tags.	92
5.9	Comparison of individual F1 scores of top level ICD-9 groups with various embeddings for MIMIC-III data using CNNTxt. Most frequent label on the left to the least frequent on the right.	94
5.10	Critical difference plots for MIMIC-III, ICD-9 level 1 groups.	94

5.11	Comparison of individual F1 scores of top level ICD-9 groups with various embeddings and classifiers for eICU data. Most frequent label on the left to the least frequent on the right.	96
5.12	Critical difference plots for eICU, ICD-9 level 1 groups.	96
5.13	Critical difference plots for MIMIC-III, level 2 ICD-9 codes, 158 labels.	97
5.14	Comparison of F1 scores of Level 2 ICD-9 codes for MIMIC data between CAML (T100SG) and DRCAML (T100SG).	99
5.15	Comparison of F1 scores of Level 2 ICD-9 codes for eICU data between CAML (T100SG) and DRCAML (T100SG).	101
5.16	Critical difference plots for eICU, level 2, 93 labels.	102
5.17	Critical difference plots for case studies for MIMIC-III.	105
5.18	Critical difference plots for case studies for eICU.	106
5.19	Comparison of F1 scores for cardiovascular disease.	107
5.20	Comparison of F1 scores for COVID-19 patient shielding.	108
5.21	Comparison of F1 scores for systemic fungal or bacterial infection.	109
6.1	Critical difference plots for MIMIC-III, top level ICD-9 groupings.	115
6.2	Comparison of individual F1 scores of top level ICD-9 groups with varied transformer models for MIMIC-III data.	117
6.3	Critical difference plots of label F1 scores for eICU, top level ICD-9 groupings.	118
6.4	Comparison of individual F1 scores of top level ICD-9 groups with varied transformer models for eICU data.	120
6.5	Comparison of F1 scores of level 2 ICD-9 codes for MIMIC-III data between PubMedBERT and Longformer (top) and Longformer and CAML (T100SG) (bottom).	125
6.6	Comparison of F1 scores of level 2 ICD-9 codes for eICU data between BioMed-RoBERTa-base, Longformer and DRCAML (T100SG).	126

6.7	Comparison of F1 scores for cardiovascular disease using transformers and embeddings-based neural networks.	129
6.8	Comparison of F1 scores for COVID-19 patient shielding using transformers and embeddings-based neural networks.	130
6.9	Comparison of F1 scores for systemic fungal or bacterial infection using transformers and embeddings-based neural networks.	131
7.1	TriplePubMedBERT architecture.	136
7.2	DualCNNTxt architecture.	138
7.3	CNNTxt with Transformer architecture.	139
7.4	Critical difference plots.	142
7.5	micro and macro F1 scores of language models for cardiovascular disease for varied epochs.	144
7.6	Critical difference plots.	146
7.7	micro and macro F1 scores of language models for systemic fungal or bacterial infection for varied epochs.	147
7.8	Critical difference plots.	148
7.9	micro and macro F1 scores of language models for COVID-19 patient shielding for varied epochs.	150
7.10	Critical difference plots.	150
7.11	Micro and macro F1 scores of language models for level 2 ICD-9 codes for varied epochs.	152
7.12	Micro and macro F1 scores of language models for level 3 ICD-9 codes for varied epochs.	153
7.13	Level 2 ICD-9 codes, where the difference between F1 scores of dual/triple language model variations and Longformer (3,000 tokens) is presented.	156
7.14	Heat maps for Levels 2 and 3 of ICD-9 codes.	157

7.15	Level 3 ICD-9 codes, where the difference between F1 scores of dual language model variations and Longformer (3,000 tokens) is presented.	158
7.16	Systemic fungal or bacterial infection, where the difference between F1 scores of dual/triple language model variations and TransformerXL (3,072 tokens) is presented.	160
7.17	Systemic fungal or bacterial infection, where the difference between F1 scores of dual/triple language model variations and Longformer (3,000 tokens) is presented.	161
7.18	Heat maps for systemic fungal or bacterial infections.	162
7.19	COVID-19 patient shielding, where the difference between F1 scores of multi-PubMedBERT and Longformer/TransformerXL is presented.	163
7.20	Heat maps for COVID-19 patient shielding.	164
7.21	Cardiovascular disease, where the difference between F1 scores of language model variations and Longformer/TransformerXL is presented.	165
7.22	Heat maps for cardiovascular disease.	166
A.1	Sample eICU Data as obtained from the Database.	172
C.1	Comparison of F1-scores for top level ICD-9 groups for embeddings obtained by varying fastText parameters for CBOW model, as presented in Table 5.4.	176
C.2	Critical difference plots for eICU, level 3, 316 labels.	177
C.3	Comparison of individual label F1-score of ICD-9 level 3 between BiGRU (T300SG) and CAML (T100SG) embeddings.	178
C.4	Comparison of individual label F1-score of ICD-9 level 3 between CNNTxt and CAML with T100SG embeddings.	179
D.1	Critical difference plots for MIMIC-III, level 2 ICD-9 codes, 158 labels.	180

D.2	Critical difference plots for eICU, level 2 ICD-9 codes, 93 labels.	180
D.3	Critical difference plots for eICU, level 3, 316 labels.	181
D.4	Comparison of individual label F1-score of ICD-9 level 3 between Longformer and ClinicalBERT.	181
D.5	Comparison of individual label F1-score of ICD-9 level 3 between PubMedBERT and Longformer.	182
D.6	Critical difference plots for case studies for MIMIC-III using transformers and embeddings-based neural networks.	183
D.7	Critical difference plots for case studies for eICU using transformers and embeddings-based neural networks.	183

List of Tables

2.1	Categories of MIMIC-III free-form text	11
2.2	Some variations of pre-processing options.	15
2.3	ICD-9 code level 1 groupings.	20
2.4	Statistics of multi-label classification problems.	25
3.1	Example of single-label classification is presented on the left where $Y \in \{0,1\}$. For multi-label example (right) $Y_i \subseteq L$, where each $Y_i \in \{0,1\}$	27
3.2	Classifiers used in this research. For the purpose of this thesis, classifiers are grouped together in three main groups.	32
3.3	Summary of Transformer Models.	43
3.4	Examples of embeddings based neural networks for predicting ICD-9 codes are presented.	55
4.1	F1 scores for ICD-9 groups for MIMIC-III data for BOW . . .	59
4.2	F1 scores for selected ICD-9 groups for eICU data for BOW .	61
4.3	Details of word embeddings	63
4.4	Five most similar words obtained using word embeddings pre- trained using health domain-specific text and general text. . .	64
4.5	Comparison of F1 scores using LR for predicting ICD-9 groups using embeddings with varied dimensions for MIMIC-III . . .	68
4.6	F1 scores for predicting ICD-9 groups using embeddings with LR and GRU for MIMIC-III	69

4.7	Comparison of F1 scores for predicting ICD-9 groups using embeddings with LR and GRU for eICU.	71
4.8	F1 score for ICD-9 groups for transformer models for MIMIC-III.	72
4.9	F1 score for ICD-9 groups for transformer models for eICU.	73
5.1	Overall performance of top level ICD-9 groups is presented for varying multi-label classifiers.	81
5.2	Comparison of F1 scores for the 18 top level ICD-9 groups is presented for varying multi-label classifiers.	82
5.3	Overall performance of top-level ICD-9 groups is presented various multi-label neural networks.	83
5.4	Variations of parameter choices for embeddings trained using fastText.	85
5.5	Skip-gram embeddings.	87
5.6	Comparison of overall performance of top level ICD-9 groupings among various multi-label neural classifiers with various embeddings for MIMIC-III data.	93
5.7	Comparison of overall performance of ICD-9 levels 1 among various multi-label neural classifiers with various embeddings for eICU.	95
5.8	Comparison of overall performance of ICD-9 Level 2 and 3 among various multi-label neural classifiers with various embeddings for MIMIC-III.	98
5.9	Comparison of overall performance of ICD-9 levels 2 and 3 among various multi-label neural classifiers with various embeddings for eICU.	100
5.10	Comparison of micro-F1, macro-F1 of case studies among various multi-label neural classifiers with various embeddings and sequence length for MIMIC-III data.	104

5.11	Comparison of micro-F1, macro-F1 of case studies among various multi-label neural classifiers with various embeddings and sequence length for eICU data.	104
6.1	Comparison of overall performance of top level ICD-9 groupings with various transformer models where input text sequence tokens are varied.	114
6.2	Comparison of overall performance of top level ICD-9 groupings among various transformers and sequence length for MIMIC-III.	116
6.3	Comparison of overall performance of top level ICD-9 groupings among various transformers and sequence length for eICU. . .	118
6.4	Time required (in seconds) for language models presented in Table 6.2 and Table 6.3 for Level 1 ICD-9 grouping ¹	121
6.5	Comparison of the overall performance of Levels 2 and 3 of ICD-9 codes among various transformers and sequence length for MIMIC-III.	122
6.6	Comparison of overall performance of Levels 2 and 3 of ICD-9 codes among various transformers and sequence length for eICU.	123
6.7	Comparison of micro-F1, macro-F1 of case studies among various transformer models and sequence length for MIMIC-III data.	127
6.8	Comparison of micro-F1, macro-F1 of case studies among various transformer models and sequence length for eICU data. . .	128
7.1	Summary of the experimental results presented for various models.	141
7.2	Comparison of micro-F1, macro-F1 of cardiovascular disease among various language models and input text for MIMIC-III data.	143
7.3	Comparison of micro-F1, macro-F1 of systemic fungal or bacterial infection among various language models and input text for MIMIC-III data.	145

7.4	Systemic fungal or bacterial infection, time per epoch using MIMIC-III. Results are averaged over three runs.	146
7.5	Comparison of micro-F1, macro-F1 of COVID-19 patient shielding among various language models and input text for MIMIC-III data.	149
7.6	Comparison of micro-F1, macro-F1 of levels 2 and 3 of ICD-9 codes among various language models and input text MIMIC-III data.	151
B.1	F-measure for selected ICD-9 groups for word embeddings with varied dimensions using LR with varied ridge values for MIMIC III.	173
B.2	F-measure for circ for word embeddings using random forest for MIMIC III.	174
B.3	F-measure for resp for word embeddings using random forest for MIMIC III.	174
B.4	F-measure for skin for word embeddings using random forest for MIMIC III.	175
C.1	Overall performance of top level ICD-9 groupings, 50 dimensional embeddings for varied parameters of CNNTxt.	177
E.1	Additional results for Cardiovascular disease, MIMIC-III data.	184
E.2	Cardiovascular disease, eICU data. Number of instances are 53,477 and number of labels are 15.	184
E.3	Additional results for systemic fungal or bacterial infection, MIMIC-III data.	185
E.4	Systemic fungal or bacterial infection, eICU data. Number of instances are 54,193 and number of labels are 42.	185

Chapter 1

Introduction

Recent years have seen an increase in the use of big data analytics to understand health care better and potentially improve patient care [5, 6, 7, 8]. Applying machine learning to health care can be categorised into three main groups: risk prediction, medical diagnosis and clinical decision support. Predicting treatable risk factors can improve life expectancy and quality of life, for example predicting cardiovascular events such as abdominal aortic aneurysms.

This research focuses on predictions, more specifically predicting medical codes from free-text medical text using machine learning techniques. ICD (standards for international Statistical Classification of Diseases and Related Health Problems) codes are used to classify diseases, symptoms, signs, causes of diseases, treatments and procedures. Almost all health conditions can be assigned a unique code. The Health Insurance Portability and Accountability Act (HIPAA) requires the use of ICD codes to be used by hospitals and medical facilities [9]. ICD codes facilitate billing activities, epidemiological studies and enable monitoring of health statistics and trends. However, to code electronic health records (EHR), the medical coder requires a thorough knowledge of the ICD coding systems, and it is significantly time-consuming. Thus, the ability to predict and automate medical coding is vital.

This research also considers applications where medical codes related to specific diseases or conditions are predicted from EHRs. These applications

include predicting the most common medical codes for cardiovascular disease, COVID-19 patient shielding and systemic fungal or bacterial infections are considered separate case studies.

Predicting medical codes is treated as a multi-label classification problem. Multi-label classification assigns a set of labels to an instance. For example, for predicting medical codes, each medical code is a label and each patient admitted to the hospital with available EHRs is an instance. From a collection of labels, each record will be assigned relevant medical codes/labels. Each label in multi-label classification is $\in \{\text{Yes}, \text{No}\}$.

1.1 Aim

This thesis sets out to improve upon the state-of-the-art (SOTA) for predicting medical codes from EHRs as multi-label classification problems. Research also focuses on improving the accuracy of predictions of infrequent labels from long medical documents. In order to achieve these goals, the following three research directions were formulated.

The main research directions were:

- Extensive evaluations of various multi-label problems such as predicting medical codes of the ICD-9 codes for different levels of ICD-9 hierarchy, medical codes related to several case studies: cardiovascular disease, COVID-19 patient shielding and systemic fungal or bacterial infections. Experiments were designed and modified based on the knowledge and understanding of the problems and the results obtained from various machine learning-based techniques.
- Use of domain-specific language models for improving predictive accuracy of multi-label classifiers.
- Exploration of options and new avenues for efficiently and effectively handling long EHRs and infrequent labels.

The rest of this chapter discusses the main challenges relating to the research, summarises the contributions of this thesis, and provides an overview of the thesis structure.

1.2 Challenges

Multi-label classification of a medical text presents various challenges. This section provides an outline of these challenges. Throughout this thesis, details are provided on how they are addressed.

1.2.1 Electronic Health Records Challenges

EHRs presented as free-form medical text pose many challenges due to the complexity and variations presented in the data and the legal and ethical aspects associated with the use of this data. This research defines “free-form medical text or clinical text” as text written by clinicians or other medical professionals in a clinical setting, such as hospitals. These include a personal and physiological description of patients, patient pathology, laboratory results, personal, social and medical histories, patient interviews, and diagnoses and procedure notes. These clinical notes can be a few sentences long or, in some cases, a few pages long. In many cases, these are longitudinal data where given patient clinical notes are collected over several days or even several years.

EHRs are created and utilised by different groups of people, for example, clinicians, nurses, insurance companies and researchers. EHRs are found in different forms such as structured data, free-text data, laboratory reports and images. Although data are available in abundance, the complexity and variations in the data forms create challenges associated with extracting relevant information. Depending on the person who created the data, variations in the terminology and abbreviations used will also occur. For example, the dosing instruction “QD” can be interpreted as either “once a day” or “four times a day” [10].

One of the main issues with the clinical notes is that they are sometimes ungrammatical, contain spelling errors, abbreviations and acronyms where they do not necessarily follow a particular style. The purpose of the notes also determines the language and structure used.

Another major issue with patient data is the privacy and ethical concerns associated with the datasets. Each country provides legal protections for the privacy of the patient, and ethical analysis relies on de-identified datasets which cannot be used to re-establish patient identity [11, 12, 13, 14].

These challenges with EHRs require a more careful handling and a better understanding of the data. For example, while pre-processing and tokenising EHRs, the understanding of the data, its structure and the above mentioned challenges will help decide the selection of natural language processing (NLP) techniques. Random selection of the most commonly used practices for other text data, such as legal data or movie reviews, may not be the best choice for EHRs. Moreover, the meaning of a token can vary depending on the vocabulary of the language models. For example, “ED” in health related data will most probably refer to emergency department, while in general text such as tweets this could mean several other words including “ED Sheeran”. This thesis provides details of the data used for the research and the measures taken towards pre-processing data and ensuring the meaning of the data is not compromised. This research only uses publicly available data, where patient privacy’s legal and ethical standards are protected by de-identification of the data.

1.2.2 Multi-Label Classifications of EHRs and label distributions

This research treats predicting medical codes as a multi-label problem. The underlying assumption is that each patient will be associated with more than one code. However, the multi-label classification of EHRs presents many challenges. Depending on the department or ward, some medical codes will occur

more frequently than others. Also, some chronic conditions are more common than others. For example, according to WHO, about 422 million people worldwide have diabetics, and around 32% of global death in 2019 occurred due to cardiovascular disease. These factors influence the frequency and label space of medical codes. There are thousands of ICD codes. Hence the label space is ample, but label distributions are highly unbalanced. Label space and distribution of labels present challenges in predicting infrequent labels. However, infrequent labels might be as important as frequent labels depending on each patient’s conditions. For example, a person who is a few weeks pregnant with a heart condition will be associated with labels relating to her heart condition and pregnancy-related medical codes. Label distribution of medical codes related to heart condition and pregnancy will not be the same. However, knowing she is pregnant will influence the treatment plan. Hence, the prediction of infrequent labels is equally crucial to frequent labels.

Most studies only focus on improving predictions of frequent labels [15, 16, 17]. However, in general, many patients present with many codes. Hence predicting the most frequent codes is not sufficient. There is a need for better multi-label techniques where the focus is also on infrequent labels, not only frequent ones.

1.2.3 Adaptation of NLP techniques in Healthcare

Developments in machine learning, especially deep-learning, have influenced advancements in many fields, including health applications. The rapid growth in computational power and the availability of EHRs are the main reasons for such changes. Rule-based systems have been the most popular option by health professionals, with systems such as cTAKES and MetaMap considered the leading information extraction tools [18, 19, 20, 21, 22]. However, recently there is a shift towards favouring machine learning, more specifically deep learning-based models [23, 24, 25, 25, 26, 27].

One of the main challenges in adapting the new NLP techniques, such

as transformer models, in healthcare is that EHRs can be very long. Many transformer models are restricted by a maximum sequence length that they can encode. This presents limitations, especially for predicting infrequent labels from long documents. Hence, there is a need to acknowledge limitations such as sequence length and develop methods to incorporate NLP advancements for handling long documents.

This thesis focuses on improving the accuracy of predicting medical codes and presents options for overcoming limitations such as dealing with long documents and adapting new NLP techniques.

1.3 Contributions

The major contributions from this research are:

- Domain-specific fastText pre-trained word embeddings for various dimensionalities are presented. These embeddings are used to demonstrate that word embeddings trained on health-related corpora provide an increase in accuracy compared to embeddings trained on general text for both binary and multi-label classification. Moreover, it is also shown that health-related data from multiple sources provides an additional benefit in increasing accuracy (Yogarajan et al., (2020a) [28], Yogarajan et al., (2020b) [29]).
- An extensive empirical study of multi-label classifications of medical text, where several language models are considered. Traditional classifiers, embeddings-based neural networks and transformer variations are used for classifications. Changes to language models, input text, pre-training embeddings, and transfer learning are considered to improve the overall performances of multi-label classifiers. For some multi-label problems, new state-of-the-art results are also presented (Yogarajan et al., (2020b) [29], Yogarajan et. al., (2021a) [30], Yogarajan et. al., (2022)).

- Concatenated language models are considered to handle long text documents and text data from various clinical departments or procedural summaries, and improvements in overall micro and macro F1 scores are achieved with fewer resources. In addition, concatenated multi-BioMed-Transformers improve F1 scores of infrequent labels across several multi-label problems, especially for long-tail labels (Yogarajan et al., (2021b)).

The details of the publications are provided in Section 1.4.

1.4 List of Publications

During the course of this research, the following papers have been published in peer-reviewed conference proceedings:

[Yogarajan et. al., (2022)] Yogarajan, V., Montiel, J., Smith, T., & Pfahringer, B. (2022). Predicting COVID-19 Patient Shielding: A Comprehensive Study. In Proc 34th Australasian Joint Conference on Artificial Intelligence, Sydney, Australia. Springer.

[Yogarajan et al., (2021a)] Yogarajan, V., Montiel J., Smith T., & Pfahringer B. (2021) Transformers for Multi-label Classification of Medical Text: An Empirical Comparison. In: Tucker A., Henriques Abreu P., Cardoso J., Pereira Rodrigues P., Riaño D. (eds) Artificial Intelligence in Medicine. AIME 2021. Lecture Notes in Computer Science, vol 12721. Springer, Cham. https://doi.org/10.1007/978-3-030-77211-6_12

[Yogarajan et al., (2020a)] Yogarajan, V., Gouk, H., Smith, T., Mayo, M., & Pfahringer, B. (2020). Comparing High Dimensional Word Embeddings Trained on Medical Text to Bag-of-Words for Predicting Medical Codes. In Asian Conference on Intelligent Information and Database Systems. Springer, Cham, pp. 97-108.

In addition, the following manuscripts are in submission:

[Yogarajan et al., (2020b)] Yogarajan, V., Montiel, J., Smith, T., & Pfahringer, B. (2020). Seeing The Whole Patient: Using Multi-Label Medical Text Classification Techniques to Enhance Predictions of Medical Codes. arXiv preprint arXiv:2004.00430.

[Yogarajan et. al., (2021b)] Yogarajan, V., Montiel, J., Smith, T., & Pfahringer, B. (2021). Concatenated BioMed-Transformers for Multi-label Classification of Medical Text.

1.5 Organisation of the Thesis

The chapters in this thesis are organised as follows:

Chapter 2 presents an overview and statistics of binary and multi-label classification problems that are addressed in this thesis. It also presents details of data used for pre-training embeddings and experimental evaluations. Details and summary statistics of the labels are presented. A summary of the details on pre-processing text is also provided.

Chapter 3 presents the background of language models, neural networks and other classifiers for binary and multi-label classification problems. It also presents details on transformer models used in this research. Experimental methodology and evaluation measures used for this research are also described. It reviews the literature relating to domain-specific language models used for the classification of medical text. It also reviews multi-label techniques and biomedical natural language processing techniques used for predicting medical codes.

Chapter 4 presents a study of predicting medical codes as a binary classification problem. New pre-trained embeddings, trained for this research, are presented. These embeddings are used for experiments where the F1 scores of 18 binary classification problems are compared. Language models where medical text is represented as bag-of-words and word embeddings are compared. Transformer models are also used for binary classification of medical codes.

Chapter 5 focuses on understanding and experimenting with language models which represent medical text using fastText pre-trained embeddings and presents results for predicting medical codes as multi-label classification. This chapter provides a comparison of popular machine learning classifiers used in multi-label medical text classification.

Chapter 6 presents a study of using transformer models for predicting medical codes as multi-label problems. This chapter considers the effectiveness and limitations of transformers. State-of-the-art results are presented and are directly compared with the embeddings-based neural networks presented in Chapter 6.

Chapter 7 considers the option of using multiple variations of concatenated language models for multi-label classification with long text and multi-sources of texts while using fewer resources in comparison to transformer models such as TransformerXL and Longformer. This chapter also considers the predictive performance of classifiers for tail-end labels. Results show improvements in overall micro and macro F1 scores when concatenated BioMed-Transformer models are considered. In addition, there is evidence of improvements in F1 scores for tail-end labels.

Chapter 8 provides a summary of the experimental findings and contributions in this thesis. It also presents some future research directions that follow from the work that has been undertaken in this thesis.

Chapter 2

Data and Labels

This chapter presents details of the data used for pre-training language models and in experiments for analysis. It also introduces details of various binary and multi-label classification problems used throughout the thesis. Overall summary statistics of the data and labels are also presented. Moreover, this chapter summarises the pre-processing options considered for the electronic health records (EHRs) used in this research.

2.1 MIMIC-III

Medical Information Mart for Intensive Care (MIMIC) is one of the most extensive publicly available medical databases [31, 32, 33]. It contains de-identified health records of 49,785 adult patient admissions and 7,870 neonates admissions in critical care units at the Beth Israel Deaconess Medical Center between 2001 and 2012. The median age of the adult patient is 65.8 years, with more male patients (55.9%) than females [31, 32, 33]. There are 26 separate tables in MIMIC-III (v1.4), including demographics, laboratory test results, procedures, medications, and physician notes. The available data types include billing, descriptive, dictionary, interventions, laboratory, medications, notes, physiological information and reports.

MIMIC-III contains 15 categories of notes in free-form text. Table 2.1 outlines the available notes and percentage of occurrences for unique hospital

Category	%	Category	%	Category	%
Discharge Summary	90.34	Nursing	15.54	Rehab services	3.85
Radiology	78.0	Physician	15.4	Social work	2.39
ECG	75.71	Respiratory	6.83	Case Management	1.06
Nursing/other	59.78	General	5.43	Pharmacy	0.12
Echo	40.41	Nutrition	5.43	Consult	0.09

Table 2.1: Categories and % frequency of free-form text with unique hospital admission counts for MIMIC-III. The total unique hospital admission count is 58,361.

patient admission. More than 90% of the unique hospital admissions contains at least one discharge summary. The research presented in this thesis primarily uses discharge summaries for experiments. Text summaries of categories ECG and Radiology are used as an additional medical text in case studies (see Chapter 7).

The main issues with clinical notes are that they are ungrammatical, contain spelling errors, and are shortened with abbreviations and acronyms. Abbreviations and acronyms do not necessarily follow a particular rule. The purpose of the notes also determines the language and structure used. For example, documentation purpose reports such as patient progress reports are written differently to discharge summaries, where they are deliberately written in shorthand with codes and abbreviations. These features are also evident in Figure 2.1.

2.2 eICU

The electronic Intensive Care Unit (eICU) is a database formed from the Philips eICU program, which is a telehealth program delivering information to caregivers and, as a result, enhancing patient care [32, 34]. It contains de-identified data for more than 200,000 patient admitted to ICUs in 2014

MIMIC-III - Discharge Summary (sample text)

Admission Date: [**2122-7-21**] Discharge Date: [**2122-7-28**]

Date of Birth: [**2040-02-27**] Sex: M

Service: Medicine

Chief Complaint:

COPD exacerbation/Shortness of Breath

History of Present Illness:

82 yo M with h/o CHF, COPD on 5 L oxygen at baseline, tracheobronchomalacia s/p stent, presents with acute dyspnea over several days, and lethargy. This morning patient developed an acute worsening in dyspnea, and called EMS. EMS found patient tachypnic at saturating 90% on 5L. Patient was noted to be tripodding. He was given a nebulizer and brought to the ER.

eICU - Drop down menu (sample text)

Admission Diagnosis |All Diagnosis | Non-operative |Diagnosis
 |Cardiovascular |Sepsis, pulmonary |Non-operative Organ Systems
 |Organ System |Cardiovascular |Was the patient admitted from the O.R.
 or went to the O.R. within 4 hours of admission? |No

Figure 2.1: Sample data of MIMIC-III (top) and eICU (bottom) obtained from the database. It includes acronyms and typos that are present in the data.

and 2015 and monitored by programs across the United States. The dataset includes vital sign measurements, documentation of care plan, the severity of illness measures, and diagnosis and treatment information. The structured data in eICU are generated through drop-down menus. The unstructured notes were removed as part of the de-identification process. However, the structured data presented in these drop-down menus contains a considerable amount of text and can be concatenated to form pre-processed medical text. Each unique patient’s eICU data has text input in tables, including treatment, admissions, plans and goals for care, and patient’s history. These text data are concatenated into one string. Figure 2.1 provides an example of concatenated eICU texts. See Appendix Figure A.1 for sample data generated through drop-down menu before concatenated.

2.3 TREC 2017

The TREC precision medicine/clinical decision support track 2017 (TREC 2017) [35] provides a considerable corpus of health-related free-form text. This includes 26.8 million published abstracts of medical literature listed on PubMed Central, 241,006 clinical trials documents, and 70,025 abstracts from recent proceedings focused on cancer therapy from American Association for Cancer Research (AACR) and American Society of Clinical Oncology (ASCO). The dataset from the TREC 2017 competition is used here for pre-training language models.

2.4 Medical Text Document Length

Figure 2.2 presents word frequencies for MIMIC-III and eICU. The length of discharge summary in MIMIC-III varies between 60 and 9,500 tokens with an average of 1,513 tokens. In general, radiology reports contain longer documents, while ECG summaries are shorter in comparison. The longest document in radiology is 44,613 tokens long, with the average document containing

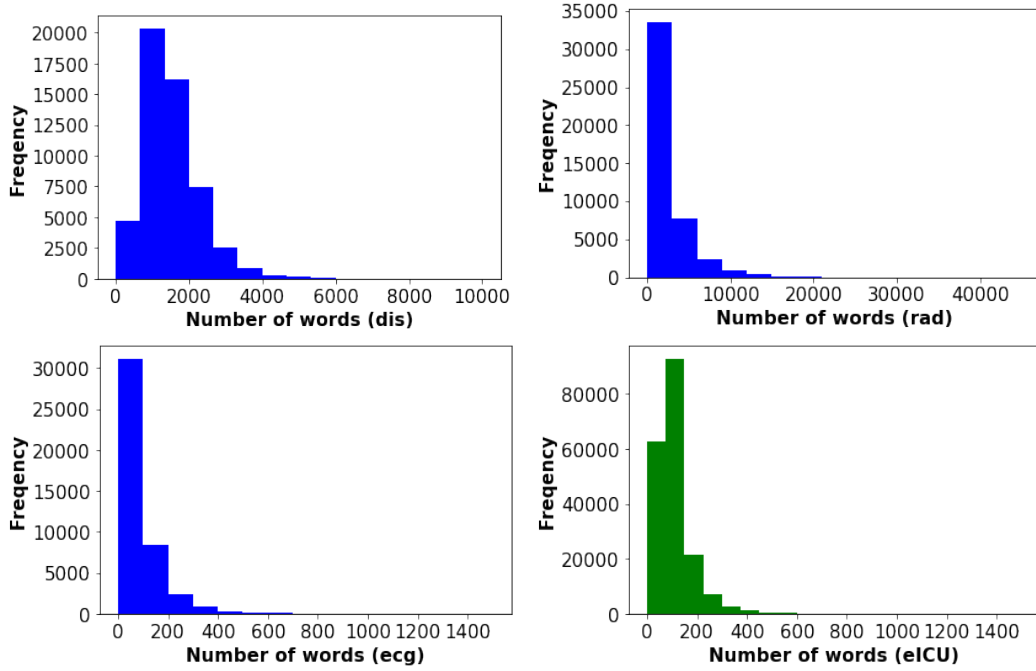


Figure 2.2: Number of tokens (or words as indicated in the figures) in MIMIC-III: discharge summary (top left, blue), radiology summary (top right, blue), ecg summary (bottom left, blue), and eICU (bottom right, green).

2,500 tokens. The maximum length of an ECG document is 1,505, and the average is 84 tokens. The word frequency in eICU data is presented in green, where the length of text input ranges from 10 to 1,374 tokens, with an average of 118 tokens. It is important to point out, the term ‘words’ used in this section refers to ‘tokens’ of a sequence.

2.5 Pre-processing Text

One of the significant issues of data mining medical text in free-form is acronyms and abbreviations. Simple changes, such as converting uppercase letters to lowercase or omitting full stops, can result in different meaning. For example, “Ab” refers to an antibody, while “AB” refers to abortion. Ideally, to maximize the use of free-form medical text “as is,” pre-processing of text should be minimized. However, for methods such as Transformers (see Chapter 3 for more details), where the maximum length of the input is usually limited to 512 tokens, an argument can be made favouring some pre-processing on the

“free-form” text.

Options	Pre-processing Options
Baseline	Text ‘as is’
Option 1	Remove newline characters, extra spaces, tabs and carriage returns
Option 2	Downcase
Option 3	Numbers: 3a Remove token that contain no alphabetic characters 3b Remove numbers and tokens that contain numbers
Option 4	Stop tokens: nltk English stopwords
Option 5	Removing punctuation
Option 6	Lemmatize
Option 7	Truncate to a maximum length
Option 8	Remove de-identified brackets. Example: FROM: <i>Please call Dr.[**Last Name (STitle) 805**] office to schedule</i> <i>a follow-up appointment within 7-10 days of discharge.</i> <i>Her office number is [**Telephone/Fax (1) 85219**].</i> TO: <i>Please call doctor office schedule follow-up appointment</i> <i>within 7-10 day discharge Her office number</i>
Option 9	Remove unique words/tokens with frequency less than 3.

Table 2.2: Some variations of pre-processing options.

Table 2.2 presents details of options for pre-processing text that were considered for experiments. For this research baseline option, and options 1, 3, 7, 8, and 9 were used. Combining various options presented in Table 2.2, the impact of pre-processing on the overall accuracy was explored. This thesis uses discharge summaries obtained from MIMIC-III “as is” (baseline option) for traditional classifiers such as logistic regression presented in the experiments for binary classifications. For other experiments, including transformers, MIMIC-III text was pre-processed by removing tokens that contain non-alphabetic characters, including all special characters and tokens that appear in fewer than three training documents (options 1, 3, 7, 8, and 9). As eICU is already pre-processed extensively, no additional pre-processing was done for this re-

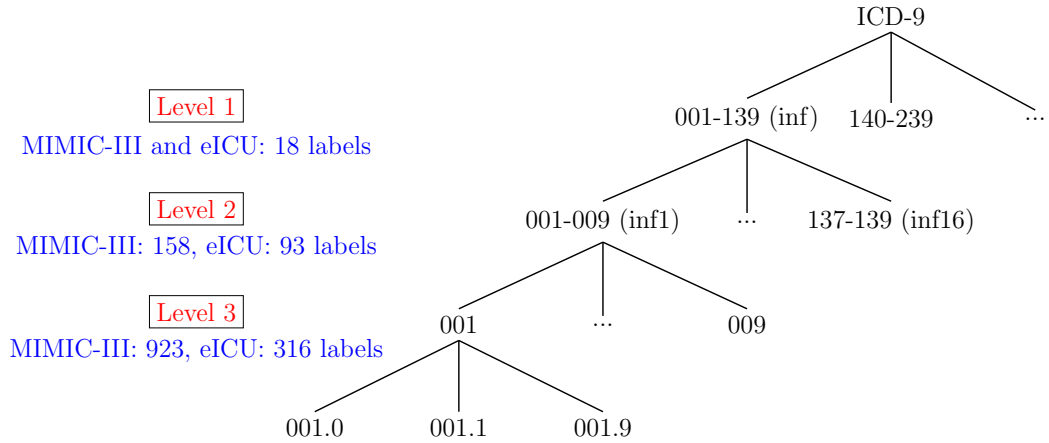


Figure 2.3: ICD-9 Hierarchy with MIMIC-III and eICU label details. All codes that occur in less than 10 unique instances are removed.

search.

2.6 Overview: Medical Codes

Medical codes, such as the International Classification of Diseases (ICD) codes, are widely used to describe diagnoses of patients [36]. Most hospitals manually assign the correct codes to patient records based on doctors’ clinical diagnosis notes. Hence, using machine learning techniques to predict ICD codes from free-form medical text and automate the medical coding process has become an important research avenue.

This research uses ICD-9 codes even though most countries have now adopted ICD-10 codes. This is primarily because MIMIC-III contains ICD-9 annotations to indicate the diagnoses and diseases of admitted patients. However, there is a clear mapping of ICD-9 codes to ICD-10 codes provided by the WHO, hence the outcome of this research can be easily adopted to other representations of medical codes, including ICD-10 codes. There are roughly 13,000 ICD-9 codes defined, with these codes following a hierarchical structure. Figure 2.3 outlines the tree structure of ICD-9. ICD-9 codes can be grouped into 18 main categories at the top level, divided into 167 sub-groups and finishing with roughly 13,000 unique codes.

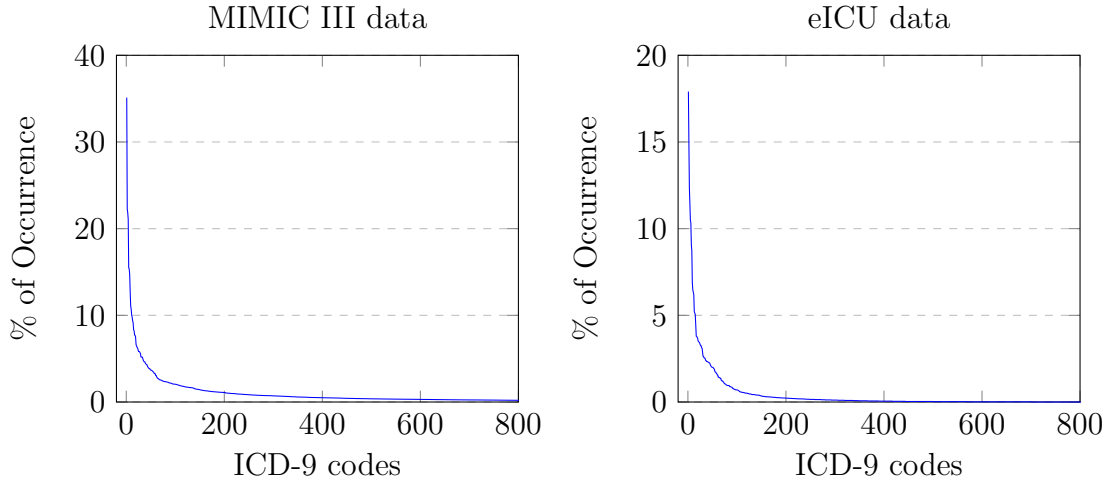


Figure 2.4: ICD-9 code frequency. Most frequent 800 ICD-9 codes: MIMIC III (left) and eICU (right).

2.7 Predicting ICD-9 codes

The research focuses on EHRs from two distinctly different large publicly available medical databases: MIMIC-III contains huge documents in a free-form medical text; eICU has concise, compressed medical data presented in a semi-structured form. Automatically predicting medical codes is the main focus of this research, due to the availability of labels in the dataset. Medical codes, such as ICD-9, are used to classify diseases, symptoms, signs, treatments, procedures and causes of diseases. Almost every health condition can be assigned a unique code¹.

There are 6,984 specific diagnosis ICD-9 codes and 2,032 specific procedure ICD-9 codes reported in MIMIC-III, among the more than 50,000 patient admission records found in this database. Patient records in MIMIC-III typically have more than one code assigned. Figure 2.4 provides an overview of the number of unique ICD-9 codes and the frequency of occurrences. It shows many ICD-9 codes occur infrequently. Recent studies which uses MIMIC-III data concentrate mainly on the top 25 or 50 most occurring ICD-9 codes [2, 37].

There are 152,614 unique patients with 914 unique individual ICD-9 codes assigned with recorded text data in eICU. As observed in MIMIC-III data,

¹<https://www.who.int/standards/classifications/classification-of-diseases>

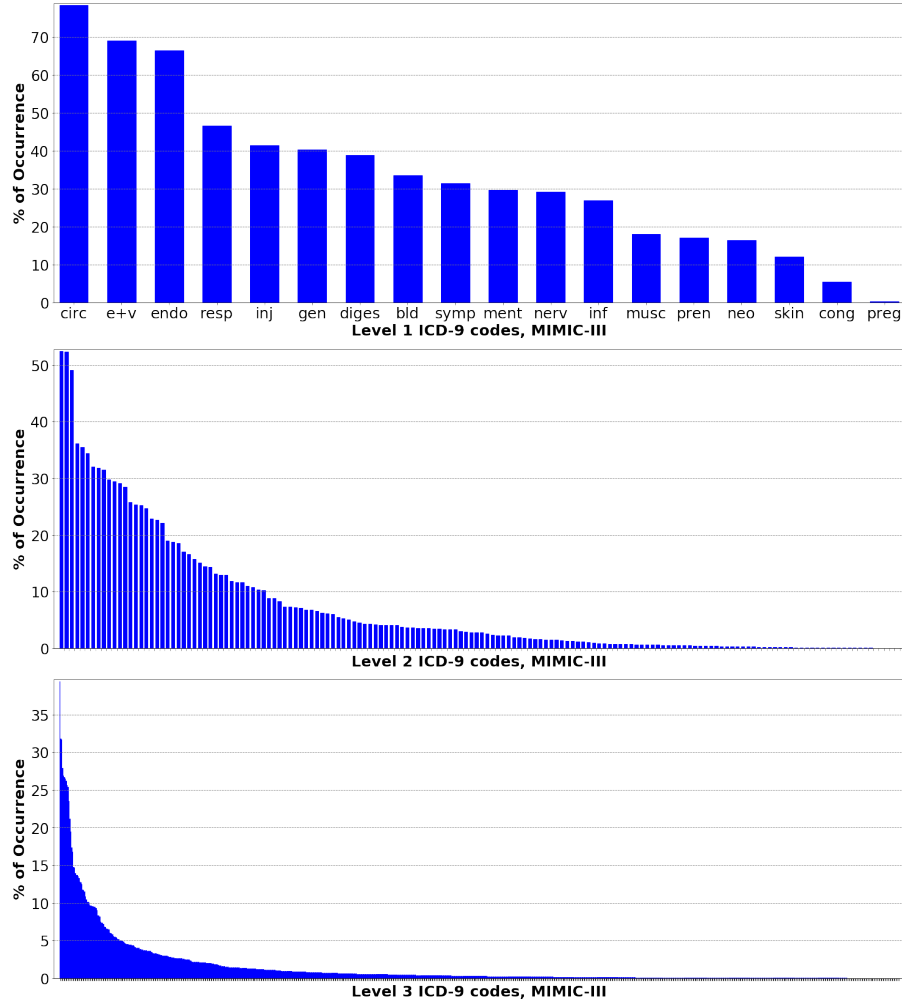


Figure 2.5: Percentage of occurrence of ICD-9 codes in unique hospital admissions in MIMIC-III: level 1 (top), level 2 (middle) and level 3 (bottom). The total number of hospital admissions with a recorded discharge summary is 52,710 for MIMIC-III.

eICU data also contain a large proportion of infrequent medical codes.

Both MIMIC-III and eICU data are obtained from patients admitted to critical care units. In general, patients are admitted to critical care units for various health complications. This results in the use of a range of medical codes, where many of the codes are infrequent. For the purpose of this research, all codes in MIMIC-III and eICU that occur in fewer than ten unique hospital admissions are removed. Consequently, as shown in Figure 2.3, MIMIC-III and eICU datasets contain 18 labels at level 1, 158 and 93 labels respectively at level 2, and 923 and 316 labels respectively at level 3. Table 2.3 provides the details of level 1 ICD-9 groups and abbreviations used in this thesis.

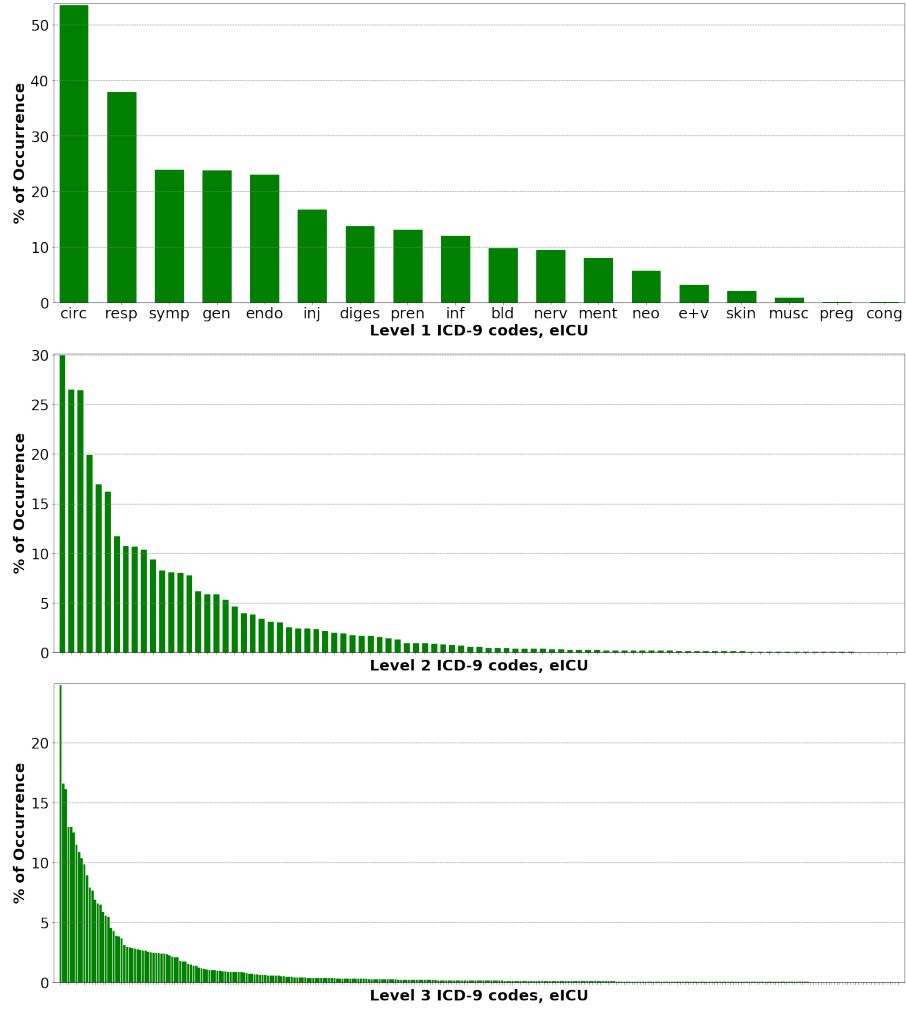


Figure 2.6: Percentage of occurrence of ICD-9 codes in unique patient stay in eICU: level 1 (top), level 2 (middle) and level 3 (bottom). The total number of unique patient stays with recorded text data is 154,882.

Figures 2.5 and 2.6 present the percentage of occurrences of level 1, 2, and 3 of ICD-9 codes for MIMIC-III and eICU respectively. Each level is considered as a flat multi-label problem and, as seen in Figures 2.5 and 2.6, at each level the labels are imbalanced. For simplicity, in this thesis the 18 categories of level 1 medical codes are initially considered separate binary classification problems. That is, each of ICD-9 codes from the MIMIC-III discharge summaries or eICU with medical text is predicted in isolation.

ICD-9 Level 1 Grouping		
001-139	infectious and parasitic diseases	inf
140-239	neoplasms	neop
240-279	endocrine, nutritional and metabolic diseases, and immunity disorders	endo
280-289	diseases of the blood and blood-forming organs	bld
290-319	mental disorders	ment
320-389	diseases of the nervous system and sense organs	nerv
390-459	diseases of the circulatory system	circ
460-519	diseases of the respiratory system	resp
520-579	diseases of the digestive system	diges
580-629	diseases of the genitourinary system	gen
630-679	complications of pregnancy, childbirth, and the puerperium	preg
680-709	diseases of the skin and subcutaneous tissue	skin
710-739	diseases of the musculoskeletal system and connective tissue	musc
740-759	congenital anomalies	cong
760-779	certain conditions originating in the perinatal period	pren
780-799	symptoms, signs, and ill-defined conditions	symp
800-999	injury and poisoning	inj
E & V	external causes of injury and supplemental classification	e+v

Table 2.3: ICD-9 code level 1 groupings.

2.8 Overview Case studies

In addition to predicting medical codes, this research considers multi-label classification of three case studies where commonly used medical codes are used as labels. This section presents an overview of these three case studies. Percentage frequency of labels for MIMIC-III and eICU for these three case studies are presented in Figures 2.7 and 2.8 respectively.

2.8.1 Cardiovascular Disease

Cardiovascular diseases are the leading cause of deaths globally and were responsible for 32% of all global deaths in 2019, i.e., 17.9 million deaths [38].

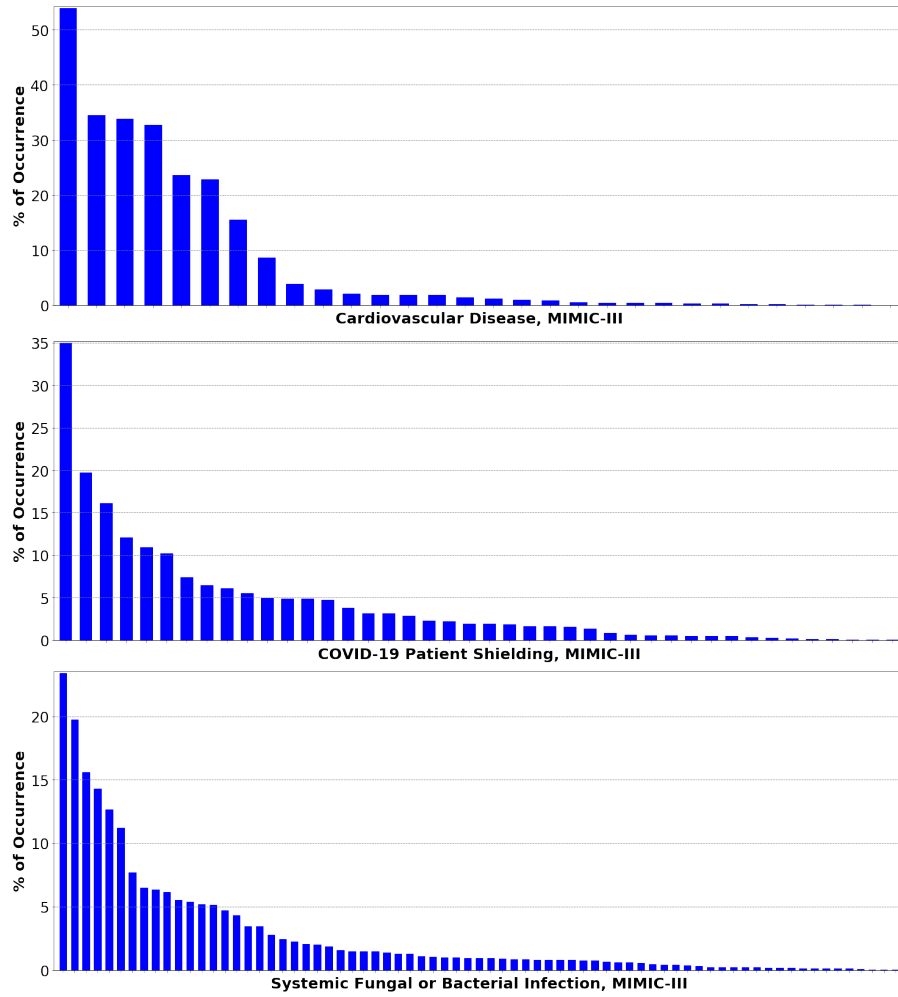


Figure 2.7: Percentage frequency of case studies cardiovascular disease (top), COVID-19 patient shielding (middle) and systemic fungal or bacterial infection (bottom) for MIMIC-III data.

American Medical Association (AMA) provides guidelines on the most commonly used medical codes associated with cardiovascular disease. Figure 2.9 presents an overview of the most common medical codes, based on AMA guidelines, associated with cardiovascular disease that are recorded for unique hospital admission with free-form medical text in MIMIC-III. Figure 2.9 also includes the level 1 ICD-9 groups associated with the codes to provide an understanding of the hierarchical grouping of the specific labels.

³<https://digital.nhs.uk/coronavirus/shielded-patient-list/methodology/annexes>

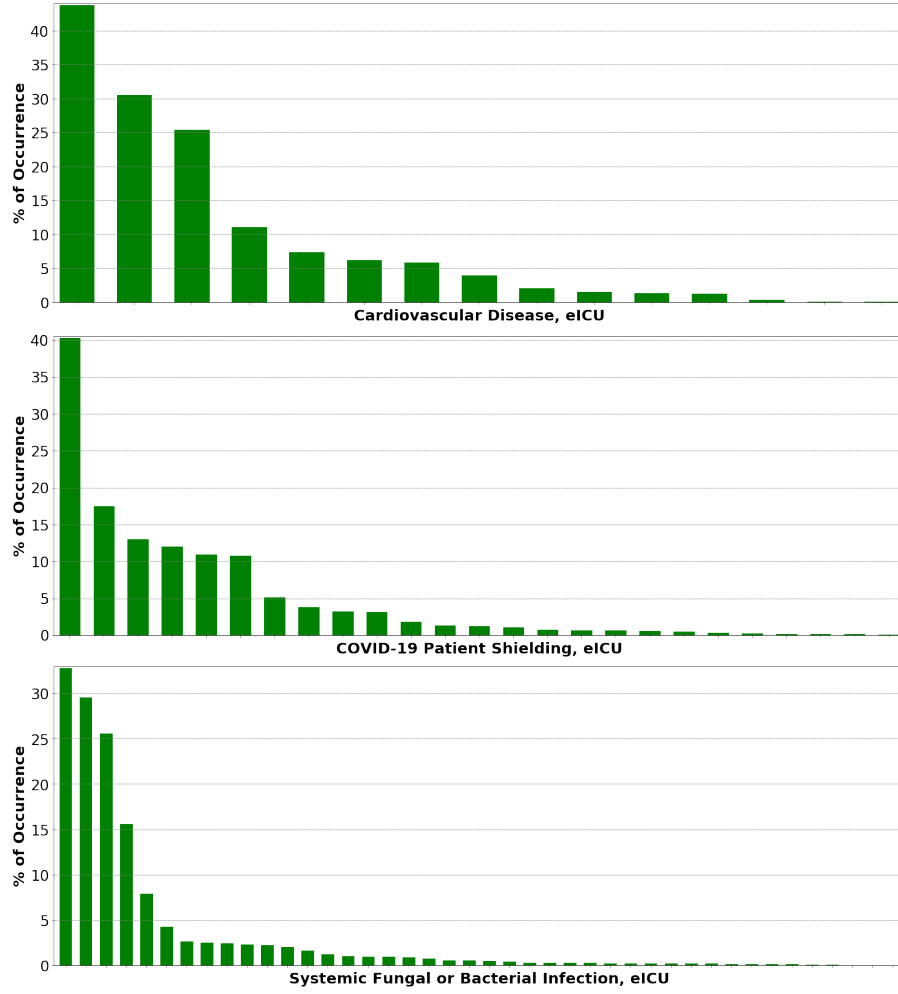


Figure 2.8: Percentage frequency of case studies cardiovascular disease (top), COVID-19 patient shielding (middle) and systemic fungal or bacterial infection (bottom) for eICU data.

2.8.2 COVID-19 patient shielding

The Coronavirus disease 2019 (COVID-19) pandemic has presented a considerable challenge to the world health care system, and the management of COVID-19 is an ongoing struggle. The ability to identify and protect high-risk groups is debated by the scientific community [39, 40]. The United Kingdom NHS Digital⁴ published a system to identify patients who meet high-risk criteria of COVID-19 and a framework to approach risk assessments [40].

This research uses such publicly published information and predicts medical codes that are listed as criteria for identifying high-risk categories from

⁴<https://digital.nhs.uk/coronavirus/shielded-patient-list/methodology/>

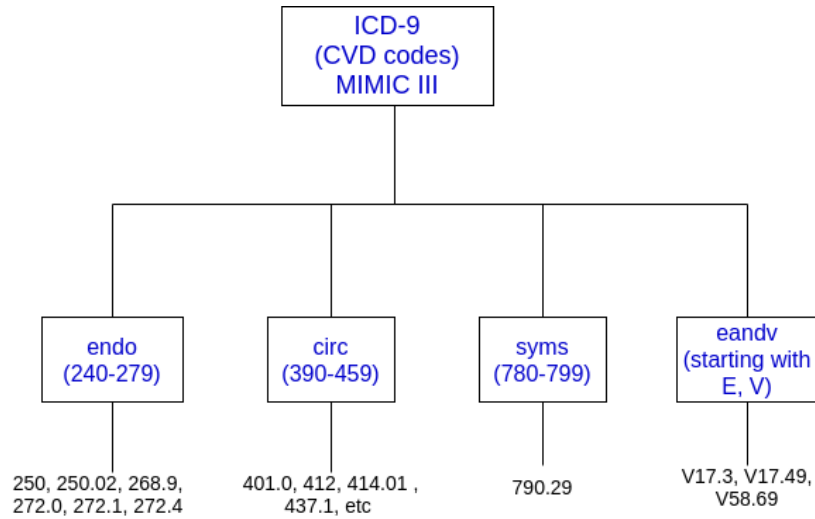


Figure 2.9: Medical codes – 30 ICD-9 codes – for cardiovascular disease (CVD) recorded for unique hospital admission with medical text in MIMIC-III.

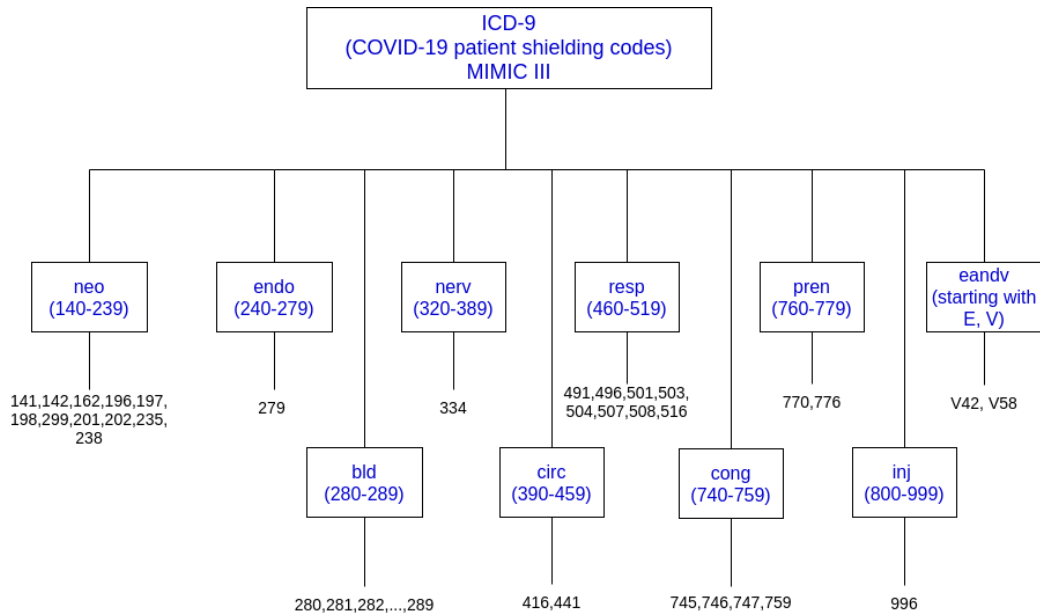


Figure 2.10: Medical codes –42 ICD-9 category codes– for COVID-19 patient shielding³ recorded for unique hospital admission with medical text in MIMIC-III.

EHRs. Due to privacy and legal issues, obtaining current patient records from hospitals is not possible. However, EHRs from MIMIC-III and eICU are used for this research. Figure 2.10 presents ICD-9 codes associated with COVID-19 patient shielding for MIMIC-III data with recorded discharge summary. Figure 2.10 also includes the level 1 ICD-9 groups associated with the codes to provide an understanding of the hierarchical grouping of the specific labels.

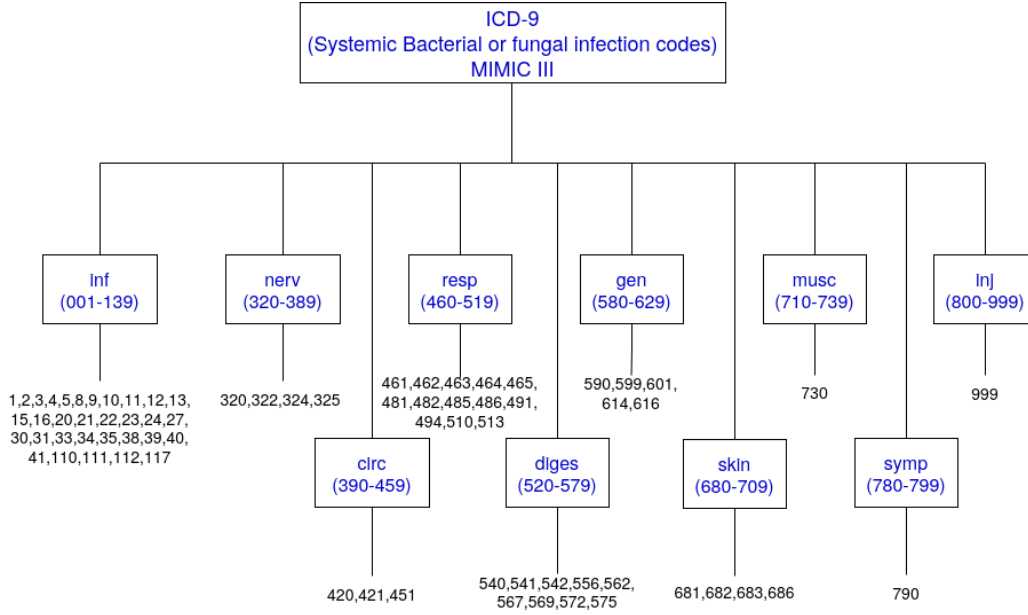


Figure 2.11: Medical codes –73 ICD-9 category codes– for systemic fungal or bacterial infection recorded for unique hospital admission with medical text in MIMIC-III.

2.8.3 Systemic Fungal or Bacterial Infection

Effectively identifying infected patients with poor hospital outcomes, such as in-hospital mortality, and managing patient care are essential aspects of health-care systems [41]. Despite the advancements in healthcare, infections play a significant role in morbidity and mortality [42], both in adults and infants born with low weight [43]. One of the causes for difficulties with infectious treatments is drug abuse. It is difficult to diagnose the fungal infections in patients whose immune systems are compromised due to drug abuse and this, in turn, leads to death [44].

This research uses the AMA guidelines to identify the most common medical codes associated with systemic fungal or bacterial infections. Figure 2.11 presents the most common ICD-9 codes for infections for unique patient hospital admission with free-form medical text in MIMIC-III. Figure 2.11 also includes the level 1 ICD-9 groups associated with the codes to provide an understanding of the hierarchical grouping of the specific labels.

2.9 Statistics: Multi-label Problems

As mentioned in this chapter, several multi-label problems are considered in this thesis. These multi-label problems are better described through information such as number of labels, number of instances, label cardinality (LCard) and label density (LDens). Table 2.4 presents a summary of the multi-label problems that are considered in this research.

For multi-label problems, the notations as per Tsoumakas et al., (2009) [45] are used:

- $L = \{\lambda_j : j = 1 \dots q\}$ refers to the finite set of labels
- $D = \{(x_i, Y_i), i = 1 \dots m\}$ refers to set of multi-label training examples

where x_i is the feature vector and $Y_i \subseteq L$ is the set of labels of the i -th example.

Classification Problem	q	m	LCard	LDens
MIMIC-III: Level 1	18	52,722	7.06	0.59
MIMIC-III: Level 2	158	52,722	11.61	0.07
MIMIC-III: Level 3	923	52,722	14.43	0.02
MIMIC-III: Cardiovascular	30	28,154	2.51	0.08
MIMIC-III: COVID-19	42	35,458	1.84	0.04
MIMIC-III: Fungal or bacterial	73	30,814	2.06	0.03
eICU: Level 1	18	154,808	2.60	0.14
eICU: Level 2	93	154,808	3.04	0.03
eICU: Level 3	316	154,808	3.37	0.01
eICU: Cardiovascular	15	53,477	1.40	0.09
eICU: COVID-19	25	34,387	1.30	0.05
eICU: Fungal or bacterial	42	54,193	1.42	0.03

Table 2.4: Statistics of multi-label classification problems.

Label cardinality ($LCard$) is the average number of labels of the examples in a dataset:

$$Label\ cardinality = \frac{1}{m} \sum_{i=1}^m |Y_i| \quad (2.1)$$

Label Density ($LDens$) is:

$$LDens = \frac{LCard}{q} \quad (2.2)$$

For single-label classification $LCard$ will always be 1 and for multi-label problems $LCard$ is > 1 . For this research, the datasets used for experiments has $LCard$ varying from 1.30 to 14.43. $LCard$ of 3.04 indicates each sample has an average of slightly more than 3 labels associated with it.

Chapter 3

Background and Related Work

Single-label classification is learning from a set of examples associated with a single class label λ from known class labels L , where $|L| > 1$. In the case of binary classification, the class labels are $|L| = 2$. In a multi-label classification problem, the examples are associated with a set of labels $Y \subseteq L$ [46, 45]. The difference between single-label and multi-label problems is shown below.

Table 3.1: Example of single-label classification is presented on the left where $Y \in \{0, 1\}$. For multi-label example (right) $Y_i \subseteq L$, where each $Y_i \in \{0, 1\}$.

X_1	X_2	X_3	X_4	Y	X_1	X_2	X_3	X_4	Y_1	Y_2	Y_3
1	0.1	0	0	1	1	0.1	4	0	1	0	0
0	0.2	1	1	0	0	0.2	1	1	0	1	1
1	0.5	2	1	1	1	0.5	2	1	1	0	0
0	0.0	2	1	0	0	0.0	2	1	0	1	0
0	0.8	3	0	1	0	0.8	3	0	1	1	1
1	0.0	0	0	1	1	0.0	4	0	1	0	1

Single-labelMulti-label

For example, patient admitted to a hospital can be associated with multiple diagnoses simultaneously. In multi-label classification, the labels can include diagnoses such as “chest pain”, “cough”, “cholesterol”, “blood pressure”, “fever”, where each label will be $\in \{\text{Yes}, \text{No}\}$.

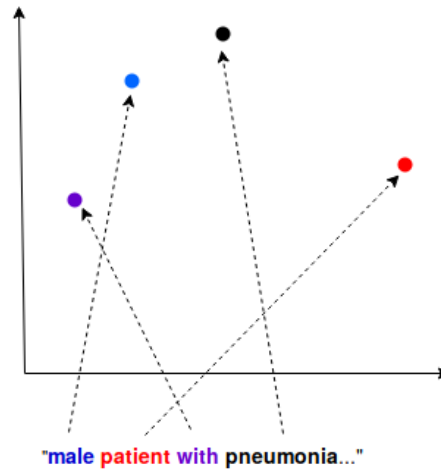


Figure 3.1: Visual representation of word embeddings, where each word is mapped to a vector. For simplicity, in this example, only a 2-D representation is used for embeddings.

This research focuses on flat multi-label classification where each level in Figure 2.3 is considered as a separate flat multi-label problem.

3.1 Text representations

An overview of bag-of-words and word embeddings is presented in this section.

3.1.1 Bag-of-Words

The bag of words (BOW) approach is a simple method for representing text and does not consider the order of words in a document. A document is represented as a sparse vector. Each element stores either the number of occurrences of a word or a binary value indicating that the word is present in the document. BOW is considered to be a relatively simple yet effective method [47, 48]. Depending on the size of the document and vocabulary selection, the resulting vector can be very sparse, making it challenging to model and extract information from the text document.

3.1.2 Word Embeddings

Embedding words in vector spaces that encode semantics has become popular in recent years [47, 49]. In general, for many NLP tasks, continuous word representations trained on large unlabelled datasets have been shown to improve performance relative to other representations [50, 51, 48]. Figure 3.1 provides a pictorial example of how these vector spaces may be organised. Word embeddings are motivated by the distributional hypothesis [52], which states there is a higher chance that words with similar meaning will occur in similar contexts. By examining a large corpus, it is possible to learn embeddings that capture the semantic similarity between words, as inferred by the contexts they are seen in. Word embeddings provide a means for effective representation learning without the complexity of deep neural networks, and can be trained efficiently on large datasets [53].

The Word2vec model uses a distributed representation of a word. There are two types of Word2vec: continuous bag-of-words (CBOW) and skip-grams [54]. The CBOW model predicts the specific word from the source context, and skip-gram does the inverse, predicting the source context from the specific word. For example, in Figure 3.2, a CBOW model would predict the word ‘dizzy’ from the words ‘woman’, ‘with’, ‘patches’, ‘visited’, while skip-gram would learn by attempting to predict the words ‘woman’, ‘with’, ‘patches’, ‘visited’ from the word ‘dizzy’. Both CBOW and skip-gram are based on feed-forward neural network architecture.

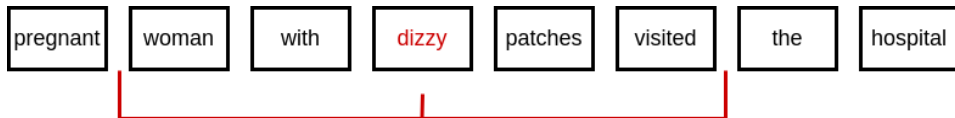


Figure 3.2: An example of context window for the word ‘dizzy’.

Formally, for a sequence of words $w_1, w_2, \dots, w_T$ with number of token T ,

context words of window c left and right of the target word, CBOW maximises:

$$\frac{1}{T} \sum_{t=1}^T \log p(w_t | \sum_{-c \leq j \leq c, j \neq 0} w_{t+j}), \quad (3.1)$$

and skip-gram maximises:

$$\frac{1}{T} \sum_{t=1}^T \sum_{-c \leq j \leq c, j \neq 0} \log p(w_{t+j} | w_t). \quad (3.2)$$

For target embeddings u_w and context embeddings v_w the probability is defined as a Softmax. For example, skip-gram will use:

$$p(w_c | w_t) = \frac{\exp(v_{w_c}^T u_{w_t})}{\sum_{w=1}^W \exp(v_w^T u_{w_t})}. \quad (3.3)$$

However, as mentioned in Mikolov et al. (2013) [55], Softmax is computationally expensive, as the complexity of the gradient is proportional to the vocabulary size of W . Solutions to this issue include Hierarchical Softmax and the more popular negative sampling loss function [55]. Hierarchical Softmax improves the computational complexity from $O(W)$ to $O(\log_2 W)$, hence reduces the number of operations needed for the algorithm. The negative samples are pulled from the distribution of $P_n(w)$, and as proven by Mikolov, et al., (2013) [55], a transformed uni-gram distribution $U(w)^{3/4}/Z$ performs better than unigram and uniform distributions on several tasks including language modeling.

FastText [50] supports both the skip-gram and CBOW methods for training word embeddings. In contrast to Word2Vec, where distinct word embeddings are learnt directly from words, fastText represents each word as a bag of character n -grams, and word embeddings are obtained by summing these character n -gram representations. For example, the tri-grams for the word “apple” are “app”, “ppl”, and “ple”. The resulting word embedding vector for “apple” will be the sum of the vectors of each of these three tri-grams. This modelling choice enables fastText to produce vectors even for novel words, out-of-vocabulary words, that were not present in the training data, as long as at least some of the n -grams have been seen before. The skip-gram loss

function used in fastText is given by:

$$\sum_{t=1}^T [\sum_{c \in C_t} l(s(w_t, w_c)) + \sum_{n \in N_{t,c}} l(-s(w_t, n))] \quad (3.4)$$

where the logistic loss function is $l : x \mapsto \log(1 + e^{-x})$, and the model is trained using a corpus T , C_t is the set of word indices surrounding a target word w_t , and $N_{t,c}$ refers to the random set of words selected from the vocabulary for each window for negative sampling. Similarity measures between two words are indicated by $s(w_t, w_c)$, where for vanilla skip-gram this will be $u_{w_t}^T v_{w_c}$.

The classification problems encountered in natural language processing typically involve predicting labels for entire documents rather than individual words. One must define a representation for documents that can be easily constructed using the embeddings learned for words. Document embeddings is obtained by computing the vector sum of the embeddings for each word in the document. This vector sum is then normalised to have length one, to ensure that documents of different lengths have representations of similar magnitudes.

3.2 Classifiers

This section presents details of the classifiers used in this research. In order to differentiate and group the classifiers, this thesis refers to logistic regression, random forest, multinomial Naive Bayes, binary relevance, classifier chains and k-nearest neighbor classifier as ‘traditional’ classifiers. Transformers play a big part throughout the thesis, and hence, are grouped and presented separately. This thesis only considers text data, hence only presents word embeddings-based neural networks. Table 3.2 presents a summary of classifier groups used in this thesis.

3.2.1 Traditional Classifiers

Logistic regression is a popular statistical model developed by David Cox in 1958 [56]. It is used to analyse datasets where one or more independent vari-

Grouping	Classifiers
'Traditional' classifiers	- Logistic Regression (LR) - Random Forest (RF) - Multinomial Naive Bayes - Binary Relevance (BR) - Classifier Chains (CC) - Multi-label k-nearest neighbor classifier (MLkNN)
Word embeddings-based	- Long Short Term Memory networks (LSTM)
Neural Networks	- Gated Recurrent Units (GRU) - CNNTxt - Convolutional Attention for Multi-Label classification (CAML) - DR-CAML - Hierarchical attention networks (HAN)
Transformers	- BERT - ClinicalBERT - PubMedBERT - RoBERTa - BioMed-RoBERTa-base - MeDAL-electra - Longformer - TransformerXL

Table 3.2: Classifiers used in this research. For the purpose of this thesis, classifiers are grouped together in three main groups.

ables determine a binary outcome. A standard logistic function is a sigmoid function where the outcome is between 0 and 1. This study uses logistic regression based on Le Cessie and Van Houwelingen (1992) [57], where the statistical model is combined with a ridge estimator. This enables the possibility of handling situations where covariates are highly correlated.

Random forest [58] creates an ensemble of multiple random decision trees that are used for prediction. Unlike decision trees, random forests are less susceptible to over-fitting as the number of trees increases. The ensemble produces predictions by aggregating the predictions of the internal members, for example, using majority vote for classification. The number of trees can

be customised.

Naive Bayes [59] is an easy to build supervised learning algorithm. It applies Bayes’ theorem with the “naive” assumption of independence, i.e. strong independence assumptions between every pair of features. One of the main advantages of the Naive Bayes algorithm is that it only requires a small amount of training data for classification. For features f_1 to f_n , multinomial Naive Bayes assumes that each $p(f_i|c)$ is a multinomial distribution. Multinomial Naive Bayes works well for data that can be turned into counts, such as word counts in a document.

The first and simplest multi-label classification algorithm is called binary relevance (BR) [60, 46]. A separate binary classification model is created for each label, such that any text with that label is a positive instance, i.e. one versus all. To predict the labels for a new text, each classifier decides if the text is in or out of the class it has been trained to recognise. The overall output for the new text is the set of all positive labels. It is essential to point out that binary relevance ignores any potential relationships between labels.

BR models make their predictions independently. However, for multi-label problems where there is a strong correlation between labels, a model could benefit from the result for another label when making its predictions. Classifier chains are constructed by training a sequence of binary classifiers—one for each label in the multi-label classification problem being considered. BR models can be ‘chained’ together into a sequence such that the predictions made by earlier classifiers are made available as additional features for the next classifier. Such a configuration is called a classifier chain (CC) [61, 62].

Denote by $h_j : \mathcal{X} \times \{0, 1\}^{j-1} \rightarrow \{0, 1\}$ the classifier corresponding to label j , where $\mathcal{X}$ is the feature space. Each classifier in the chain is applied to produce a prediction,

$$y_j^{(i)} = h_j(\vec{x}_j^{(i)}), \quad (3.5)$$

where $\vec{x}_1^{(i)}$ is a feature vector, and $\vec{x}_j^{(i)} = [\vec{x}_{j-1}^{(i)}, y_{j-1}^{(i)}]$ when $j > 1$. The square brackets represent concatenation.

The effectiveness of CC is influenced by the chaining order over the class labels. Ensembles of classifier chains built with diverse random chaining orders can help account for the effect of chaining order.

Multi-label k-nearest neighbor classifier (MLKNN) [63] is a multi-label variant of the standard k-Nearest Neighbor (kNN) algorithm that predicts the set of the most common labels among the k -nearest neighbours. In order to avoid any anomalies inside the neighbourhood, a Bayesian calibration step refines the raw predictions. An essential characteristic of this approach is its excellent scalability with respect to the number of labels: the set of nearest neighbours needs to be calculated only once for a given query text.

3.2.2 Word Embeddings-based Neural Networks

Although the basic concepts of neural networks have been around since the 1940s, the development of technologies and hardware have enabled the possibilities of building deeper and more efficient architectures [64, 47, 49]. A simple neural network consists of an input layer, hidden layer and output layer, where the values of hidden layers are inputs for the output layer. For an input of $x = [x_1, x_2, \dots, x_n]$ and weight vector $W = [w_1, w_2, \dots, w_n]$, with a bias b and the activation function g , the resulting neuron is:

$$y = g\left(\sum x_i \cdot w_i + b\right) \quad (3.6)$$

The most common activation functions include

$$\text{sigmoid function: } \sigma(x) = \frac{1}{1 + e^{-x}}$$

$$\text{rectified linear unit (ReLU): } \text{ReLU}(x) = \max(x, 0)$$

$$\text{hyperbolic tangent: } \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

The weight W and bias b are learnt while training a network, and loss functions are used to minimise the error and optimise, and are generally solved via gradient-based methods. Overfitting problems are minimised by using

techniques such as Early Stopping [65], regularisation or dropout [66] and data augmentation.

Recurrent Neural Networks (RNN) [67, 49] are a group of neural networks that are designed to handle sequential data, such as text, where the data contains complex temporal dependencies and hidden information. In contrast to feed-forward networks where information only flows in one direction, RNNs have loops that allow information from the past to be used in the future. This enables the possibility of differentiating between “patient was admitted” and “admitted to his crimes”. One significant issue with RNNs is the vanishing gradient problem. When the operation of backpropagation through time is done, the weights are multiplied many times, resulting in a small (vanishing) or big (exploding) value.

The solution to this issue with RNNs is a variant of RNNs called Long Short Term Memory networks (LSTM) [68]. LSTM consists of an innovative gating mechanism, input gate, forget gate, and output gate, ensuring a constant error flow and avoiding long-term dependency problems. The memory in LSTM is stored in an internal state, and the three gates play a vital role in deciding which information be included, added or removed from the memory. Over time, the memory cells learn which information is essential based on the weights. Gated Recurrent Units (GRU) [69] are a more simple variant of LSTM.

Gated Recurrent Units (GRU) [69] are a type of recurrent neural network. GRU networks do not include separate memory cells and have fewer parameters than LSTM networks. GRU features the additive component, allowing each unit to remember a feature in input for a more extended series of steps.

For an input vector x_t , at time t , vectors for update gate (z_t), reset gate (r_t) and output (h_t) can be calculated as follows:

$$\begin{aligned}
 z_t &= \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \\
 r_t &= \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \\
 h_t^* &= \tanh(W_h \cdot [r_t \times h_{t-1}, x_t] + b_h) \\
 h_t &= (1 - z_t) \times h_{t-1} + z_t \times h_t^*
 \end{aligned} \tag{3.7}$$

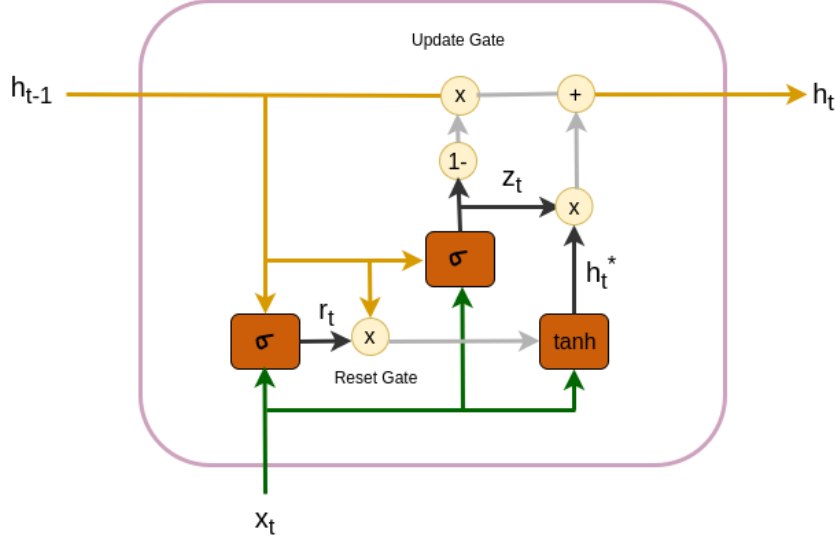


Figure 3.3: Gated Recurrent Unit (GRU) architecture.

where W_z , W_r , and W_h are weight matrices to train; b_z , b_r and b_h are biases and σ is the sigmoid activation function. Like LSTM, GRU units are also designed to handle vanishing gradients better. Bidirectional GRU (Bi-GRU) considers sequences from right to left and the reverse order. A simple architecture of GRU is presented in Figure 3.3. Here update gate helps model determine how much of the information passed from previous steps to future. Reset gate is used to determine how much of the past information will be forgotten. The ability to store and filter information through update and reset gates helps eliminate vanishing gradient problem.

CNNText [1] combines one-dimensional convolutions with a max-over-time pooling layer and a fully connected layer. If $x_{i:i+j}$ is a concatenation of words from a sentence, each word, $x_i, x_{i+1}, \dots$ is mapped to its k -dimensional embeddings using the lookup table of word embeddings. A new feature is produced using convolution operation, where a filter $w \in R^{hk}$ is applied to a window of h words. Max-over-time pooling is applied over the feature map to capture the most important feature value. In general, multiple filters are used with varying window sizes. The final prediction is made by computing a weighted combination of the pooled values and applying a sigmoid function. A simple architecture of CNNText is presented in Figure 3.4.

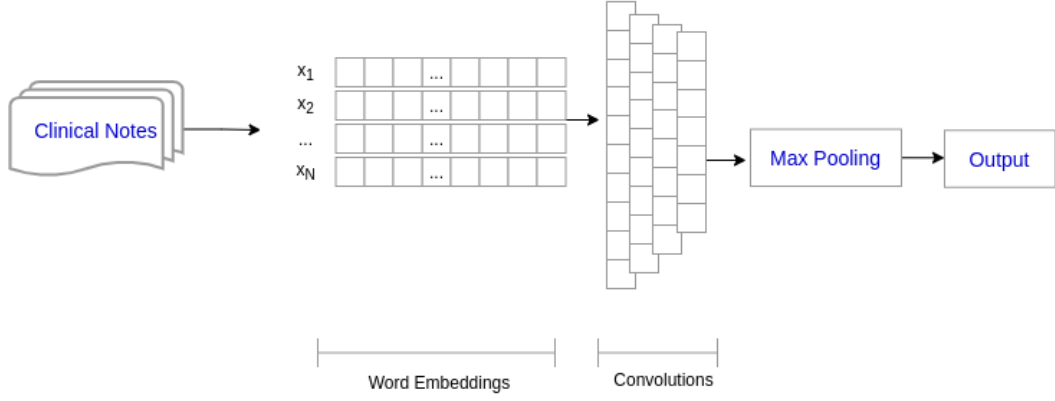


Figure 3.4: CNNTxt [1] architecture.

Mullenbach et al. (2018) [2] present Convolutional Attention for Multi-Label classification (CAML), which achieved SOTA results among the embeddings-based neural networks for predicting ICD-9 codes from MIMIC III data [70]. CAML combines convolution networks with an attention mechanism. Simultaneously, a second module is used to learn embeddings of the descriptions of ICD-9 codes to improve predictions of less frequent labels and target regularisation. For each word in a given document, word embeddings are concatenated into a matrix, and a one-dimensional convolution layer is used to combine these adjacent embeddings. The document is represented by matrix $\mathbf{H} \in R^{d_c \times N}$ where d_c is the size of convolutional filter and N is the length of the document. A per-label attention mechanism is applied, where $\mathbf{H}^T \mathbf{u}_l$ is computed for a given label l and a vector parameter $\mathbf{u}_l \in R^{d_c}$. The resulting vector is passed through a softmax operation with an output α_l . The vector representation for each label is calculated using:

$$\mathbf{v}_l = \sum_{n=1}^N \alpha_{l,n} h_n \quad (3.8)$$

The probability for l is calculated using another linear layer and a sigmoid transformation:

$$\hat{y}_l = \sigma(\beta_l^T \mathbf{v}_l + b_l), \quad (3.9)$$

for a vector of prediction weights $\beta_l \in R^{d_c}$ and scalar offset b_l . Binary cross-entropy loss and the l_2 -norm of the model weights are minimized using the

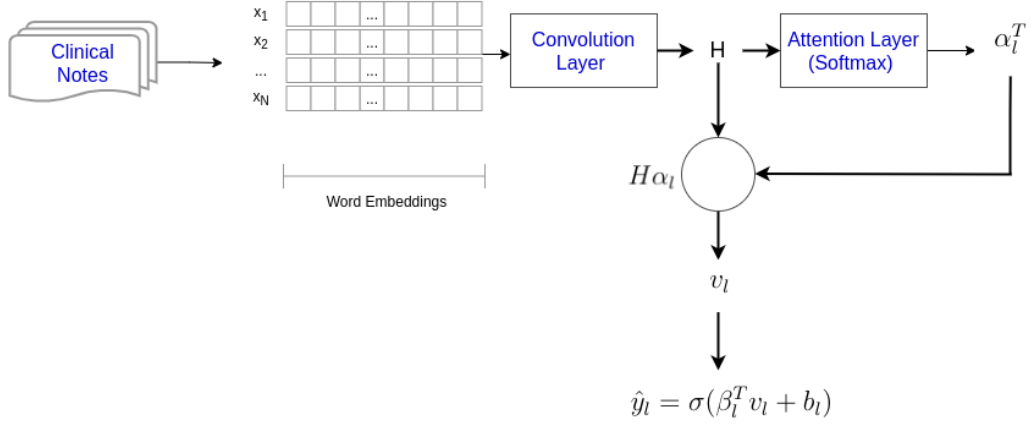


Figure 3.5: CAML [2] architecture.

Adam [71] optimizer. Figure 3.5 presents an overview of the CAML architecture. A regularising objective with a trade-off hyperparameter was added to the loss function of CAML. This variant is called Description Regularized-CAML (DR-CAML) [2].

Hierarchical Attention Networks (HAN) [3] is one of the first networks to consider the hierarchical structure of documents and mirror the structure by breaking the documents into sentences and then each sentence into words. As shown in Figure 3.6, words of each sentence are encoded, and an attention mechanism is applied, which results in a sentence vector. The same process is repeated for sentences to build the document representations, where the input is the sentence vector. The general idea of the attention mechanism is that not every word in a sentence and every sentence in a document is equally important to understand the main message. HAN incorporates two BiGRU layers with individual attention, where the first layer operates at word-level and receives word embeddings for every word in a sentence. Bahdanau-style attention [72] is used in HAN, where $\tanh$ is used as activation function:

$$\hat{a}_t = a(h_t) = \tanh(W_a h_t) \quad (3.10)$$

This is followed by a softmax function, where $\alpha = \text{softmax}(\hat{a})$. Finally, the weights are multiplied with their corresponding hidden states and the products are summed up to obtain adjusted hidden state. For example, a sentence vector

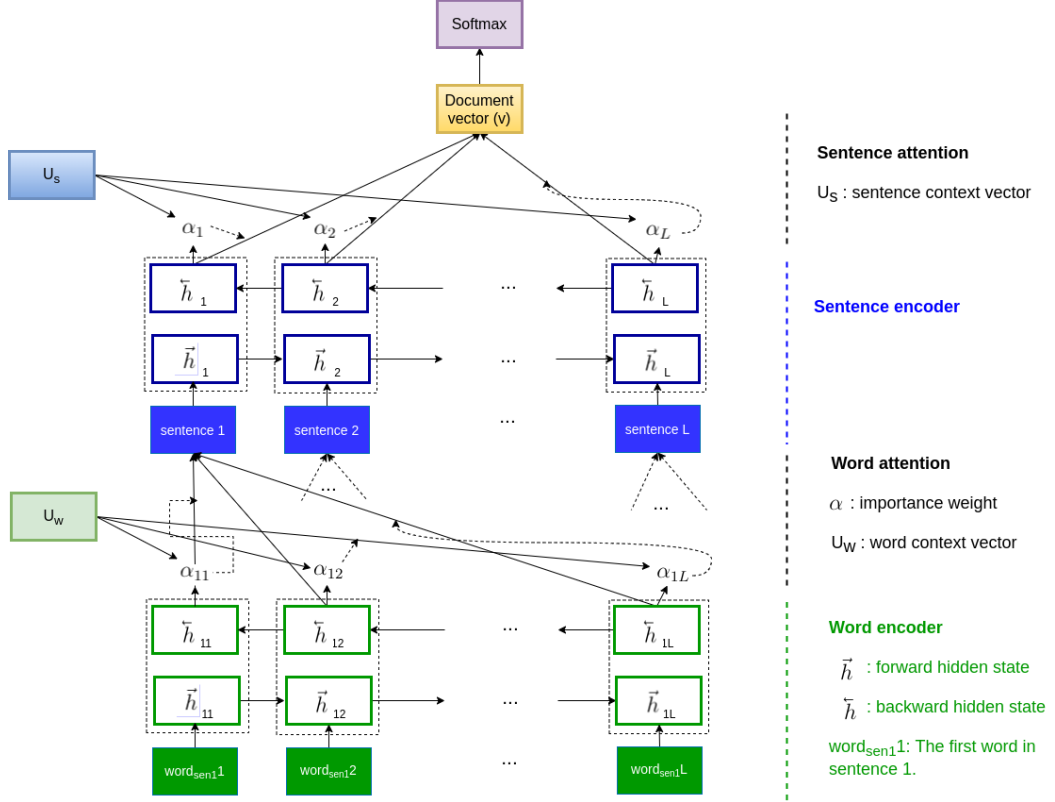


Figure 3.6: HAN [3] architecture.

in HAN is:

$$s_i = \sum_t \alpha_{it} h_{it} \quad (3.11)$$

The second BiGRU layer performs similar calculations using the sentence vector to obtain the document vector. For this research, the option of using Bidirectional LSTM instead of BiGRU as the two layers is also explored.

3.2.3 Transformers

Transformers [4] are one of the main recent developments in NLP which have achieved state-of-the-art (SOTA) results in many language tasks [73, 74, 75]. Transformers are feed-forward models based on the self-attention mechanism with no recurrence. Self-attention is a method for considering the context of a word while processing it. Similar to the sequence-to-sequence attention mechanism, self-attention is considered a soft measure where multiple words are considered. Transformers models take all the tokens in the sequence at once in parallel, enabling the capture of long-distance dependencies.

Equation 3.12 presents the definition of self-attention, where the input embeddings of sequence $x_1, x_2, \dots, x_n$ is mapped to an adapted sequence $z_1, z_2, \dots, z_n$. For every input x_i three vectors, query q_i , key k_i and value v_i , are calculated. Attention function are computed for a set of queries simultaneously and are packed together into a matrix Q . Similarly, keys and values are packed together into matrices K and V . Weights of the matrices are learnt during training, which are used to calculate these three vectors mentioned. Self-attention is also known as the scaled dot-product attention. For the linear projections Q , K , V , self-attention is [4]:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (3.12)$$

where the input queries and keys are of dimension d_k , and values of dimension d_v . Multiple weight matrices produce different query, key and value vectors for the same word, allowing the model to have multiple representations, i.e. the transformer model uses multiple attention heads [76]. Figure 3.7 presents the architecture of BERT-base where the encoder portion of the transformer is shown. Vaswani et al. (2017) [4] presents more details on transformer architecture.

Masked language modelling (MLM) and next sentence prediction (NSP) are the two most common methods used during training. MLM randomly omits words from a sentence and trains the model to predict the missing or full words based on context. NSP trains sentence pairs together, where two sentences are used as inputs. NSP then trains the model to discern whether or not the second sentence is a valid continuation of the first. Whole-word masking (WWM) is a modification of the MLM idea; however, with WWM, if a subword is chosen, then the whole word must be masked, forcing the language models to capture more contextual semantic dependencies. Electra [77] is a sample-efficient alternative to MLM, which corrupts the input by replacing some tokens with plausible alternatives sampled from a small generator network, instead of masking the input, as done in MLM.

After pre-training the model, the BERT variations are fine-tuned on domain-

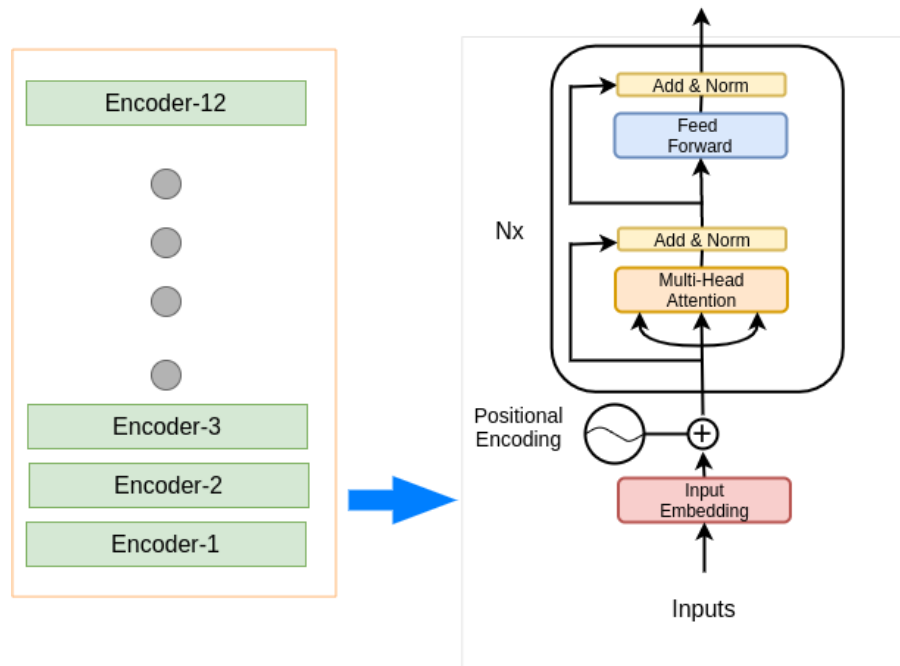


Figure 3.7: BERT-base [4] Architecture. Details of a single encoder is shown on the right hand side of the figure.

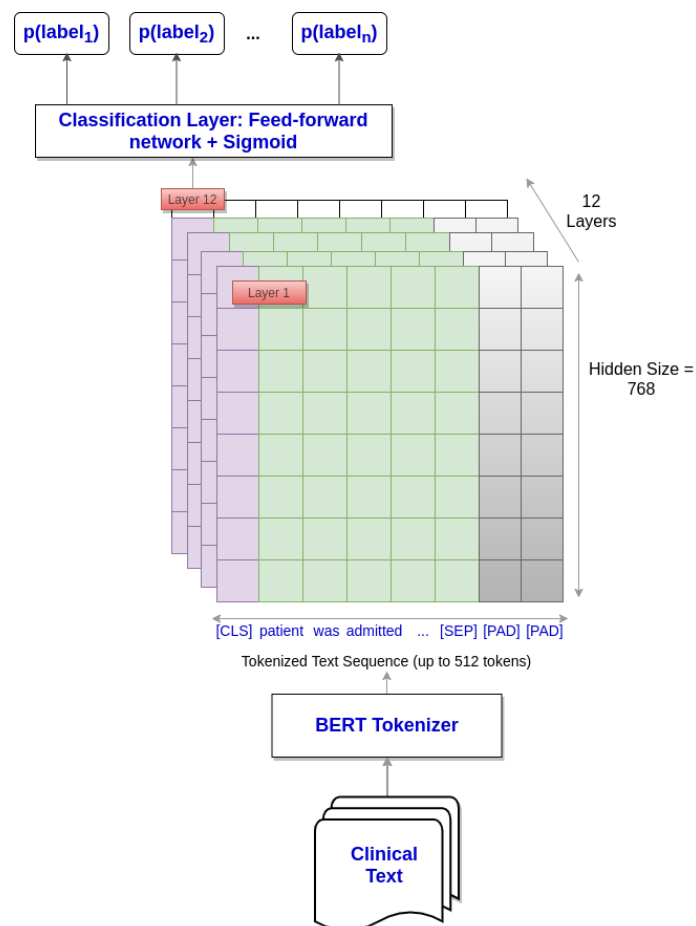


Figure 3.8: Example of BERT-base model for multi-label classification.

specific data. All the parameters are fine-tuned end-to-end. Pre-trained transformer models learn good, context-dependent ways of representing text sequences used on a specific downstream task. The models only need to fine-tune their representations to perform a particular task. When compared to pre-training transformers, fine-tuning is relatively inexpensive. Continuous training approach would initialize with the standard BERT model, pre-trained using Wikipedia and BookCorpus. It then continues the pre-training process with masked language modelling and next sentence prediction using domain-specific data. In the case of continuous training, the vocabulary is the same as the original BERT model, which is considered a disadvantage for domain-specific task [74]. For this research, PubMedBERT [74], a domain-specific BERT based model trained from scratch using biomedical related text, is used. A major advantage of using PubMedBERT is it trained on only in-domain vocabulary. A summary of Transformer models used in this thesis is presented in Table 3.3, and each is further described in the following sections.

BERT (Bidirectional Encoder Representations from Transformers) [75] is a deep neural network model that applies bidirectional training of a transformer encoder architecture [4] to language modelling. The BERT model relies on two pre-training tasks: masked language modelling and next sentence prediction. Transformer models such as BERT use WordPiece tokenization [83] to generate its vocabulary, where the tokens consist of words, subwords, and characters. This allows better handling of out-of-vocabulary (OOV) words, as OOV terms can be identified by combining WordPiece tokens. The BERT-base model is made up of 12 layers, 768 hidden units or feature numbers and 12 self-attention heads. BERT-base is neural network architecture which contains 110M parameters and is pre-trained on English Wikipedia and BookCorpus.

The ClinicalBERT [78] model follows the same model architecture as the BERT-base model [75] and was further pre-trained on all notes from MIMIC-III. Figure 3.7 presents the BERT-base architecture where 12 encoders are

¹MeDAL dataset is created from clinical abbreviations obtained from PubMed abstracts.

Transformers	Training Data	Training Details
BERT [75]	Books + Wiki (16GB)	MLM and NSP
ClinicalBERT [78]	MIMIC-III (3.7GB)	continual pre-training
PubMedBERT [74]	PubMed (21GB)	WWM, from scratch
RoBERTa [79]	Web crawl (160GB)	MLM
BioMed-RoBERTa-base [80]	Scientific papers	continual pre-training
MeDAL-electra [81]	MeDAL dataset ¹	Electra, continual pre-training
Longformer [82]	Can handle long documents. Provides computational and memory efficiency. Training Data: Books + Wiki + Realnews + Stories	
TransformerXL [73]	Can handle long documents. Models longer-term dependency by combining recurrence and relative positional encoding. Training Data: general text including Wiki	

Table 3.3: Summary of Transformer Models. Except for Longformer and TransformerXL, all other models included in this research can only handle a sequence length of up to 512 tokens.

sequentially stacked. Each encoder includes a transformer with its own attention head, as shown in the figure. Figure 3.8 presents an example of a BERT-base model used for a multi-label classification task.

PubMedBERT [74] used the same architecture as the BERT-base model. However, unlike ClinicalBERT, where the general-domain pre-trained BERT model was further pre-trained using domain-specific data, PubMedBERT is domain-specifically pre-trained from scratch to better capture the biomedical language [74]. PubMedBERT is pre-trained from scratch using abstracts from PubMed and full-text articles from PubMedCentral. This model achieves state-of-the-art performance on many biomedical natural language processing tasks and holds the top score on the Biomedical Language Understanding and Reasoning Benchmark (BLURB) [74].

RoBERTa [79] is a robustly optimised BERT approach with improved training methodology and 160GB of general-domain training data, in comparison

to the 16GB data used in BERT. RoBERTa uses dynamic masking while training, where masking patterns are generated each time a sequence is fed into the model. Also, unlike BERT models, in RoBERTa models, next sentence prediction (NSP) loss is removed, and full sentences from one or more documents with the maximum length of 512 tokens are used. Liu et al. (2019) show RoBERTa outperforming BERT models on all GLUE benchmark tasks [84]. BioMed-RoBERTa-base [80] is based on the RoBERTa-base [79] architecture. RoBERTa-base was continuously pre-trained using 2.68 million scientific papers from the Semantic Scholar corpus. Gururangan et al. (2020) [80] show that the domain-specifically pre-trained model BioMed-RoBERTa-base outperforms RoBERTa-base on biomedical domain-specific tasks.

MeDAL-electra [81] is a continually pre-trained model from Electra-small using the MeDAL dataset. The MeDAL dataset is a publically available dataset created from abbreviations used in clinical/medical domains, obtained from PubMed abstracts.

Longformer [82] is a transformer model designed to handle longer sequences without the limitation of the maximum token size of 512, set by other transformers such as the BERT model. Model was trained using Book corpus, English Wikipedia, Realnews dataset and the Stories corpus. Longformer reduces the model complexity by reformulating the self-attention computation. This modified self-attention operation scales linearly with sequence length, instead of quadratically as in the original transformer models, making it possible to handle long documents. Longformer combines attention patterns such as sliding window, dilated sliding window and global attention (see Beltagy et al. (2020) [82] for more details). When compared to Equation 3.12, the longformer uses two sets of projections, one to compute attention scores for sliding window and another for global attention, providing the needed flexibility for the best performance on downstream tasks [82]. When compared to Transformer-XL [73], which can also handle long documents, the longformer is not restricted to the left-to-right approach of processing the documents.

TransformerXL [73] is an architecture that enables the representation of language and learn dependency beyond a predefined context length, without disrupting temporal coherence. In general, transformers are implemented with fixed-length context. Hence, long documents are truncated to the fixed-length segments and trained. The recurrence mechanism is introduced in TransformerXL, where during training, a hidden state sequence for the previous segment is reused [73]. This supports long-term dependency, and employs relative positional encoding scheme to avoid temporal confusion.

3.3 Experimental Methodology

This thesis makes use of several open-source frameworks. Neural network models presented in this thesis are mostly implemented using PyTorch², with some in Keras/Tensorflow³. All evaluations were done using sklearn metrics⁴. Transformer implementations are based on the open-source PyTorch-transformer repository.⁵

Transformer models were fine-tuned on all layers without freezing. FastText is used for training word embeddings and representing documents. Adam [71], an adaptive learning rate optimisation algorithm, is used as the optimiser for all neural networks presented in this thesis. A non-linear sigmoid function $f(z) = \frac{1}{1+e^{-z}}$, with a range of 0 to 1 as shown in Figure 3.9, is used as the activation function. Binary-cross-entropy [85] loss, Equation 3.13, over each label is used for multi-label classification.

$$Loss_{BCE}(X, y) = - \sum_{l=1}^L (y_l \log(\hat{y}_l) + (1 - y_l) \log(1 - \hat{y}_l)) \quad (3.13)$$

For binary classification, the Waikato Environment for Knowledge Analysis (WEKA) [86, 87] framework is used to train traditional classifiers on these documents. For multi-label classification MEKA [88], an open-source Java

²<https://github.com/pytorch/pytorch>

³<https://www.tensorflow.org/>

⁴<https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics>

⁵<https://github.com/huggingface/transformers>

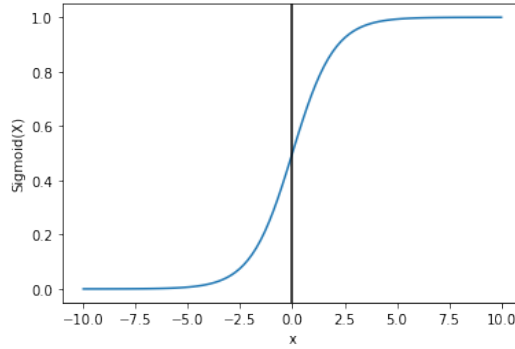


Figure 3.9: Sigmoid Function

system specifically designed to support multi-label classification experiments, is used for traditional classifiers.

All experiments using traditional classifiers are validated through 10-fold cross validation. Due to resource restrictions, all neural network results use train test hold-out set validation where the results are averaged over three runs. It is important to point out that averaging results over three runs is not enough and 10-fold cross validation is definitely more reliable. However, the variations of these three independent runs are usually within a range of ± 0.015 .

3.4 Evaluation Measures

Compared to single-label classification, multi-label classification requires different evaluation measures [46, 45, 89]. Example-based evaluation methods are calculated based on the average differences of the actual and predicted sets of labels over all test examples. Label-based evaluation methods calculate separate evaluations for each label and then averages over all labels. This thesis uses label-based classification metrics micro-F1 and macro-F1, and example-based classification metrics Hamming loss, and example-based ranking metrics ranking loss for evaluations. Label-based jaccard similarity scores are also calculated. For binary classification problems considering the imbalanced nature of the data, F1 scores are used as evaluation metrics. This section presents an

overview of such evaluation measures.

For multi-label problems, the notations as per Tsoumakas et al. (2009) [45] are used:

- $L = \{\lambda_j : j = 1 \dots q\}$ refers to the finite set of labels
- $D = \{(x_i, Y_i), i = 1 \dots m\}$ refers to sets of multi-label training examples

where x_i is the feature vector and $Y_i \subseteq L$ is the set of labels of the i -th example.

3.4.1 F-measure

For binary classification or individual labels of multi-label classification, F-measure or F1-score is used as the primary evaluation measure. F1-score is calculated by,

$$\begin{aligned} P &= \frac{\text{true positive rate}}{\text{true positive rate} + \text{false positive rate}} \\ R &= \frac{\text{true positive rate}}{\text{true positive rate} + \text{false negative rate}} \\ F1 &= 2 \times \frac{P \times R}{P + R} \end{aligned} \tag{3.14}$$

where P is precision and R is recall.

For multi-label classification problems, label-wise macro and micro F1 scores are used. Macro-F1 is the mean of the label-wise F1 score. Micro-F1 is the harmonic mean between the micro-recall and micro-precision.

3.4.2 Hamming Loss

Hamming loss [90] is defined as:

$$\text{Hamming-Loss} = \frac{1}{m} \sum_{i=1}^m \frac{|Y_i \Delta Z_i|}{|L|}, \tag{3.15}$$

where Z_i is the set of labels predicted and Δ stands for the symmetric difference of two sets, which is equivalent to the XOR operation in Boolean logic [45].

3.4.3 Ranking Loss

Ranking Loss compares the relevant labels to irrelevant labels and calculates the number of times that irrelevant labels are ranked lower than relevant labels [45], given by

$$\text{R-Loss} = \frac{1}{m} \sum_{i=1}^m \frac{1}{|Y_i| |\overline{Y_i}|} |\{(\lambda_a, \lambda_b) : r_i(\lambda_a) > r_i(\lambda_b), (\lambda_a, \lambda_b) \in Y_i \times \overline{Y_i}\}|, \quad (3.16)$$

where $\overline{Y_i}$ is the complementary set of Y_i with respect to L .

3.4.4 Jaccard Similarity

Jaccard similarity index compares two sets of labels and compares the labels that are shared and are distinct. It is calculated as follows:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \quad (3.17)$$

3.4.5 Statistical assessment of differences

Non-parametric tests are used to verify statistically significant differences between algorithms, as described in [91, 92]. First, Davenport's corrected Friedman test is used with $\alpha = 0.05$ to check if the null hypothesis that all algorithms perform the same can be rejected. If there are differences, then the post-hoc Nemenyi test determines the critical difference (CD) to identify algorithms with different performance. The critical difference plots are presented in the results chapters.

3.5 Related Work

This section presents a review of the literature related to the research presented in this thesis.

3.5.1 Multi-label Learning

Traditionally, multi-label classification methods are grouped into two main categories, “problem transformation” and “algorithm adaptation” methods. The problem transformation approach is when a multi-label problem is converted into multiple single-label classification problems [61, 62, 89]. Algorithm adaptation is when a specific algorithm for single-label classification problem is adapted to directly predict multi-labels [46, 93, 89, 94]. Development of multi-label classifiers for dealing with class imbalance incorporated ideas such as under/over-sampling, synthetic training sample generation and cost-sensitive learning [95].

For large scale or “extreme” multi-label classification, where labels ranged from a few hundred to a million labels, methods are categorised into embedding and tree-based methods. This thesis considers multi-label problems with label sizes in tens and hundreds, and hence the problems are not extreme. However, if all available MIMIC-III labels were considered, this would be a possible avenue to explore.

Hierarchical multi-label classification is a special case of multi-label classification that categorises a hierarchical relationship to form a structure like a Tree or Directed Acyclic Graph [96, 97, 98]. For a tree hierarchy, such as that of ICD-9 hierarchy, each class only has one parent class. This PhD research restricts itself to flat multi-label problems. However, the hierarchical approach is a promising avenue to consider in future.

There are many examples of advancements of deep-learning used in multi-label text classification problems. See Sections 3.5.2 and 3.2.2 for examples and details of such advancements.

3.5.2 Natural Language Processing related to Health care

This section presents a brief review of domain-specific NLP approaches used in health care and some examples of existing systems. Medical text processing and mining can incorporate rule-based algorithms, statistical and machine learning-based algorithms and hybrid models (where rule-based and machine learning-based are combined). Early approaches of clinical text mining using NLP are primarily rule-based. Recently, there is a shift in favouring machine learning or, more specifically, deep learning-based models. Hybrid models are popular where the availability of labelled data is limited, and the use of expert knowledge is a possibility [99, 100].

Rule-based techniques use dictionaries and hand-coded rules and rely on patterns expressed as regular expressions that are defined and tuned by humans [10, 101, 102, 100]. Rule-based systems also require an expert to provide knowledge of these rules. They do not need significant training data and are easily modifiable. However, rule-based methods have limitations, as they do require additional data curation or annotation by domain experts [102, 103]. Also, with rule-based systems, the rules in one system are not easily adopted by another system [104].

Machine learning approaches, alternatively, can be trained to recognise patterns and predict and extract information automatically. However, the main downside to machine learning techniques is that they require significantly large labelled datasets, as well as feature engineering [10, 103]. Machine learning systems perform worse than rule-based systems on rarely occurring labels due to a dearth of training data.

Over the years, rule-based and machine learning-based tools have been developed to aid information extraction from free-form medical text. Some examples of such early developments of information extraction (IE) systems using medical text data include MedLEE (Medical Language Extraction and

Encoding System) [105], SPRUS (Special Purpose Radiology Understanding System) [106], SymText (Symbolic Text processor) [107], Medtex [108], SPECIALIST system and MetaMap [109]. Apache cTAKES (clinical Text Analysis and Knowledge Extraction System), arguably the most popular open-source NLP based information extraction tool, uses techniques from both rule-based systems and machine learning for EHR in free-form text [18, 19, 20, 21, 22]. Limitations to the systems mentioned above include: increase in processing time based on the complexity of the text; reduced accuracy in the presence of ambiguity [18, 110]; the need to maintain a lexically variant-rich dictionary; and failing to recognise complex levels of synonymy [111].

The last decade has seen several examples of deep learning techniques applied to clinical data. For example, a study of early detection of heart failure on longitudinal EHR data used an RNN with word embeddings to train vector representations of diagnosis codes, medications codes and procedure codes [26]. Che *et al.* (2015) [112] use deep learning algorithms to predict physiologic patterns associated with known clinical phenotypes. There are several other examples of systems using RNNs with embeddings layers or conditional random field (CRF) on prediction based research, and information extraction of unstructured text EHR data [23, 24, 25, 25]. Examples of using deep learning models to achieve state-of-the-art (SOTA) performance in health data analysis include ICU mortality prediction [27], phenotype discovery [113] and disease prediction [114]. Purushotham *et al.*, (2017) [115] use MIMIC-III to present benchmark models on clinical prediction tasks such as mortality prediction, forecasting length of stay, and ICD-9 code group prediction. Since 2019, there exists examples of systems where BERT-base models are used in clinical applications [116, 117].

Many NLP tasks, health-related or otherwise, use word embeddings to represent text data, due to their ability to encode semantic similarity between words. Word embeddings represent a single word or sub-word as vectors. Examples of the use of word embeddings for health applications include: learning

medical concepts such as diagnosis codes, medication codes, procedure codes [118], early detection of heart failure [26], and medical event detection [23]. Many previous techniques have used the word2vec [53, 55] or GloVe [119] packages for training embeddings. One issue with the methods employed by word2vec and GloVe is that they cannot produce embeddings for words that were not seen during training. In contrast, fastText [50, 51, 48] makes use of character-level n-grams, which enables one to generate embeddings for words that do not appear in the training vocabulary. The use of character-level n-grams is of particular importance in the medical domain, where a significant number of compound words are used [37, 120, 121].

Applications in healthcare use standard Wikipedia or common-crawl text to train word embeddings. In cases where the health-related text is used to train word embeddings, most published models only use between 200 and 400 dimensions [122, 123]. Studies show that the use of large corpora from more than one source can improve the performance of embeddings [124, 125]. Chen *et al.*, (2019) [120] and Zhang *et al.*, (2019) [121] provide embeddings on health-related texts, with word embeddings of 700- and 200- dimensional embeddings respectively. Zhang *et al.*, (2019) [121] make use of the sub-word information during the training of word embeddings. This thesis also makes use of sub-word information during the training of word embeddings. Embeddings presented in this thesis make use of large corpora of health-related text from multiple sources (see Chapter 2 and Chapter 4 for details).

Word embeddings pre-trained on health-related text have shown to perform better on health-related tasks than those trained on general text. Examples of such publicly available embeddings trained on health-related text include BioSentVec [120] pre-trained on PubMed and MIMIC-III data using fastText, and BioWordVec [121] pre-trained on PubMed and MeSH data using sent2vec.

Before 2018, RNNs with word embeddings were the main systems used for NLP tasks. However, in the last two to three years, there have been considerable advancements in transformer models, which have shown substantial

improvements in many NLP tasks, including BioNLP tasks. With minimum effort, transfer learning of pre-trained models by fine-tuning on downstream supervised tasks achieves very good results [16, 126]. For example, PubMedBERT [74] achieves SOTA performance on many biomedical natural language processing tasks such as named entity recognition, question answering and relation extraction and holds the top score on the Biomedical Language Understanding and Reasoning Benchmark (BLURB) [74]. Yang et al. (2020) [127] showed the RoBERTa-MIMIC model outperforming SOTA methods for clinical concept extraction tasks.

The main observation with the clinical tasks that use transformer models is that the input text sequence cannot exceed 512 tokens [128]. Gao et al. (2021) [128] presents evidence to show that BERT-based models underperform in clinical text classification tasks when long data such as MIMIC-III is used, when compared to the baseline embeddings based neural networks such as CNN. Si and Roberts (2021) [129] presents an alternative system to overcome the issue with long documents by taking a similar approach to HAN and implementing a hierarchical transformer network. Transformer-based encoders are used to learn from words to sentences, sentences to notes and notes to patients progressively (see Section 3.2.2 for details on HAN). The transformer based HAN system presents SOTA methods for in-hospital mortality prediction and phenotype predictions using MIMIC-III. However, it requires considerable computational resources [129].

As with word embeddings, there are also many publicly available domain-specific transformer models, such as ClinicalBERT [130], PubMedBERT [74], BioMed-RoBERTa-base [80] and MeDAL-electra [81]. For both word embeddings based networks and transformers, there is evidence to show domain-specific pre-trained models outperform general text pre-trained models [80, 74, 28]. This research uses word embeddings pre-trained on health-related text and transformers pre-trained on general and health-related data [28, 29, 30].

3.5.3 Predicting Medical Codes

Automatically predicting medical codes from EHRs has been studied over the years, where rule-based, machine learning-based and deep learning approaches have been proposed. Since 1990s, there are many examples of attempts to automate extraction of medical codes such as ICD from clinical notes. Examples include a Bayesian network approach by Gundersen et al. (1996) [131] and a hierarchical approach by Lima et al. (1998) [132]. Perotte et al (2013) [133] developed a hierarchical support vector machine approach to predict medical codes. Techniques including CNNs, RNNs and Hierarchical Attention Networks are some examples of deep learning approaches [70, 126]. Moons et al. (2020) [70] presents a survey of deep learning methods for ICD coding of medical documents and indicates CAML [2] as the SOTA method for automatically predicting medical codes from EHRs.

Table 3.4 presents examples of embeddings based neural networks in predicting medical codes. The number of ICD-9 codes, i.e., the number of labels used, varies across systems. The number of labels and the frequency of the chosen labels influence the F1 score, with the top 50 ICD-9 codes generally leading to a higher F-measure. MIMIC-III is the biggest publicly accessible de-identified dataset. It is the most popular free-form medical text used in many applications, including predicting medical codes [115, 27, 32, 33].

Embeddings are the popular method used to represent text in a neural network, and all systems presented in Table 3.4 use embeddings from algorithms such as word2vec, Doc2Vec and ELMo to represent free-form medical text. Huggard et al. (2019) [138] also show that embeddings obtained from fastText result in significantly higher F-measures on biomedical name entity recognition when compared to other embeddings such as that of ELMo. This research uses fastText to obtain embeddings.

Examples of transformer models in automatically predicting medical codes includes submissions to CLEF eHealth 2019 ICD-10 predictions from German documents [126, 139], and BERT and XLNet performance on most frequent

System	Architecture	Label details
Zeng et al. (2019) [134]	Deep transfer learning, Multi-scale CNN, 100-dimensional general text pre-trained embeddings	most frequent 200 labels
Du et al. (2019) [135]	ML-Net, ELMo based, LSTM	most frequent 70 labels
Baumel et al. (2018) [136]	HAN (Bi-GRU), 100-dimensional general text pre-trained embeddings	1047 labels and 6527 labels
Mullenbach et al. (2018) [2]	CAML, DR-CAML, 100-dimensional Word2Vec embeddings pre-trained on MIMIC-III	most frequent 50 labels and 8922 labels
Li et al. (2018) [132]	DeepLabeler, CNN, 128 dimensional Doc2Vec and 100 dimensional Skipgram word embeddings general text pre-trained embeddings	6984 labels
Rios and Kavuluru (2018) [137]	CNN, few-shot learning 300 dimensional Skip-gram embeddings pre-trained on PubMed data.	6932 labels

Table 3.4: Examples of embeddings based neural networks for predicting ICD-9 codes are presented.

ICD-9 codes from MIMIC-III with a maximum number of tokens set at 512 [15]. Biswas et al. (2021) [17] present a Transformer based architecture to capture the interdependence among the tokens and makes use of a code-wise attention mechanism to learn code-specific representations in MIMIC-III documents. They show that the proposed new architecture outperforms CAML and other networks when the most frequent 50 ICD-9 codes are considered a multi-label problem. However, it is unclear how well transformer models can perform with long clinical documents and in multi-label problems with a large number of labels [15, 128, 17]. Also, many studies [15, 16, 17] focus on high-frequency labels. Nonetheless, datasets such as MIMIC-III and eICU consist of many infrequent labels where most codes only occur in a minimal number of clinical documents. This research considers several variations of labels and frequency, presents multiple transformer methods and compares them with SOTA meth-

ods for various sequence lengths and labels.

Chapter 4

Domain-specific Embeddings and Binary Classification

Extracting knowledge from electronic health records (EHRs) using NLP techniques such as embeddings has proven to be very popular and useful [118, 26, 23]. However, in many cases, embeddings are trained on data that do not accurately reflect how language is used in a healthcare context. Embeddings trained on the general text, used for tasks that involve specialised language, results in a domain shift, which will typically cause suboptimal performance [140]. Ideally, to classify documents derived from EHRs, the embeddings should be trained on an extensive collection of free-form text extracted from EHRs. This is not possible for various legal and ethical reasons: the collections of health records available for research purposes are not large enough to train high-quality word embeddings.

This chapter presents an investigation on how domain-specific source data used for pre-training embeddings and the dimensionality of embeddings impacts the performance of the medical text classification. The prediction of medical codes is used as the example application, where for simplicity and understanding of NLP techniques, medical code predictions are treated as a binary classification problem. The top-level 18 ICD-9 groups, as shown in Figure 2.3, are considered 18 separate binary classification problems.

In addition to embeddings, this chapter also considers BOW and transformers to represent EHRs from MIMIC-III and eICU. Experiments show that in most ICD-9 groups, F1 scores of transformers and embeddings are better than that of BOW. Categories **symp** and **preg** are the two exceptions where it is shown that the bag of words representation are better than word embeddings and transformers.

Embeddings and BOW parts of this chapter was published as a conference paper [28], where MIMIC-III data and traditional binary classifiers were used. For eICU data, this chapter only presents selected ICD-9 groups with selected classifiers. However, for the multi-label problems this thesis presents extensive experimental evaluations for both MIMIC-III and eICU.

4.1 Bag-of-Words (BOW)

This section presents F1 scores for both MIMIC-III and eICU for BOW. Results for all 18 categories for MIMIC-III and selected groups for eICU are presented. Figure 4.1 presents a flowchart of using BOW for predicting ICD-9 groups from MIMIC-III or eICU data. WEKA's implementation of BOW is used with dictionary sizes of 1,000, 10,000 and 100,000 words. ICD-9 level 1 groups are considered as an individual binary classification problem. WEKA's implementation of Multinomial NB and LR is used for classification, and scikit-learn implementations of evaluation measures were used for evaluations. For validation, ten-fold cross-validation was used.

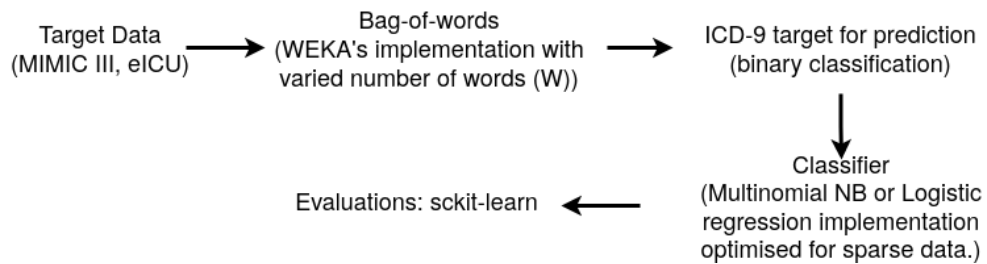


Figure 4.1: Flow chart of using bag-of-words for predictions.

ICD-9	BOW LR			BOW multinomial NB		
	1,000	10,000	100,000	1,000	10,000	100,000
circ (78.4%)	0.930	0.920	<u>0.931</u>	0.913	0.922	0.927
e+v (69.1%)	0.801	0.788	<u>0.808</u>	0.806	0.783	0.806
endo (66.5%)	0.814	0.825	<u>0.840</u>	0.794	0.811	0.817
resp (46.6%)	0.763	0.775	<u>0.788</u>	0.712	0.733	0.734
inj (41.4%)	0.642	0.675	<u>0.693</u>	0.640	0.672	0.681
gen (40.3%)	0.733	0.751	<u>0.769</u>	0.668	0.688	0.691
diges (38.8%)	0.701	0.728	<u>0.748</u>	0.640	0.666	0.673
bld (33.6%)	0.558	0.595	0.608	0.587	0.602	<u>0.611</u>
symp (31.4%)	0.476	0.507	0.524	0.563	0.570	<u>0.576</u>
ment (29.7%)	0.567	0.616	<u>0.635</u>	0.531	0.549	0.569
nerv (29.1%)	0.491	0.594	<u>0.628</u>	0.538	0.567	0.589
inf (27.0%)	0.606	0.667	<u>0.693</u>	0.604	0.613	0.621
musc (18.0%)	0.344	0.476	<u>0.488</u>	0.381	0.398	0.456
pren (17.1%)	0.574	0.557	<u>0.620</u>	0.571	0.584	0.589
neop (16.4%)	0.665	0.713	<u>0.766</u>	0.506	0.525	0.628
skin (12.0%)	0.438	0.483	<u>0.526</u>	0.357	0.362	0.421
cong (5.4%)	0.348	0.485	<u>0.519</u>	0.365	0.361	0.429
preg (0.3%)	0.737	0.705	0.709	0.099	0.516	<u>0.756</u>

Table 4.1: F1 scores for ICD-9 groups for MIMIC-III data for BOW with different dictionary sizes using logistic regression (left) and multinomial naive Bayes (right) are presented. Bold is used to indicate the best F1 scores among a varied number of words for BOW with the same classifier. The best F1 score across all is underlined for each category. Frequency of occurrence of ICD-9 groups in unique hospital admission with MIMIC-III text data is indicated in brackets, and labels are ordered from most frequent at the top to least frequent at the bottom.

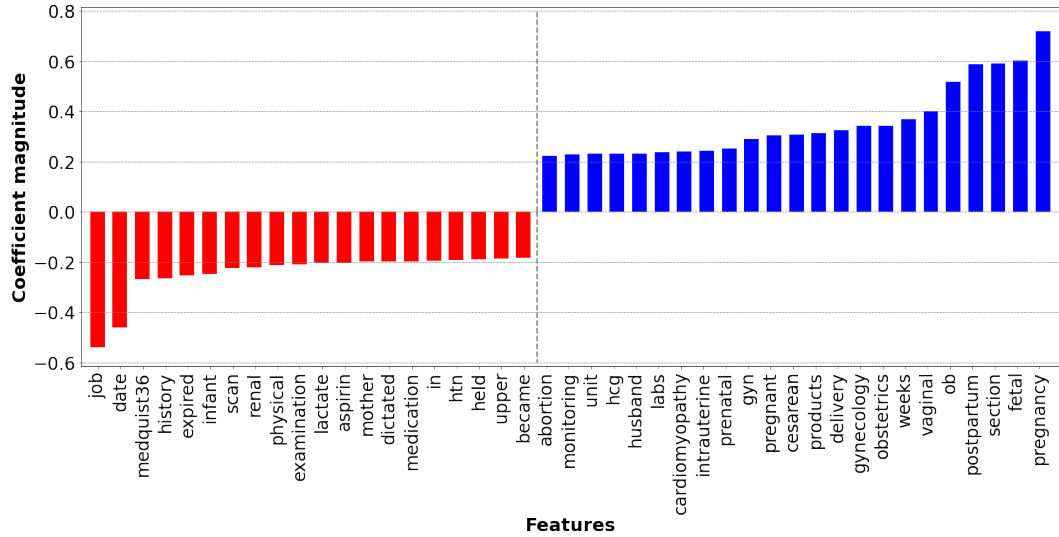


Figure 4.2: Coefficient of features LR with BOW for category **preg** for MIMIC-III. Blue is used for words which have the positive influence on making prediction for category **preg** and red for words which have negative influence.

4.1.1 MIMIC-III

Table 4.1 provides a comparison of F1 score for predicting ICD-9 groups from free-form MIMIC-III discharge summaries using BOW with different dictionary sizes. A total of 52,710 discharge summaries were used, with text length ranging from a few sentences to close to twenty pages. Classifiers used are logistic regression with ridge value of 1 and multinomial Naive Bayes. LR used for these experiments is modified to optimise for sparse data. F1 scores for dictionary sizes 1,000, 10,000 and 100,000 are presented. In general, there is an increase in F1 score as the size of the dictionary increases. A dictionary size of 100,000 results in the best F1 scores, for all ICD-9 categories, except **preg**, when LR is used. One explanation for such behaviour is that in MIMIC-III, **preg** occurs in very few instances, and the vocabulary related to **preg** is also specific and limited. In comparison to multinomial Naive Bayes, the F1 scores obtained using LR are better for most categories. Figure 4.2 presents the coefficients of features where LR is used for category **preg**. Evidently, the words related to **preg** are with the largest coefficients and have the most influence in prediction.

ICD-9	BOW multinomial NB		
	1,000	10,000	100,000
circ (53.52%)	0.796	0.798	0.798
symp (23.9%)	0.522	0.532	0.532
endo (23.0%)	0.641	0.648	0.648
bld (9.8%)	0.423	0.431	0.431
e+v (3.1%)	0.341	0.356	0.356

Table 4.2: F1 scores for selected ICD-9 groups for eICU data for BOW with different dictionary sizes using multinomial naive Bayes. Bold to indicate the best F1 scores among a varied number of words for BOW. Frequency of occurrence of ICD-9 groups in unique patient stays with eICU text data is indicated in brackets, and labels are ordered from most frequent at the top to least frequent at the bottom.

4.1.2 eICU

Table 4.2 provides a comparison of F1 score for predicting selected ICD-9 groups from eICU using BOW for a varied number of words using multinomial Naive Bayes. F1 scores of dictionary sizes 1,000, 10,000 and 100,000 are presented. F1 scores of dictionary sizes of 10,000 and 100,000 are the same for all categories presented. The vocabulary in eICU is a lot smaller than MIMIC-III with the total number of features being 2,700. Hence, a dictionary size of 10,000 captures all of the features used. As observed with MIMIC-III data, BOW using LR only marginally improved the results obtained using BOW with multinomial NB, therefore LR was omitted from Table 4.2.

4.2 Word Embeddings

This section presents details on training and visualising embeddings. Embeddings are used to classify medical codes from both MIMIC-III and eICU data.

4.2.1 Training Word Embeddings

Word embeddings of various dimensions using health-related data are pre-trained with fastText [50, 48, 51]. For direct comparison, these embeddings are primarily trained to the exact specifications as the Wikipedia and common crawl fastText models in [141]. However, in future chapters, other parameter choices and specifications are explored for training embeddings and are used for analysis for multi-label classifications.

Both MIMIC-III and TREC 2017 datasets are used to train word embeddings. The sizes of these datasets are 4GB and 24GB, respectively. The word embeddings are trained to the exact specifications of Wikipedia and common crawl pre-trained fastText models –W300 [141]– using the CBOW method, character n-grams of length 5, a window of size 5, ten negative samples per positive sample, and various settings for the number of dimensions. The learning rate used for training these models is 0.05. Also, two recently published medical text trained word embeddings of dimension size 200 and 700 [120, 121] are included for comparison.

Table 4.3 presents the embedding trained using fastText, previously published word embeddings, and the concatenated word embeddings. Concatenated embeddings are word embeddings formed by concatenating multiple word embeddings. For example, in the T300+M300, the first 300 elements are the word vectors obtained using the TREC dataset, and the second 300 elements are taken from the embeddings trained on MIMIC-III. The table includes details on dimensions, input data, training time¹ and the size of the model. Both the size of the input data and the number of dimensions influence the training times and model sizes. For the models trained for this thesis, number of dimensions of a model with same input data is linearly proportional to the training time. Model size also increases as the input data and the number of dimensions increases.

¹Training was run on a 4 core Intel i7-6700K CPU @ 4.00GHz with 64GB of RAM.

²<https://www.ncbi.nlm.nih.gov/pubmed/>

Models	Dimensions	Source Data	Train Time	Model Size
M300	300	MIMIC	1 hour	5GB
T300	300	TREC	7 hours	13GB
TM300	300	TREC+MIMIC	9 hours	15GB
T600	600	TREC	13 hours	23GB
TM600	600	TREC+MIMIC	16 hours	30GB
T900	900	TREC	19 hours	35GB
TM900	900	TREC+MIMIC	23 hours	54GB
W300 [141]	300	Wiki	-	7GB
BWV200 [120, 121]	200	PubMed ² +MIMIC	-	26GB
BSV700 [120, 121]	700	PubMed+MIMIC	-	21GB
T300+M300	600	TREC+MIMIC	8 hours	18GB
W300+T300+M300	900	Wiki+TREC+MIMIC	8+ hours	25GB
T900+W300	1,200	Wiki+TREC	19+ hours	42GB
TM900+W300	1,200	Wiki+TREC+MIMIC	23+ hours	61GB

Table 4.3: Word embeddings trained for this thesis (top), from previous work (middle), or concatenations thereof (bottom). Dimension details are presented, as are training times (rounded to hours) and word embeddings model sizes.

4.2.2 Visualising Embeddings

To better understand the embeddings pre-trained using health-related text and general text, Table 4.4 presents most similar words from an example input sentence ‘**patient** brought into **ED complaining of chest pain** is **admit** in **cardio...**’. The top 5 most similar words were obtained using gensim’s³ built in function `most_similar`⁴ which computes cosine similarity between the projection weight vectors of a word and the vectors of each word in the embeddings model. The two models used for comparison are T300, which is a health-related pre-trained embeddings, and W300, which is a general text pre-trained embeddings, as presented in Table 4.3.

³https://tedboy.github.io/nlps/api_gensim.html

⁴https://tedboy.github.io/nlps/generated/generated/gensim.models.Word2Vec.most_similar.html

Input word	Similar words for T300
patient	patientPatient, patientPrior, patientwhich, patientWHO, oldpatient
ED	EDs, nonemergencydepartment, emergencydepartment, EDAll, EDThe
complaining	uncomplaining, complained, complaint, complains, complain
chest	BChest, chestupper, CTChest, PChest, chestX
pain	painwe, painXROA, painPain, pain. . . , painrelieved
admit	admitting, hospitalize, admits, deny, admissible
cardio	cardiovascular, cardiocerebral, heartcardiovascular, cardiovascular, cardiocerebro
Input word	Similar words for W300
patient	patients, Patient, clinician, patient.The, physician
ED	non-ED, ED., EDs, SHEERAN, AE
complaining	griping, complain, complaining, whining, bitching
chest	chest., chest.The, chest.It, chest-, chest.This
pain	pain.This, discomfort, pain-, pain.What, pain.But
admit	confess, admitt, admit.I, admit-, concede
cardio	workout, cardio-based, Cardio, HIIT, cardio.

Table 4.4: Five most similar words obtained using word embeddings trained by fastText using health domain-specific text T300 (top) and general text W300 (bottom). For input tokens **patient**, **ED**, **complaining**, **chest**, **pain**, **admit**, **cardio**. See Figures 4.3 and 4.4 for a visual representation of these words.

Figures 4.3 and 4.4 present visual representations of the most frequent 10 words of the above mentioned input tokens. A three dimensional scatter plot is presented where scikit-learn’s principal component analysis (PCA)⁵ is used to reduce the dimensionality of data from the embeddings and project it to a lower-dimensional space.

⁵<https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html>

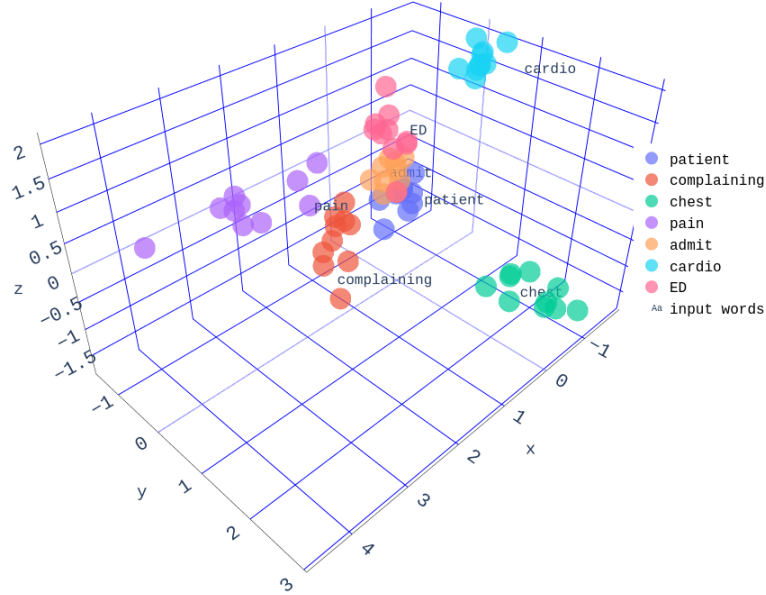


Figure 4.3: Visualising word embeddings trained by fastText using health domain-specific text T300. For input text of **patient**, **ED**, **complaining**, **chest**, **pain**, **admit**, **cardio**, the 10 most similar words are presented.

Figure 4.3 shows clusters of most similar words for input tokens, where the differences within each cluster are visible. Figure 4.4 presents visual representations of the most frequent words for the input tokens, using general text pre-trained embeddings W300, and is presented as two separate graphs.

In contrast to Figure 4.3, the top figure of 4.4 shows one cluster for most of the input tokens with **ED** being an out-liar. Domain-specific embeddings can differentiate the words in input text where the clusters are visibly separate while also mapping the relationship between the words. Both Table 4.4 and Figures 4.3 and 4.4 show a clear difference in the clusters of similar words and, as explained, show the advantage of using domain-specific embeddings for health-related input text.

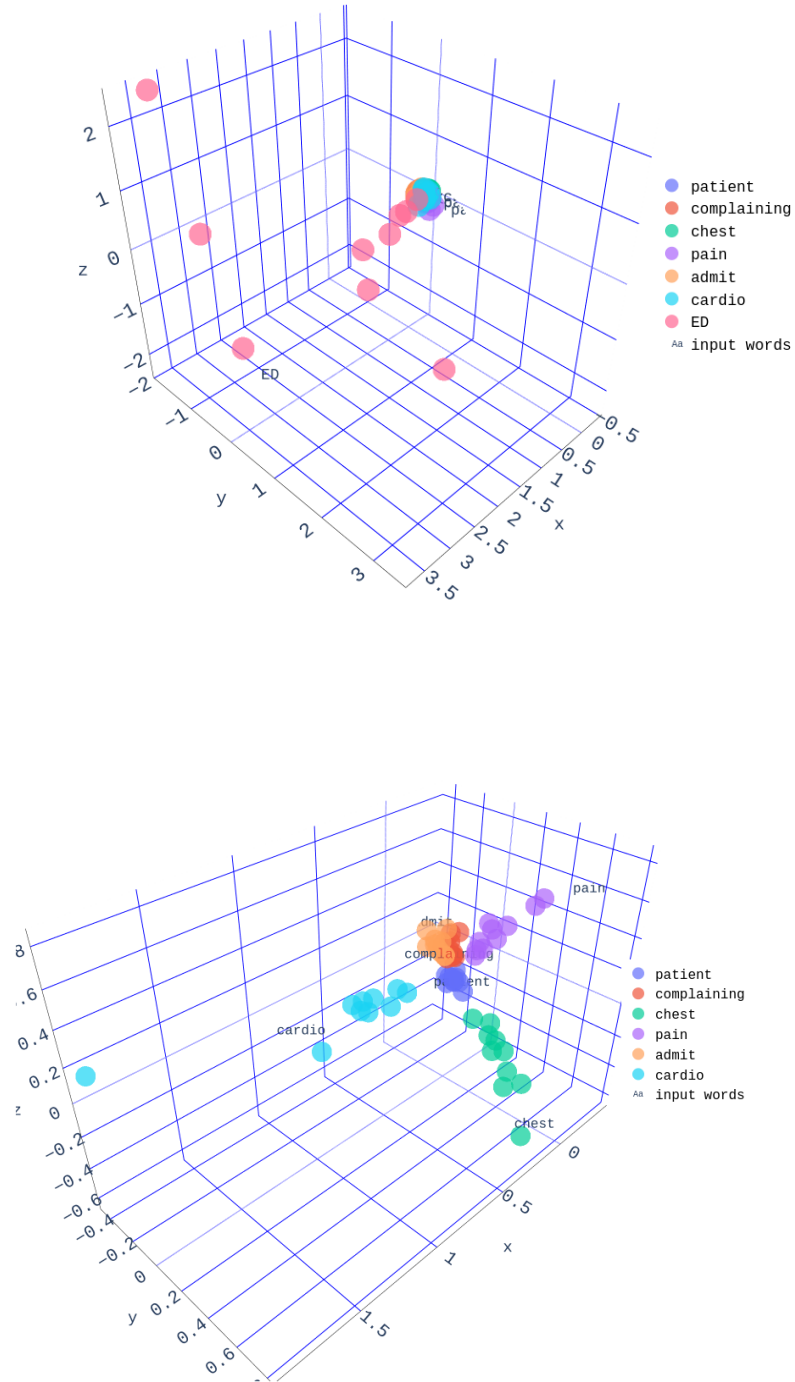


Figure 4.4: Visualising word embeddings trained by fastText using general text W300. The top figure includes token ED while the bottom figure is focused on all other tokens, excluding ED. On the top figure token ED dominates the rest of the tokens, hence, the inclusion of the bottom figure. For input text of ‘patient, ED, complaining, chest, pain, admit, cardio’, the ten most similar words are presented.

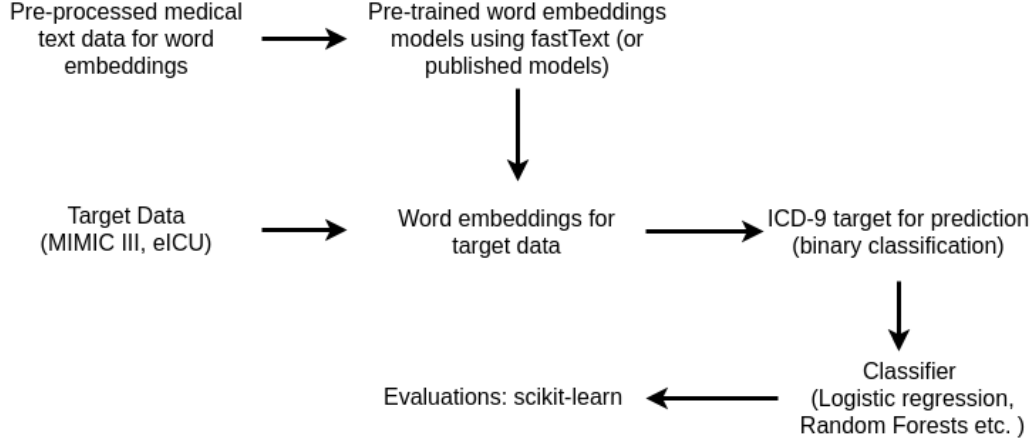


Figure 4.5: Flow chart of using word embeddings for predictions.

4.2.3 Binary Classification using Embeddings

This section presents F1 scores for MIMIC-III and eICU, where text is represented using word embeddings. Results for all 18 categories for MIMIC-III and selected groups for eICU are presented. Figure 4.5 presents a flowchart for the use of embeddings for predicting ICD-9 groups from MIMIC-III or eICU data. FastText pre-trained embeddings presented in Table 4.3 are used. WEKA’s implementation of LR and random forests is used for classification, and scikit-learn implementations of evaluation measures are used for evaluations. For validation, ten-fold cross-validation is used.

Tensorflow implementation of GRU incorporating the details of the network structure is also used for experiments where input text of sequence length 3,000 is used with padding for text with the number of tokens less than 3,000. All experimental results using neural networks are obtained from validations based on training-testing scheme, and are averaged over three runs.

4.2.3.1 MIMIC-III

Logistic regression with a ridge value of 1 and GRU is used for word embeddings experiments presented in this chapter. The use of random forests with various parameter choices were explored. However, random forest did not perform as well as LR or GRU. Other ridge values for logistic regression were also

ICD-9	BWV200	W300	M300	T300	TM300	T600	TM600	T3+M3	BSV700
circ	0.931	0.932	0.932	0.932	0.931	<u>0.935</u>	0.934	0.924	0.931
e+v	0.829	0.828	0.829	0.829	0.828	<u>0.832</u>	<u>0.832</u>	<u>0.832</u>	0.831
endo	0.847	0.845	0.846	0.849	0.846	<u>0.851</u>	0.849	0.850	0.847
resp	0.774	0.774	0.774	0.778	0.772	<u>0.789</u>	0.788	0.787	0.776
inj	0.660	0.649	0.663	0.662	0.660	0.675	0.676	0.677	<u>0.682</u>
gen	0.721	0.716	0.724	0.731	0.725	<u>0.740</u>	<u>0.740</u>	<u>0.740</u>	0.732
diges	0.679	0.692	0.693	0.696	0.692	<u>0.712</u>	0.705	0.710	0.696
bld	0.557	0.566	0.573	0.570	0.570	0.593	0.589	<u>0.594</u>	0.586
symp	0.475	0.486	0.482	0.487	0.483	0.504	0.502	0.500	<u>0.505</u>
ment	0.533	0.530	0.530	0.542	0.539	<u>0.577</u>	0.576	<u>0.577</u>	0.559
nerv	0.530	0.534	0.527	0.543	0.531	<u>0.571</u>	0.558	0.564	0.553
inf	0.634	0.634	0.641	0.647	0.647	0.663	0.659	<u>0.664</u>	0.648
musc	0.254	0.274	0.258	0.294	0.267	<u>0.338</u>	0.314	0.319	0.315
pren	0.589	0.590	0.588	0.594	0.587	0.601	0.597	0.598	<u>0.603</u>
neop	0.693	0.688	0.702	0.705	0.690	0.728	0.721	<u>0.732</u>	0.727
skin	0.343	0.335	0.344	0.346	0.344	0.389	0.384	0.386	<u>0.397</u>
cong	0.365	0.371	0.369	0.391	0.350	<u>0.438</u>	0.406	0.435	0.424
preg	0.525	0.502	0.543	0.565	0.512	0.579	0.566	<u>0.599</u>	0.586

Table 4.5: A comparison of F1 scores for predicting ICD-9 groups using 200, 300, 600 and 700-dimensional word embeddings are presented. BWV200 and BSV700 are both published word embeddings and are compared with 300-dimensional word embeddings and 600-dimensional word embeddings, respectively. Bold is used to indicate the best F1 scores among low dimensional groups of word embeddings (200-300) and the higher dimensional word embeddings (600-700). The best F1 score across all presented word embeddings is underlined for each category.

explored. LR provided consistent F1 scores across a range of different ridge values. Appendix B presents a summary of random forest experiments and LR results with varied ridge value.

Table 4.5 provides a comparison of the F1 score for predicting ICD-9 groups from free-form MIMIC-III discharge summaries for 300-dimensional and 600-dimensional embeddings. For 300-dimensional embeddings, W300 are word embeddings trained by fastText on Wikipedia and other common crawl text.

ICD-9	LR					GRU	
	600	900	best 900- dim model	1,200	best 1,200- dim model	T300	T600
circ	0.935	0.937	T900	0.936	T9W3	0.938	<u>0.939</u>
e+v	0.832	<u>0.833</u>	W3T3M3	<u>0.833</u>	T9W3	0.803	0.804
endo	0.851	0.853	T900	<u>0.854</u>	T9W3	<u>0.854</u>	0.853
resp	0.789	0.792	W3T3M3	0.794	TM9W3	0.810	<u>0.819</u>
inj	0.677	0.684	WTM+T900	0.689	TM9W3	<u>0.708</u>	<u>0.708</u>
gen	0.740	0.748	TM9	0.751	TM9W3	<u>0.795</u>	0.776
diges	0.712	0.724	T900	0.730	T9W3	<u>0.781</u>	0.757
bld	0.594	0.601	TM9	0.607	T9W3	0.617	<u>0.636</u>
symp	0.504	0.514	W3T3M3	0.517	T9W3	<u>0.521</u>	0.516
ment	0.577	0.592	TM9	0.606	TM9W3	<u>0.671</u>	0.662
nerv	0.571	0.577	T900	0.586	T9W3	<u>0.664</u>	0.653
inf	0.664	0.671	T900	0.677	T9W3	<u>0.733</u>	0.729
musc	0.338	0.354	T900	0.372	T9W3	<u>0.539</u>	0.478
pren	0.601	0.607	W3T3M3	0.608	Both	<u>0.618</u>	0.610
neop	0.732	0.741	W3T3M3	0.746	T9W3	<u>0.790</u>	0.779
skin	0.389	0.418	T900	0.435	T9W3	<u>0.585</u>	0.567
cong	0.438	0.465	T900	0.463	T9W3	0.510	<u>0.539</u>
preg	0.599	0.593	W3T3M3	<u>0.605</u>	TM9W3	0.557	0.524

Table 4.6: F1 score for ICD-9 groups for word embeddings with varied dimensions using LR (left) and GRU (right) for MIMIC-III. For LR, the best F1 score across 900-dimensional and 1200-dimensional word embeddings for each category is presented. Corresponding best 900- and 1,200- dimensional models are also listed. For details of the models see Table 4.3. Bold is used to indicate the best F1 scores among varied dimensional word embeddings with the same classifier. The best F1 score across all is underlined for each category.

W300 embeddings are readily available for use in any application. Except for the circulatory label, which is the most frequent one (78.4%), word embeddings specially trained on medical corpora have better F1 scores. Overall, T300

provides better F1 scores than other 300-dimensional word embeddings for most ICD-9 groups. Compared to the recently published BWV200, the 300-dimensional word embeddings performed better for all categories and on par with **e+v**.

For 600-dimensional word embeddings, Table 4.5 presents comparisons across embeddings obtained in a single training phase (T600 and TM600) and word embeddings obtained via concatenation (T300 + M300). Comparing the 600 dimension word embeddings trained for this research to the published 700-dimensional word embeddings (BSV700), F1 scores of this 600-dimensional word embeddings are on par with or better than those recently published high dimensional word embeddings.

Table 4.6 presents a comparison for predicting ICD-9 code from free-form discharge summaries in MIMIC-III with various dimensions of word embeddings and between classifiers LR and GRU. The best F1 scores for 600-, 900- and 1200- dimensional embeddings are presented for each ICD-9 group. Models that produced the best 900-dimensional, and 1,200-dimensional embeddings are also indicated. Refer to Table 4.3 for details of word embeddings, input data and model dimensionality. The higher the dimensionality, the better the F1 scores are for predicting the ICD-9 groups using LR. Compared to LR, GRU performs better for most categories, except **e+v**, **endo** and **preg**. For GRU, 300-dimensional embeddings T300 is better than T600 for the majority of the categories.

4.2.3.2 eICU

Table 4.7 presents a comparison for predicting selected ICD-9 groups from eICU data with various dimensions of word embeddings and between classifiers LR and GRU. F1 scores of the 600-dimensional word embeddings trained for this research (T600) is better than that of the published 700-dimensional word embeddings (BSV700). For GRU, 300-dimensional embeddings T300 are better than T600 for the majority of the categories. Also, both LR and GRU's

ICD-9	LR						GRU	
	W300	T300	M300	TM300	T600	BSV700	T300	T600
circ	0.819	0.820	0.818	0.818	<u>0.825</u>	0.814	0.820	0.805
symp	0.436	0.442	0.442	0.438	<u>0.470</u>	0.445	0.386	0.410
endo	0.647	0.647	0.647	0.647	0.660	0.641	<u>0.663</u>	0.639
bld	0.375	0.382	0.376	0.369	<u>0.412</u>	0.374	0.345	0.270
e+v	0.585	0.592	0.583	0.581	0.595	0.569	<u>0.627</u>	0.618

Table 4.7: F1 score for ICD-9 groups for word embeddings with varied dimensions using LR (left) and GRU (right) for eICU. For details of the models see Table 4.3. Bold is used to indicate the best F1 scores among varied dimensional word embeddings with the same classifier. The best F1 score across all is underlined for each category.

performances are on par with each other.

4.3 Transformers

This section presents F1 scores for both MIMIC-III and eICU for various transformer models. Results for all 18 categories for MIMIC-III and selected groups for eICU are presented. All experimental results using neural networks are obtained from validations based on training-testing scheme, and are averaged over three runs. The maximum number of tokens are calculated from the start of the input sequence, i.e. 512 tokens indicate the first 512 tokens seen by the transformer model.

4.3.1 MIMIC-III

Table 4.8 presents a comparison for predicting ICD-9 code from free-form discharge summaries in MIMIC-III with various transformer models. BERT-base and Longformer are pre-trained using general text, and ClinicalBERT, PubMedBERT, and BioMed-Roberta are pre-trained using health-related data. Longformer is the only transformer among the ones presented that can handle

ICD-9	BERT	ClinicalBERT	PubMedBERT	BioMed-Roberta	Longformer	Longformer
tokens =	512	512	512	512	512	3,000
circ	0.932	0.937	0.938	0.935	0.933	<u>0.949</u>
e+v	0.832	<u>0.837</u>	0.836	0.836	<u>0.837</u>	0.836
endo	0.873	0.763	0.872	0.865	0.867	<u>0.893</u>
resp	0.771	0.771	0.784	0.778	0.780	<u>0.860</u>
inj	0.665	0.682	0.671	0.670	0.678	<u>0.722</u>
gen	0.700	0.716	0.724	0.734	0.705	<u>0.838</u>
diges	0.752	0.743	0.773	0.771	0.765	<u>0.825</u>
bld	0.562	0.572	0.575	0.558	0.560	<u>0.670</u>
symp	0.442	0.519	0.468	0.471	0.418	<u>0.555</u>
ment	0.664	0.650	0.704	0.685	0.690	<u>0.737</u>
nerv	0.652	0.626	0.678	0.648	0.592	<u>0.721</u>
inf	0.646	0.649	0.668	0.665	0.655	<u>0.774</u>
musc	0.470	0.512	0.578	0.501	0.479	<u>0.599</u>
pren	0.611	0.639	<u>0.663</u>	0.658	0.646	0.640
neop	0.778	0.778	0.810	0.784	0.781	<u>0.826</u>
skin	0.445	0.455	0.480	0.496	0.486	<u>0.623</u>
cong	0.545	0.555	0.610	0.592	0.530	<u>0.631</u>
preg	0.602	0.716	0.755	0.762	0.750	<u>0.795</u>

Table 4.8: F1 score for ICD-9 groups for transformer models, where the maximum token length of the input data, MIMIC-III, is specified. Bold is used to indicate the best F1 scores for each category for maximum token size of 512, and underline is used for overall best F1 score.

a more extended sequence of text. Among the transformer models that restrict the maximum token size of the input text to 512, health-related pre-trained models perform better than BERT-base. PubMedBERT is the best performing method for the 512 tokens setting. Overall, Longformer, with a longer input sequence length of up to 3,000 tokens, performs better than all other models, including Longformer with sequence length restricted to 512.

4.3.2 eICU

Table 4.9 presents a comparison for predicting selected ICD-9 groups from eICU data with various transformer models. Given that eICU text is shorter

ICD-9	ClinicalBERT	PubMedBERT	Longformer	Longformer
tokens =	512	512	512	1,250
circ	0.828	0.828	<u>0.834</u>	<u>0.834</u>
sym	0.458	<u>0.466</u>	0.447	0.460
endo	<u>0.685</u>	0.667	0.673	0.670
bld	0.350	0.376	0.414	<u>0.435</u>
e+v	0.631	0.641	0.631	<u>0.649</u>

Table 4.9: F1 score for ICD-9 groups for transformer models, where the maximum token length of the input data, eICU, is 512 for ClinicalBERT and PubMedBERT, and both 512 and 1,250 for Longformer. Bold is used to indicate the best F1 scores for each category for maximum token size of 512, and underline is used for overall best F1 score.

than MIMIC-III, Longformer’s maximum token length of the input sequence is set at 1,250. Overall F1 score obtained using Longformer is better than that of the other transformers presented.

4.4 Discussion

This chapter considered top-level ICD-9 groups as 18 individual binary classification problems. An investigation on the impact of the performance of medical text classification due to the source domain used for training word embeddings is presented. Also, the role that embedding dimensionality plays in determining the accuracy of the resulting classifiers is demonstrated. The predictions of ICD-9 codes are used as an example of health applications to show that high dimensions, specifically trained on health-related corpora, have better F1 scores than word embeddings with lower dimensions or when trained on general text like Wikipedia. A comparison with recently published word embeddings shows that embeddings pre-trained for this research perform better for most ICD-9 groups and are very similar for the rest. Reasons for such differences include pre-processing of input data, parameter selection, and source of the

datasets used for training the embeddings.

The sizes and training times required for training word embeddings are also presented. Model sizes and training times are influenced by the input data size and the number of dimensions generated and can become quite large. The main reason for the large model sizes is the use of hash tables for storing character n-gram information. FastText does provide ways to reduce the final word embeddings model sizes. However, such compression may possibly also impacts accuracy. Further analysis is needed to better understand the impact of compression on accuracy.

In addition to embeddings, BOW representations and the more recent transformer models were also used. Domain-specific transformer models perform better than the base models trained on general text for cases where maximum input sequence length is limited to 512. In general, Longformer had an advantage over the other models, primarily because Longformer can handle longer sequences.

Figures 4.6 and 4.7 present a summary of the best F1 score presented in this chapter for BOW, embeddings and transformers for MIMIC-III and eICU,

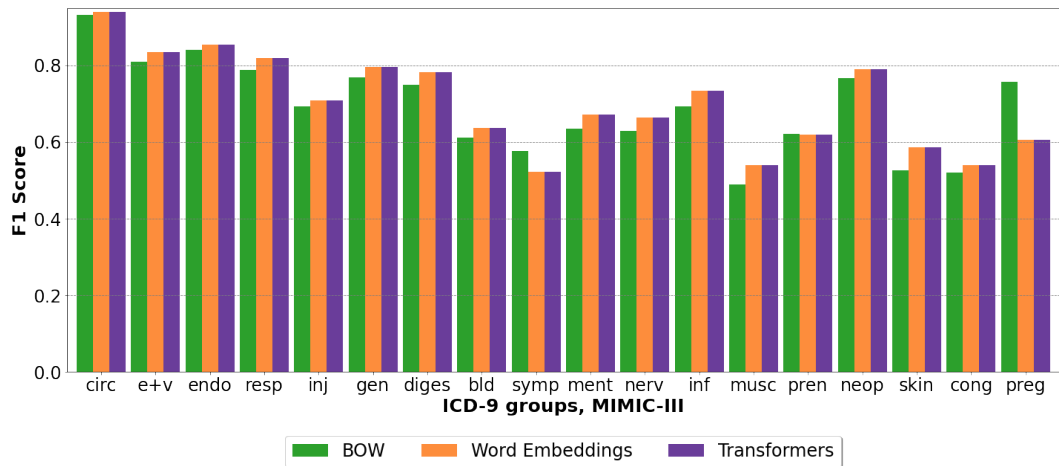


Figure 4.6: Best F1 scores, based on the results presented in this chapter, of BOW, embeddings, and transformers for all 18 ICD-9 groups treated as individual binary classification problem for MIMIC-III. The bar plots are ordered such that the most frequent label **circ** is on the left with the least frequent **preg** at the end.

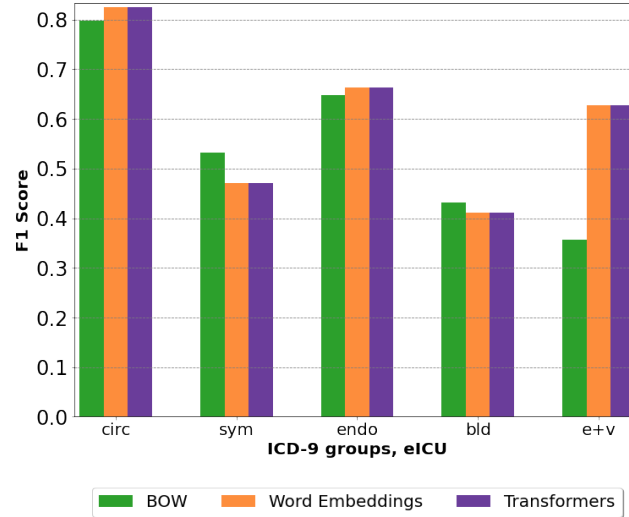


Figure 4.7: Best F1 scores, based on the results presented in this chapter, of BOW, embeddings, and transformers for selected ICD-9 groups treated as individual binary classification problem for eICU data. The bar plots are ordered such that the most frequent label **circ** is on the left with the less frequent **e+v** at the end.

respectively. For most categories in Figure 4.6, embeddings and transformers perform better than BOW, except for categories **symp** and **preg**. In terms of BOW performance, pregnancy is the most interesting ICD-9 group. Its frequency is only 0.31% in the MIMIC-III dataset. This observation can be better understood by taking a closer look at the coefficients, as shown in Figure 4.2, for each feature to determine its weight for LR. The model predicts words such as 'pregnancy', 'fetal', 'section', 'postpartum', 'ob', 'vaginal', 'weeks', 'obstetrics', 'gynecology', 'delivery', 'products', 'cesarean', 'pregnant', 'gyn', 'pre-natal', 'intrauterine', 'cardiomyopathy', 'labs', 'husband', 'hcg' with the largest coefficients.

Figure 4.7 presents the best F1 scores for BOW, embeddings and transformers for eICU data. In general, transformers have better F1 scores; however, **symp** and **bld** perform better with BOW.

For simplicity and initial understanding of the data and classification tasks, this chapter considered ICD-9 groupings as a set of binary classification problems. Alternatively, they can also be represented as a single multi-label classi-

fication problem. Each unique patient admitted to the hospital can have more than one diagnosis and be categorised into different groups or have more than one diagnosis from the same ICD-9 group. The rest of this thesis research considers multi-label representations of medical codes where the correlation between the medical codes are exploited in various classification problems.

Chapter 5

Word Embeddings for Multi-label Classification of Medical Text

The human body is a very complex system, and often patients admitted to hospitals with one initial prognosis or diagnosis have multiple related or unrelated chronic diseases. Modern medical practice emphasises the need to understand the patient as a whole [142, 143, 144, 145, 146]. The effects of different conditions may interact with each other and complicate the management of each disease [142]. This, in turn, leads to poorer outcomes, such as increased preventable hospital re-admissions, overall hospital re-admissions, and increased total medical and long term care costs [146, 145]. For example, a patient newly diagnosed with human epidermal growth factor receptor 2 (HER2) positive breast cancer may also have underlying, possibly undiagnosed, heart failure. This can be crucial, as some treatments for breast cancer can cause cardiac damage. Accurately identifying the symptoms of heart failure allows the physician to best balance the risks and benefits of such treatments.

The need to understand the patient as a whole motivates using multi-label classification techniques to improve the prediction of the medical codes of patients with multi-morbidity from the medical text. In single-label classi-

fication, as presented in Chapter 4, only one target variable is predicted per instance, whereas in multi-label classification, the goal is to predict multiple output variables for each input instance. In the above example, the patient is an instance with potential labels such as cancer, hypertension, heart failure, cholesterol and many more related and unrelated health complications. Moreover, the naive one-vs-all approach where each label is treated independently as binary classification, as presented in Chapter 4, suffers from considerable increase in required computational resource, both for training and predictions, when labels are thousands to millions [147].

This chapter considers techniques that enable maximising the feature extraction of the medical text of the embeddings layer for multi-label classification problems. The embeddings layer is considered a significant component for text representation [47]. Embeddings allow words to transform from isolated distinct symbols to a numerical representations. The distance between vectors and distance between words can be equated, and the relationship between words can be generated. This chapter considers multi-label problems presented in Table 2.4 as individual flat multi-label problems. This chapter extends the work presented in Chapter 4. More specifically, this chapter:

- acknowledges the multi-morbidity nature of patients and considers the multi-label variations of medical text classification to enhance the prediction of concurrent medical codes.
- presents new embeddings on the health-related text and compares several variations to embeddings models when dealing with an imbalanced multi-label medical text classification problem.
- presents a study exploring variations to tagging words, including the traditional part-of-speech (POS).
- provides a comparison of popular machine learning classifiers used in medical text classification, including neural networks.

- presents a detailed study and discussion of results, extended by varying the formations of embeddings, size of the embeddings and number of labels considered for the prediction of medical codes.

Results from this chapter show that among traditional machine learning classifiers, ECC-LR performed well. However, word embeddings-based neural networks out-performs ECC-LR and other traditional machine learning classifiers. Various fastText parameters were explored where results show skip-gram models out-performing CBOW models. FastText pre-trained models T100SG and T300SG are the best performing embeddings. To better understand the text, text sequence, and relationship, options such as POS tagging and concatenating embeddings were explored. However, in general, these options were not very successful in improving the overall performances. SOTA methods CAML and DR-CAML out-perform other classifiers for various levels of ICD-9 codes.

5.1 ICD-9 Level 1

In Chapter 4, for simplicity, 18 top-level ICD-9 groupings were considered as individual binary classification problems. However, as mentioned earlier, patients are usually admitted to hospitals with more than one related or unrelated medical condition or symptoms in most cases. It is more appropriate to consider these groups as a single multi-label classification problem. This section considers the 18 top-level ICD-9 groups as a single flat multi-label problem. In order to understand the differences in performances of multi-label classifiers', a detailed study comparing performance overall and for individual F1 scores is presented. In addition, comparisons of embeddings and fastText parameter choices are presented. This enabled a better understanding of the parameter choices, and as a result improve the performance of the multi-label classifiers. The option of tagging words, including POS, is also explored, however, there were no noticeable improvements to the F1-scores of the 18-labels. MIMIC-III

is used as the primary dataset for most experiments, and eICU is used as an additional dataset. Critical difference plots are presented as supportive statistical analysis. The Nemenyi posthoc test (95% confidence level) identifies statistical differences between learning methods. CD graph show the average ranking of individual F1 scores obtained using various language models. The lower the rank, the better a model is. The difference in average ranking is statistically significant if it is larger than the critical difference (CD), or in other words, if there is no bold line connecting the two settings.

5.1.1 Comparing Multi-label Classifiers

This section first considers the traditional multi-label classifiers followed by embeddings-based neural networks. Table 5.1 presents the overall performance of top-level ICD-9 groups for MIMIC-III data for traditional machine learning classifiers binary relevance (BR), classifier chains (CC), ensembles of classifier chains (ECC) and multi-label k-nearest neighbor (MLKNN). All experiments use the T300 word embeddings and 10-fold cross-validation. Experiments using BR and CC use logistic regression as the base classifier, and ECC explores SGD and LR as base classifiers. Only experiments with a ridge value of 1 for LR as the base classifiers is presented. However, other values were experimented with and logistic regression performed well and provided consistent F1 scores across a range of different ridge values. In comparison to BR, multi-label classifiers perform better overall, except for rank loss. For micro and macro F1, ECC-LR is better than other classifiers presented in Table 5.1.

Table 5.2 presents a comparison of label F1 scores between the classifiers and variations presented in Table 5.1 for the 18 top-level ICD-9 groups. Performance when considering the 18 groups as individual binary classification problems is also presented (BR-LR). All experiments use the T300 word embeddings and 10-fold cross-validation. As anticipated, using multi-label variations does provide an advantage over the individual binary classification case. Evidently, for most ICD-9 groups, ECC using logistic regression (LR) performs

Classifiers	Micro-F1	Macro-F1	Hamming Loss	Rank loss	Jaccard similarity
BR-LR, $R = 1$	0.724	0.618	0.171	0.105	0.579
MLKNN, $k=10$	0.668	0.500	0.203	0.145	0.520
CC-LR, $R = 1$	0.719	0.616	0.173	0.200	0.571
ECC-SGD, $E=50, I=10$	0.725	0.594	0.176	0.135	0.579
ECC-SGD, $E=50, I=30$	0.723	0.595	0.173	0.120	0.577
ECC-SGD, $E=50, I=100$	0.723	0.599	0.173	0.120	0.577
ECC-SGD, $E=100, I=10$	0.725	0.602	0.175	0.134	0.580
ECC-SGD, $E=100, I=30$	0.723	0.599	0.173	0.120	0.577
ECC-SGD, $E=100, I=100$	0.721	0.596	0.171	0.112	0.575
ECC-SGD, $E=500, I=10$	0.725	0.611	0.176	0.135	0.588
ECC-SGD, $E=500, I=30$	0.723	0.611	0.173	0.120	0.578
ECC-SGD, $E=500, I=100$	0.722	0.608	0.171	0.112	0.576
ECC-LR, $R=1, I=10$	0.734	0.645	0.183	0.170	0.587
ECC-LR, $R=1, I=30$	0.734	0.643	0.178	0.161	0.587
ECC-LR, $R=1, I=100$	0.735	0.643	0.178	0.155	0.588

Table 5.1: Overall performance of top level ICD-9 groups is presented for varying multi-label classifiers. I indicates number of iterations and E is the number of epochs to perform. T300 is used for embeddings. Bold is used to indicate the best results. For micro-F1, macro-F1 and Jaccard similarity, higher values are better; and for Hamming loss and rank loss, lower values are better.

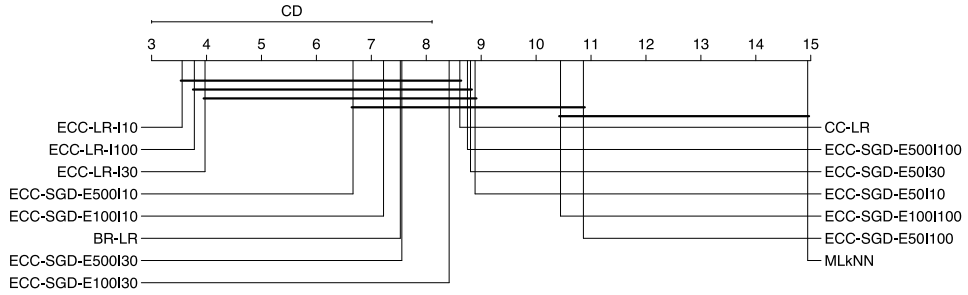


Figure 5.1: Critical difference plots. Nemenyi post-hoc test (95% confidence level), identifying statistical differences between multi-label classifier presented in Table 5.2. The lower the rank the better it is. The difference in average ranking is statistically significant if there is no bold line connecting the two settings.

best. Optimising the number of iterations and epochs for ECC can improve F1 score results. ECC-LR with a ridge value of $R = 1$ and the number of iterations

ICD-9	BR-LR	MLkNN	CC-LR	ECC-SGD			ECC-LR	
groups				Best case			Best case	
				F1	E =	I =	F1	I =
circ	0.932	0.921	0.932	<u>0.933</u>	100	10	0.932	30, 100
e+v	0.829	0.823	0.812	<u>0.831</u>	100	30	0.830	30, 100
endo	0.848	0.839	0.848	<u>0.851</u>	50	100	0.848	10, 30, 100
resp	0.777	0.703	0.770	<u>0.784</u>	500	10	0.782	30
inj	0.662	0.590	0.653	0.683	500	10	<u>0.686</u>	30
gen	0.731	0.657	0.731	0.735	500	30	<u>0.739</u>	10, 30, 100
diges	0.696	0.600	0.694	0.706	50	30	<u>0.713</u>	10
bld	0.571	0.494	0.571	0.577	50	100	<u>0.612</u>	10
symp	0.487	0.361	0.463	0.489	50	10	<u>0.552</u>	10
ment	0.542	0.299	0.538	0.562	500	30	<u>0.590</u>	10
nerv	0.543	0.376	0.522	0.530	100	10	<u>0.589</u>	10
inf	0.647	0.547	0.651	0.667	500	30	<u>0.683</u>	10
musc	0.298	0.086	0.302	0.272	500	30	<u>0.410</u>	10
pren	<u>0.594</u>	0.575	0.592	0.588	500	10	0.590	10
neop	0.703	0.500	0.705	0.709	500	10	<u>0.718</u>	10
skin	0.347	0.075	0.349	0.328	500	10	<u>0.413</u>	10
cong	0.384	0.294	0.383	0.361	500	100	<u>0.449</u>	30
preg	<u>0.592</u>	0.267	0.572	0.542	500	100	0.514	100

Table 5.2: Comparison of F1 scores for the 18 top level ICD-9 groups is presented for varying multi-label classifiers. I indicates number of iterations and E is the number of epochs to perform. T300 is used for embeddings. Bold is used to indicate F1 scores better than that of the BR-LR, and underline is used to indicate the best F1 score across all presented.

$I = 10$ achieves the best results overall. This observation is also supported by the critical difference plots presented in Figure 5.1, where ECC-LR-I10 has the lowest rank. Experiments across a range of different ridge values provided almost identical values for F1 score, hence only a ridge value of 1 is included in Tables 5.1 and 5.2.

Classifiers	Micro-F1	Macro-F1	Hamming Loss	Rank loss	Jaccard similarity
CNNText	0.823	0.743	0.157	0.298	0.694
BiGRU	0.808	0.711	0.178	0.318	0.673
CAML	0.837	0.755	0.154	0.277	0.713
DRCAML	0.838	0.761	0.153	0.275	0.715
HAN-LSTM	0.795	0.708	0.147	0.303	0.671
HAN-GRU	0.798	0.715	0.145	0.302	0.675
ECC-LR, I=10	0.734	0.645	0.183	0.170	0.587

Table 5.3: Overall performance of top-level ICD-9 groups is presented for various multi-label neural networks. T300 is used for embeddings. Bold is used to indicate the best results. For micro-F1, macro-F1 and Jaccard similarity, higher values are better; for Hamming loss and rank loss, lower values are better. ECC-LR with $I = 10$ is presented for comparison.

Table 5.3 considers overall performance of top-level ICD-9 groups for several multi-label neural networks. For consistency and comparison, T300 is used for embeddings. Results for ECC-LR with $I=10$ are also presented for direct comparison. Maximum input text sequence length is set to 3,000 tokens for all neural networks. The number of filters used in CNNText is 100 and filter sizes of 1,2,3,4 are used. Appendix C.1 provides a summary of results using other combination of parameters for CNNText. All experimental results using neural networks are obtained from validations based on training-testing scheme, and are averaged over three runs. Rank loss of ECC-LR is lower than that of the neural networks compared in this research. Overall, micro and macro F1 scores and Jaccard similarity are better for DRCAML, with a relatively big improvement over ECC-LR. This is also supported by critical difference plots presented in Figure 5.3, where DRCAML has the lowest rank and ECC-LR the highest rank with no bold line connecting the two indicating that difference in average ranking is statistically significant. Input for ECC-LR is static features of 300-dimensional vectors obtained using T300 for MIMIC-III data.

Figure 5.2 presents a comparison of individual F1 scores for the top-level ICD-9 groups for multi-label classifiers presented in Table 5.3 for the MIMIC-

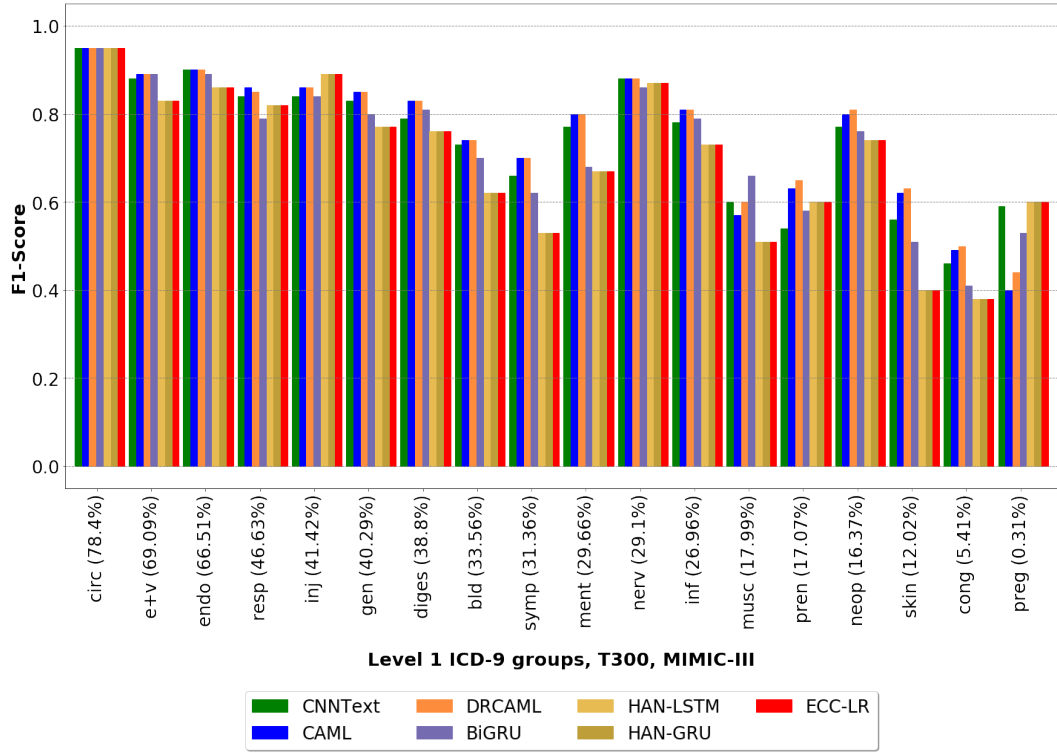


Figure 5.2: Comparison of F1 scores for the 18 top level ICD-9 groups is presented for various multi-label neural networks. T300 is used for embeddings. ECC-LR with $I = 10$ is presented for comparison. Most frequent label on the left to the least frequent on the right.

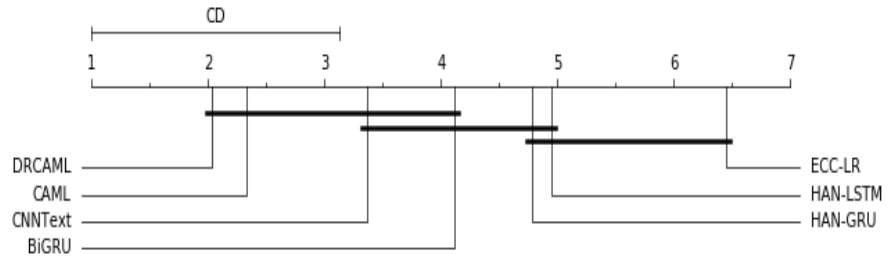


Figure 5.3: Critical difference plots. Nemenyi post-hoc test (95% confidence level), identifying statistical differences between multi-label classifier presented in Table 5.3 and Figure 5.2.

III dataset. The choice of classifiers impacts F1 scores across all 18 ICD-9 groups, but not necessarily always for the better. For most labels, ECC-LR performs similar to HAN variants presented. However, for **pren**, **inj** and **preg**, the F1 score of ECC-LR is better than some neural networks.

Option	Dimensions	Window Size	neg	Character minn	<i>n</i> -gram maxn	Loss Function	Epoch
I	50	5	10	5	5	softmax	5
II	50	3	10	5	5	softmax	5
III	50	7	10	5	5	softmax	5
IV	50	5	5	5	5	softmax	5
V*	50	5	10	0	0	softmax	5
VI	50	5	10	3	3	softmax	5
VII	50	5	10	5	5	hierarchical softmax	5
VIII	50	5	10	5	5	negative sampling	5
IX	50	5	10	5	5	softmax	10

Table 5.4: Variations of parameter choices for embeddings trained using fastText are presented. These options are used for both CBOW and skip-gram. Option “I” contains the exact same parameter choices as the published model W300. “neg” refers to the number of negative samples per positive sample. [minn, maxn] refers to minimum length and maximum length respectively. *Option V sets maxn = 0, this means no sub-words will be used by fastText. Hence, the model should give similar results to that of word2vec.

5.1.2 FastText Parameter Choices

Table 4.3 in Chapter 4, presented embeddings models trained to the exact same specifications as general text trained published models. To better understand the effects of fastText parameter choices on performance of multi-label classifiers, a more detailed comparison of variations of specifications for training embeddings is presented in this chapter. The combination of variations to parameter choices is presented in Table 5.4. The learning rate used to train all of the variations presented is 0.05. For simplicity, all dimensions were set to 50. The twoword representation models are skip-gram and CBOW.

Figure 5.4 presents a comparison of F1 scores of the fastText parameter choices I-IX for both CBOW and skip-gram embeddings as outlined in Table 5.4 for MIMIC-III data. For simplicity, BR with logistic regression is used as the classifier for these experiments. Results correspond to 18 embeddings $\times$ 13

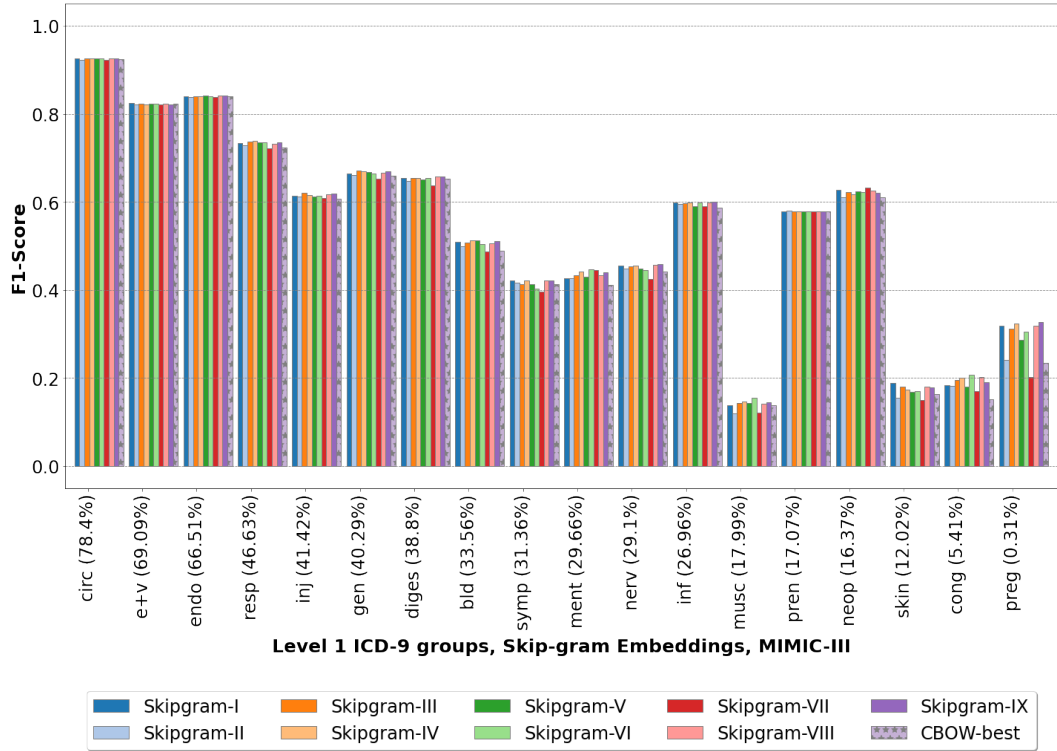


Figure 5.4: Comparison of F1 scores for top level ICD-9 groups for embeddings obtained by varying fastText parameters, as presented in Table 5.4. Options I to IX match that of Table 5.4. Embeddings are trained on TREC data and the classifier used for experiments is BR with logistic regression and $R = 1$. Best F1 scores for CBOW models are presented for comparison (see Appendix Figure C.1 for more details of CBOW model F1 scores). Most frequent label on the left to the least frequent on the right.

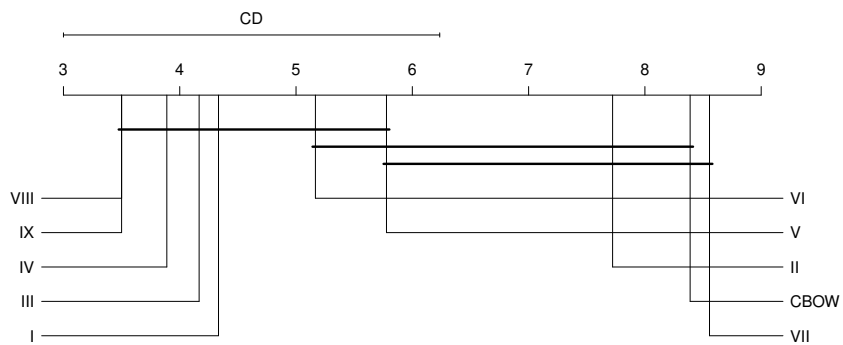


Figure 5.5: Critical difference plots. Nemenyi post-hoc test (95% confidence level), identifying statistical differences between learning methods. Best F1 scores for CBOW models are presented for comparison.

classifiers $\times$ 10 folds cv for a total of 2340 tests. The best F1 score among the models using the options presented in Table 5.4 for CBOW is also presented.

Models	Dimensions	Source Data	Train Time	Model Size
T100SG	100	TREC	7 hours	4GB
T300SG	300	TREC	28 hours	13GB
T600SG	600	TREC	51 hours	23GB

Table 5.5: Skip-gram embeddings, with dimension details, training times and embeddings model sizes. See Table 4.3 for details on other embeddings models.

Evidently, for all 18 groups, skip-gram out-performs CBOW. Option I has the same specifications as the W300 embeddings. FastText parameters impact F1 scores across all 18 ICD-9 groups, but not necessarily always for the better. Thus, care must be taken when selecting these parameters. Critical difference plots presented in Figure 5.5 indicates the best options as VIII and IX.

For this research, in addition to the embeddings presented in Table 4.3, skip-gram models T100SG, T300SG, T600SG were also trained using fast-Text. Table 5.5 presents details of the skip-gram embeddings used in this research with details of dimensional size, source data, model size and training time¹. The embeddings presented in Table 5.5 are trained as per option I in Table 5.4 for comparison. The F1 scores of comparison between embeddings and classifiers are presented in Section 5.1.5.

5.1.3 Concatenating Embeddings

The option of splitting the free-form medical data into sections and concatenating the embeddings was also explored. This enabled the option of capturing the meaning each sub-section of discharge summary with separate embeddings representations. The discharge summary in MIMIC-III is split into seven logical sections: Admission Date, Past Medical History, Pertinent Results, Brief Hospital Course, Medications on Admission, Discharge Diagnosis and Followup Instructions. Embeddings for each section can be obtained and concatenated

¹Processing was run on a 4 core Intel i7-6700K CPU @ 4.00GHz with 64GB of RAM.

(sample text shown below in Figure 5.7). For example, if a 50-dimensional embeddings model is used, the resulting concatenated embeddings has 350 dimensions. If the discharge summary does not include any of the sub-sections mentioned above, then the respective embeddings are all zeros. For hospital admissions with more than one available discharge summary, all the summaries are first embedded independently and then averaged into one final embedding.

Another variation considers concatenating statistical outcomes of the embeddings from each of the sections of a given hospital admission. This research looks at the minimum, maximum, mean, standard deviation, lower quartile and upper quartile of the embeddings for these experiments. For example, for each hospital admission the discharge summary will have 7 splits, hence 7 embeddings for each split. The mean of these 7 embeddings are calculated. Similarly, minimum, maximum, standard deviation, lower quartile and upper quartile are also calculated. Hence, the resulting embeddings will have six times as many dimensions as the original one.

Figure 5.6 presents a comparison of embeddings formed by concatenating embeddings. The base embeddings used here are T50 and ECC-LR is used for classification. CONCAT300 is formed by concatenating the embeddings of the statistical outcomes, i.e. $\text{CONCAT300} = 50 \text{ dim} \times (\text{min} + \text{max} + \text{mean} + \text{sd} + \text{q1} + \text{q3})$. CONCAT350 is formed by concatenating the embeddings of the seven text splits $7 \times 50 \text{ dim}$. In comparison, both CONCAT300 and CONCAT350 improve F1 scores relative to the base embeddings T50 except for the ICD-9 group **pren**. CONCAT350 generally performs better than CONCAT300. However, the T300 embeddings outperform both CONCAT300 and CONCAT350 across all 18 groups. Also, the improvements that CONCAT300 and CONCAT350 produce over T50 are not replicated for larger embeddings. For instance, when starting with T300 and generating CONCAT1800, or CONCAT2100, no significant improvements are observed. This may be because T300 already performs much better than T50, possibly not leaving much room for further improvement.

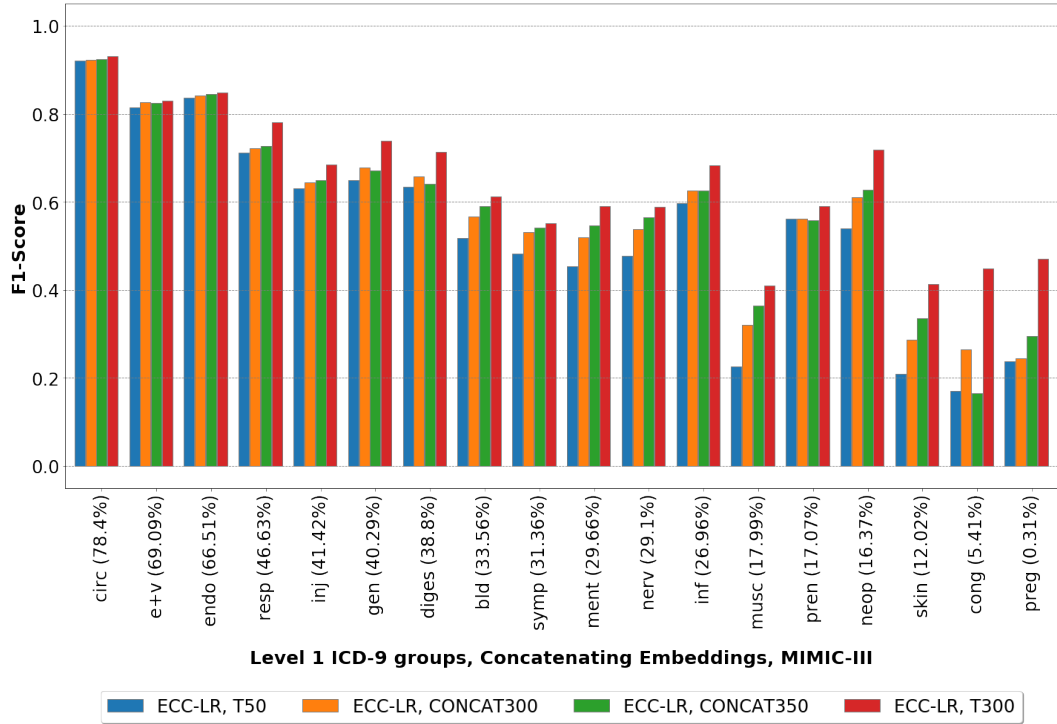


Figure 5.6: A comparison of F1 scores for top-level ICD-9 groups is presented. F1 scores for two models T50 and T300, and two variations of concatenations CONCAT300 and CONCAT350 are presented. All experiments used ECC with logistic regression as the base classifier, using a ridge value of one.

5.1.4 Tagging Words

Two variations of tagging words in free-form medical text were considered as options for improving the understanding of free-form medical text and potentially improving the accuracy of predictions. The first variant, part-of-speech (POS), is a technique used to automatically identify the syntactic categories of words in a given sentence. POS provides information about each word and its neighbors. Knowing whether a word is a noun or a verb can help in categorising the likely neighboring words and the syntactic structure around the word. Common examples of such POS tags are: noun, verb, adjective, adverb, pronoun, preposition, conjunction and interjection. POS taggers are created using Natural Language Toolkit (NLTK²) [148], where if the input text is:

History of Present Illness 54 year old female with recent diagnosis
of ulcerative colitis on mercaptopurine

²<http://nltk.org/>

output is:

HistoryNN ofIN PresentNNP IllnessNNP 54CD yearNN oldJJ fe-
maleNN withIN recentJJ diagnosisNN ofIN ulcerativeJJ colitisNN
onIN mercaptopurineJJ

NN indicates a noun, IN is referring to preposition or conjunction, NNP is referring to a proper noun, CD is referring to numeral and JJ is referring to adjective.

The second variation is text split tags where the words of MIMIC-III discharge summaries were tagged using the text splits presented in Section 5.1.3, in-order to incorporate the collective meanings of each split. Tokens in each of these sections are tagged with 0_, 1_, 2_, 3_, 4_, 5_, 6_ for text in the seven splits Admission Date, Past Medical History, Pertinent Results, Brief Hospital Course, Medications on Admission, Discharge Diagnosis and Followup Instructions, respectively as shown in Figure 5.7.

Figure 5.8 presents a comparison of F1 scores for top level ICD-9 groups among the MIMIC-III discharge summaries with POS tags, with text split tags and for the raw text without any tagging. Evidently, except for categories **bld**, **sym**, and **preg**, using a POS tagger does not improve the F1 scores. For categories **bld**, **sym** and **preg**, the use of the POS tagger improves the F1 score, from 0.612 to 0.616, from 0.552 to 0.555, and from 0.470 to 0.491, respectively. When using the text split tagger, the F1 score for **circ** is equivalent to the no-tagger case, and for category **endo** there is a small improvement from 0.848 to 0.850.

Both POS and text split tags were explored to better understand the word representations and possibly improve the overall performance. However, with a very few exceptions, in general there were no significant improvements in F1 scores. In fact, with many ICD-9 groupings there was a drop in F1 score when compared to non-tagged embeddings with ECC-LR.

5.1.5 Comparison between Embeddings and Classifiers

Table 5.6 presents an extensive comparison of the overall performance of top-level ICD-9 groups among various embeddings and classifiers for MIMIC-III.

Text split for MIMIC-III, sample text:

Admission Date: [**2164-4-23**] Discharge Date: [**2164-4-27**] Date of Birth: [**2096-1-7**] Sex: M Service: CARDIOTHORACIC Allergies: No Known Allergies Adverse

Past Medical History: diabetes type II diagnosed [**2160**] controlled on oral agents w/ hyperlipidemia hypertension

Pertinent Results: [**2164-4-23**] Echo PREBYPASS No spontaneous echo contrast is seen in the body of the left atrium or left atrial

Brief Hospital Course: Mr. [**Known lastname**] 68 yr old male with history of MI, developed worsening chest pain underwent cath which revealed

Medications on Admission: ATENOLOL 50 mg Daily. LIPITOR 20 mg Daily. PLAVIX 75 mg Daily. started on [**2164-4-14**] LD 417 DILTIAZEM HCL 240 mg Daily.

Discharge Diagnosis: Diabetic. blood in your vomit. Hypertension Chronic renal

Followup Instructions: Please call Dr office to schedule

output is:

```
0_Admission 0_Date 0_2164423 0_Discharge 0_Date 0_2164427 0_Date
0_of 0_Birth 0_209617 0_Sex 0_M 0_Service 0_CARDIOTHORACIC
0_Allergies 0_No 0_Known 0_Allergies 0_Adverse
1_Past 1_Medical 1_History 1_diabetes 1_type 1_II 1_diagnosed 1_2160
1_controlled 1_on 1_oral 1_agents 1_hyperlipidemia 1_hypertension
2_Pertinent 2_Results 2_2164423 2_Echo 2_PREBYPASS 2_No
2_spontaneous 2_echo 2_contrast 2_is 2_seen 2_in 2_the 2_body 2_of
2_the 2_left 2_atrium 2_or 2_left 2_atrial
3_Brief 3_Hospital 3_Course 3_Mr 3_Known 3_lastname 3_68 3_yr 3_old
3_male 3_with 3_history 3_of 3_MI 3_developed 3_worsening 3_chest 3_pain
3_underwent 3_cath 3_which 3_revealed
4_Medications 4_on 4_Admission 4_ATENOLOL 4_50 4_mg 4_Daily
4_LIPITOR 4_20 4_mg 4_Daily 4_PLAVIX 4_75 4_mg 4_Daily 4_started
4_on 4_2164414 4_LD 4_417 4_DILTIAZEM 4_HCL 4_240 4_mg 4_Daily
5_Discharge 5_Diagnosis 5_Diabetic 5_blood 5_in 5_your 5_vomit
5_Hypertension 5_Chronic 5_renal
6_Followup 6_Instructions 6_Please 6_call 6_Dr 6_office 6_to 6_schedule
```

Figure 5.7: Sample text split of MIMIC-III data before and after tagging.

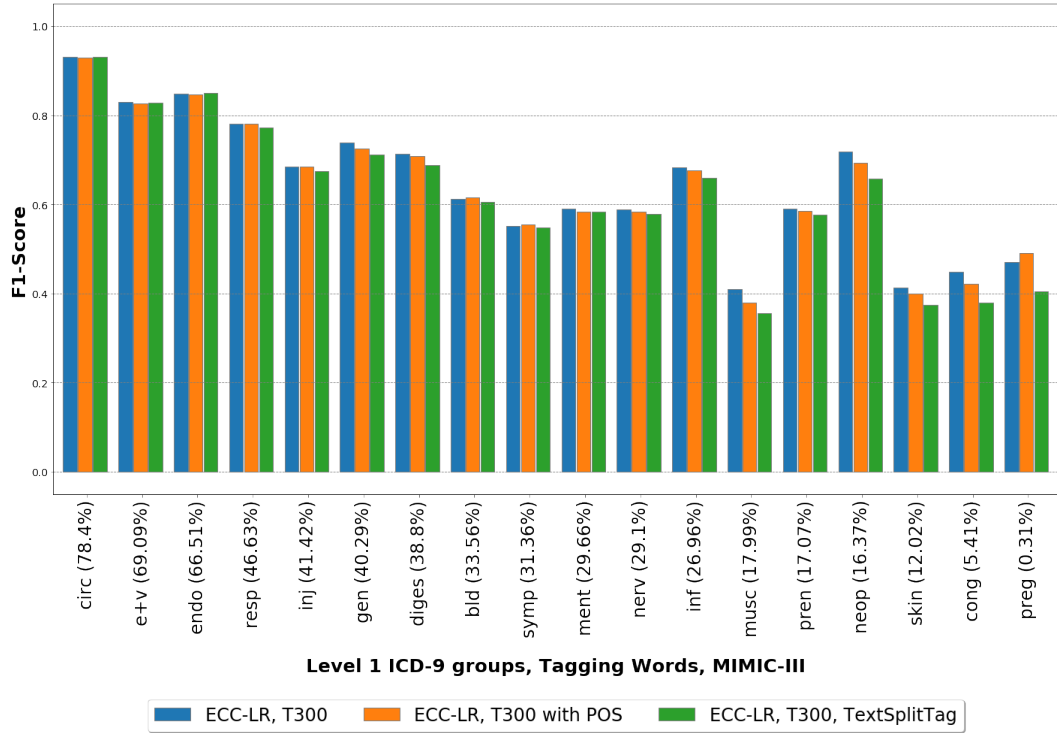


Figure 5.8: A comparison of F1 scores for top-level ICD-9 groups between discharge summaries with and without a POS tagger and with text split tags. All experiments used ECC with logistic regression as the base classifier, using a ridge value of one.

For ECC-LR, increasing dimensionality of the embeddings, and in essence, the number of input features, improves the overall performance, except T900, which is better than T600 but not when compared to the skip-gram model T600SG. Moreover, for ECC-LR, medical text pre-trained embeddings are better than the general text pre-trained embeddings W300.

For the neural networks presented in Table 5.6, embedding sizes of up to 300 perform better than more high-dimensional embeddings. For instance, using T300SG with CNNTTest performs better than using the published medical pre-trained embeddings BSV700 for CNNTText. Overall, considering the micro F1 score for embeddings-based SOTA methods CAML and DRCAML, T100SG performs better than higher-dimensional embeddings, including the general text pre-trained embeddings W300. For macro F1 score, T300SG performs best and consistently so across all multi-label neural networks presented in this research. Critical difference plots for T100SG and T300SG for multi-label

Classifiers	Embeddings	Micro-F1	Macro-F1	Hamming Loss	Rank loss	Jaccard similarity
ECC-LR	W300	0.730	0.639	0.187	0.174	0.582
ECC-LR	T300	0.734	0.645	0.183	0.170	0.587
ECC-LR	T300SG	0.738	0.650	0.180	0.168	0.592
ECC-LR	T600	0.742	0.659	0.175	0.158	0.596
ECC-LR	T600SG	0.745	0.668	0.173	0.155	0.601
ECC-LR	T900	0.744	0.665	0.177	<u>0.151</u>	0.598
CNNText	T100SG	0.824	0.734	0.158	0.304	0.692
CNNText	W300	0.823	0.741	0.159	0.302	0.691
CNNText	T300	0.823	0.743	0.157	0.298	0.693
CNNText	T300SG	0.826	0.743	0.161	0.298	0.694
CNNText	T600	0.820	0.717	0.158	0.301	0.690
CNNText	T600SG	0.817	0.724	0.161	0.307	0.685
CNNText	BSV700	0.822	0.726	0.164	0.303	0.691
CAML	T100SG	0.842	0.731	0.146	0.271	0.721
CAML	W300	0.841	0.766	0.147	0.269	0.721
CAML	T300	0.837	0.755	0.154	0.277	0.712
CAML	T300SG	0.840	0.770	0.149	0.271	0.719
CAML	T600	0.839	0.758	0.152	0.274	0.716
CAML	T600SG	0.840	0.766	0.150	0.271	0.718
DRCAML	T100SG	<u>0.843</u>	0.733	<u>0.145</u>	0.270	<u>0.722</u>
DRCAML	W300	0.839	0.757	0.151	0.273	0.717
DRCAML	T300	0.838	0.761	0.153	0.275	0.715
DRCAML	T300SG	0.840	<u>0.772</u>	0.150	0.272	0.719
DRCAML	T600	0.839	0.759	0.153	0.275	0.715
DRCAML	T600SG	0.840	0.766	0.150	0.272	0.717
BiGRU	T100SG	0.821	0.731	0.166	0.306	0.693
BiGRU	W300	0.827	0.737	0.167	0.298	0.698
BiGRU	T300	0.808	0.711	0.178	0.318	0.673
BiGRU	T300SG	0.830	0.752	0.161	0.291	0.703

Table 5.6: Comparison of overall performance of top level ICD-9 groupings among various multi-label neural classifiers with various embeddings for MIMIC-III data. ECC-LR, with R=1 and I = 10 is presented for comparison. Bold is used to indicate the best results for each classifier and underline is used to indicate best overall measure. For micro-F1, macro-F1 and Jaccard similarity, higher values are better; and for Hamming loss and rank loss lower values are better.

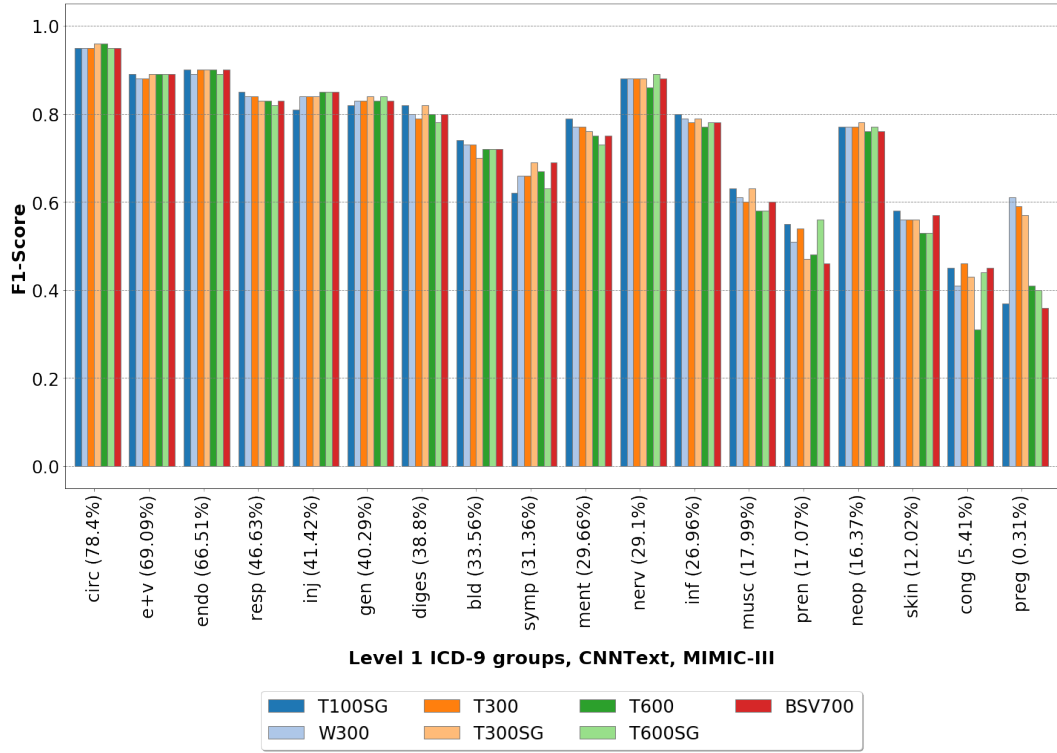


Figure 5.9: Comparison of individual F1 scores of top level ICD-9 groups with various embeddings for MIMIC-III data using CNNTText. Most frequent label on the left to the least frequent on the right.

classifiers are presented in Figure 5.10.

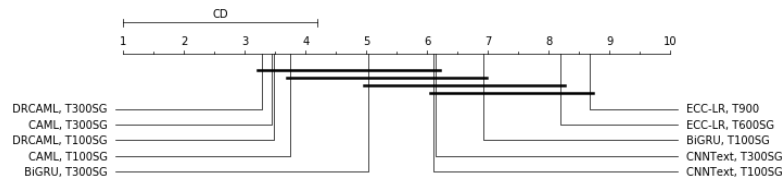


Figure 5.10: Critical difference plots for ICD-9 Level 1 groups. Nemenyi post-hoc test (95% confidence level), identifying statistical differences between selective multi-label neural networks with varied embeddings presented in Table 5.6 for MIMIC-III data.

Figure 5.9 presents the F1 scores of labels for various embeddings using CNNTText for ICD-9 top-level groups presented in Table 5.6. For most labels, skip-gram models perform better than other embeddings presented. However, for **preg**, W300 performs better than health-related pre-trained embeddings. The published embeddings BSV700 performs poorly for category **preg** in com-

Level 1 ICD-9 codes, 18 labels, eICU						
Models	Embeddings	Micro -F1	Macro-F1	Hamming Loss	Rank loss	Jaccard similarity
CNNText	T100SG	0.642	0.477	<u>0.099</u>	0.372	<u>0.536</u>
CNNText	W300	0.639	0.481	0.104	0.361	0.529
CNNText	T300SG	0.632	0.487	0.105	0.374	0.529
BiGRU	T100SG	0.623	0.498	0.138	0.283	0.485
BiGRU	W300	0.576	0.453	0.164	0.309	0.433
BiGRU	T300SG	0.617	0.501	0.129	0.322	0.485
CAML	T100SG	0.645	0.511	0.125	0.267	0.523
CAML	W300	0.639	0.504	0.130	0.266	0.509
CAML	T300SG	0.638	0.504	0.130	0.267	0.508
DRCAML	T100SG	<u>0.647</u>	<u>0.512</u>	0.123	0.268	0.526
DRCAML	W300	0.640	0.504	0.129	0.267	0.510
DRCAML	T300SG	0.639	0.499	0.128	<u>0.261</u>	0.515

Table 5.7: Comparison of overall performance of ICD-9 levels 1 among various multi-label neural classifiers with various embeddings for eICU. Bold is used to indicate the best results for each embeddings and underline is used to indicate best overall measure. For micro-F1, macro-F1 and Jaccard similarity, higher values are better; and for Hamming loss and rank loss, lower values are better.

parison.

Table 5.7 and Figure 5.11 present the overall performance and F1 scores of level ICD-9 groups for eICU data. Multi-label neural networks with embeddings T100SG, T300SG and W300 are experimented with using eICU data with a maximum input text sequence size of 1,250 tokens. It is essential to point out that sequence length of 99.7% of the eICU data are less than or equal to 1,250 tokens (see Figure 2.2). Micro and macro F1 scores of DRCAML with T100SG is the best among all the classifiers presented. This observation is also supported by critical difference plots presented in Figure 5.12. Figure 5.11 shows very poor performance across the networks for infrequent labels compared to frequent labels, with the exception of F1 score of 0.42 for BiGRU and 0.13 for CNNText for *preg*.

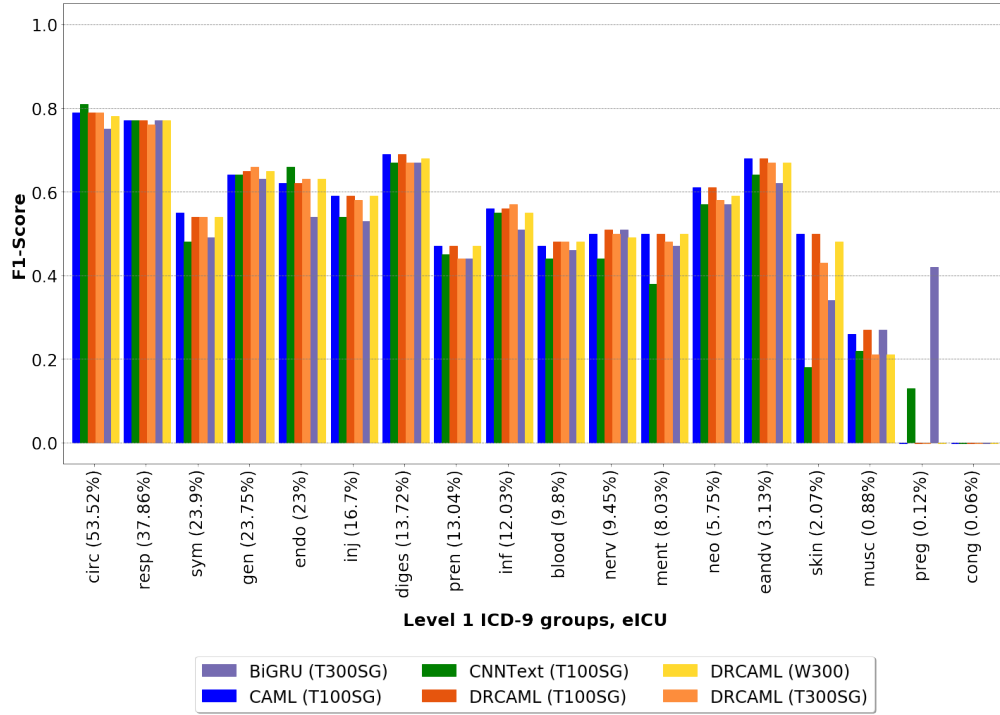


Figure 5.11: Comparison of individual F1 scores of top level ICD-9 groups with various embeddings and classifiers for eICU data. Most frequent label on the left to the least frequent on the right.

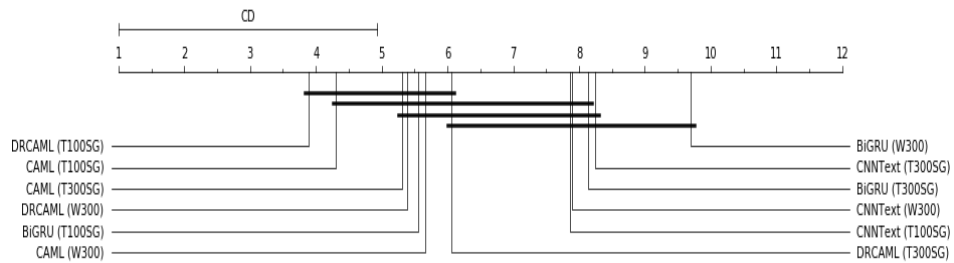


Figure 5.12: Critical difference plots for ICD-9 Level 1 groups. Nemenyi post-hoc test (95% confidence level), identifying statistical differences between selective multi-label neural networks with various embeddings presented in Table 5.7 for eICU data.

5.1.6 Summary

This section presented a detailed study of multi-label top-level ICD-9 groups, where several classifiers and embeddings were explored. Among the traditional machine learning classifiers, ECC-LR performed well; however, neural networks performed better. The skip-gram model performs better than CBOW,

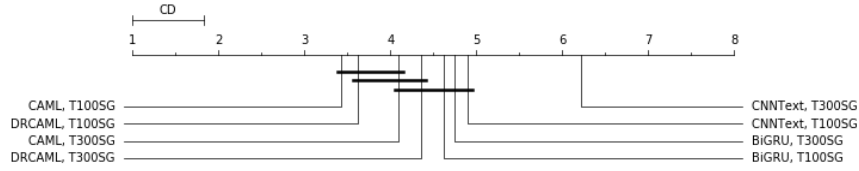


Figure 5.13: Critical difference plots for Level 2 ICD-9 codes, 158 labels. Nemenyi post-hoc test (95% confidence level), identifying statistical differences between multi-label neural networks presented in Table 5.8 for MIMIC-III data.

and T100SG and T300SG are the better performing embeddings among those presented. Options of tagging words and concatenating embeddings were explored, however, results did not show improvement in the overall performances, with very few exceptions.

5.2 ICD-9 Level 2 and 3

In this section, Level 2 and 3 ICD-9 codes are modelled as flat multi-label problems for MIMIC-III and eICU data. Table 5.8 presents overall comparison for both Level 2 and 3 of ICD-9 codes for MIMIC-III data. Input sequence length was limited to a maximum of 3,000 tokens. For the MIMIC-III data Level 2 ICD-9 codes include 158 labels, where T100SG performs better than T300SG for all classifiers. Embeddings-based methods CAML and DRCAML are the two better performing networks. These observations are further supported by critical difference plots for T100SG and T300SG for multi-label classifiers presented in Figure 5.13.

Figure 5.14 presents a comparison of F1 scores for Level 2 ICD-9 code labels for MIMIC-III data, where the frequency of occurrence of the labels is ordered from high (left) to low (right). CAML and DRCAML with T100SG embeddings are used for direct comparison. Label frequency less than one percent is indicated. For higher frequencies, both methods perform on par with each other. However tail-end labels include multiple labels with F1 scores of zero.

Level 2 ICD-9 codes, 158 labels, MIMIC-III						
Models	Embeddings	Micro-F1	Macro-F1	Hamming Loss	Rank loss	Jaccard similarity
CNNText	T100SG	0.692	0.404	0.050	0.345	0.530
BiGRU	T100SG	0.700	0.414	0.051	<u>0.298</u>	0.540
CAML	T100SG	<u>0.716</u>	<u>0.428</u>	<u>0.047</u>	0.303	<u>0.560</u>
DRCAML	T100SG	<u>0.716</u>	0.415	<u>0.047</u>	0.305	0.559
CNNText	T300SG	0.670	0.356	0.051	0.386	0.504
BiGRU	T300SG	0.679	0.378	0.053	0.332	0.519
CAML	T300SG	0.709	0.402	0.048	0.301	0.552
DRCAML	T300SG	0.710	0.391	0.048	0.306	0.552
Level 3 ICD-9 codes, 923 labels, MIMIC-III						
CNNText	T100SG	0.582	0.214	0.013	0.466	0.421
BiGRU	T100SG	0.577	0.215	0.014	0.458	0.419
CAML	T100SG	<u>0.635</u>	<u>0.264</u>	<u>0.012</u>	<u>0.392</u>	<u>0.474</u>
DRCAML	T100SG	0.633	0.241	<u>0.012</u>	0.403	0.471
CNNText	T300SG	0.563	0.203	0.014	0.481	0.402
BiGRU	T300SG	0.581	0.220	0.014	0.452	0.423
CAML	T300SG	0.620	0.231	0.013	0.407	0.456
DRCAML	T300SG	0.617	0.220	<u>0.012</u>	0.424	0.453

Table 5.8: Comparison of overall performance of ICD-9 Level 2 and 3 among various multi-label neural classifiers with various embeddings for MIMIC-III. Bold is used to indicate the best results for each embeddings and underline is used to indicate best overall measure. For micro-F1, macro-F1 and Jaccard similarity, higher values are better; and for Hamming loss and rank loss, lower values are better.

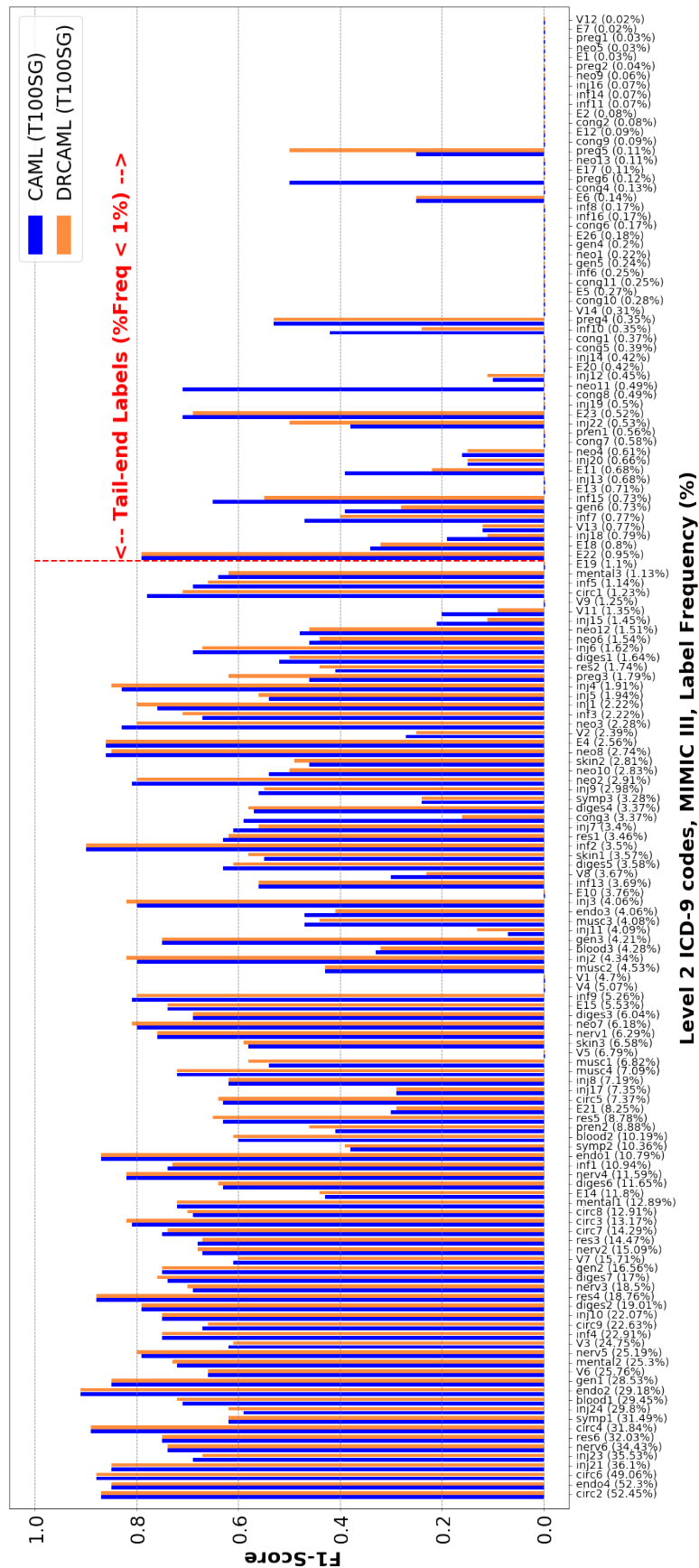


Figure 5.14: Comparison of F1 scores of Level 2 ICD-9 codes for MIMIC data between CAML (T100SG) and DRCAML (T100SG).

Level 2 ICD-9 codes, 93 labels, eICU						
Models	Embeddings	Micro -F1	Macro-F1	Hamming Loss	Rank loss	Jaccard similarity
CNNText	T100SG	0.554	0.297	0.032	0.369	0.416
BiGRU	T100SG	0.570	0.307	<u>0.031</u>	0.335	<u>0.440</u>
CAML	T100SG	0.567	0.314	0.034	0.280	0.427
DRCAML	T100SG	<u>0.572</u>	<u>0.320</u>	0.033	0.294	0.434
CNNText	T300SG	0.466	0.243	0.053	0.289	0.335
BiGRU	T300SG	0.515	0.305	0.043	0.289	0.358
CAML	T300SG	0.557	0.314	0.036	<u>0.274</u>	0.412
DRCAML	T300SG	0.560	0.310	0.035	0.282	0.416
Level 3 ICD-9 codes, 316 labels, eICU						
CNNText	T100SG	0.469	0.162	0.012	0.483	0.339
BiGRU	T100SG	0.487	0.215	<u>0.010</u>	0.458	<u>0.391</u>
CAML	T100SG	<u>0.504</u>	<u>0.216</u>	0.012	<u>0.362</u>	0.375
DRCAML	T100SG	0.501	0.214	0.012	0.381	0.384
CNNText	T300SG	0.424	0.166	0.017	0.391	0.285
BiGRU	T300SG	0.477	0.208	0.012	0.450	0.360
CAML	T300SG	0.494	0.202	0.013	0.366	0.358
DRCAML	T300SG	0.497	0.189	0.012	0.383	0.364

Table 5.9: Comparison of overall performance of ICD-9 Level 2 and 3 among various multi-label neural classifiers with various embeddings for eICU. Bold is used to indicate the best results for each embeddings and underline is used to indicate best overall measure. For micro-F1, macro-F1 and Jaccard similarity, higher values are better; and for Hamming loss and rank loss, lower values are better.

Table 5.8 includes an overall comparison of individual F1 scores for Level 3 ICD-9 code labels for MIMIC-III data. As for Level 2 ICD-9 codes, SOTA methods CAML and DRCAML with T100SG perform better overall. Appendix Figure C.4 presents a comparison of F1 scores of individual 923 labels between CNNText and CAML with T100SG for ICD-9 Level 3 codes.

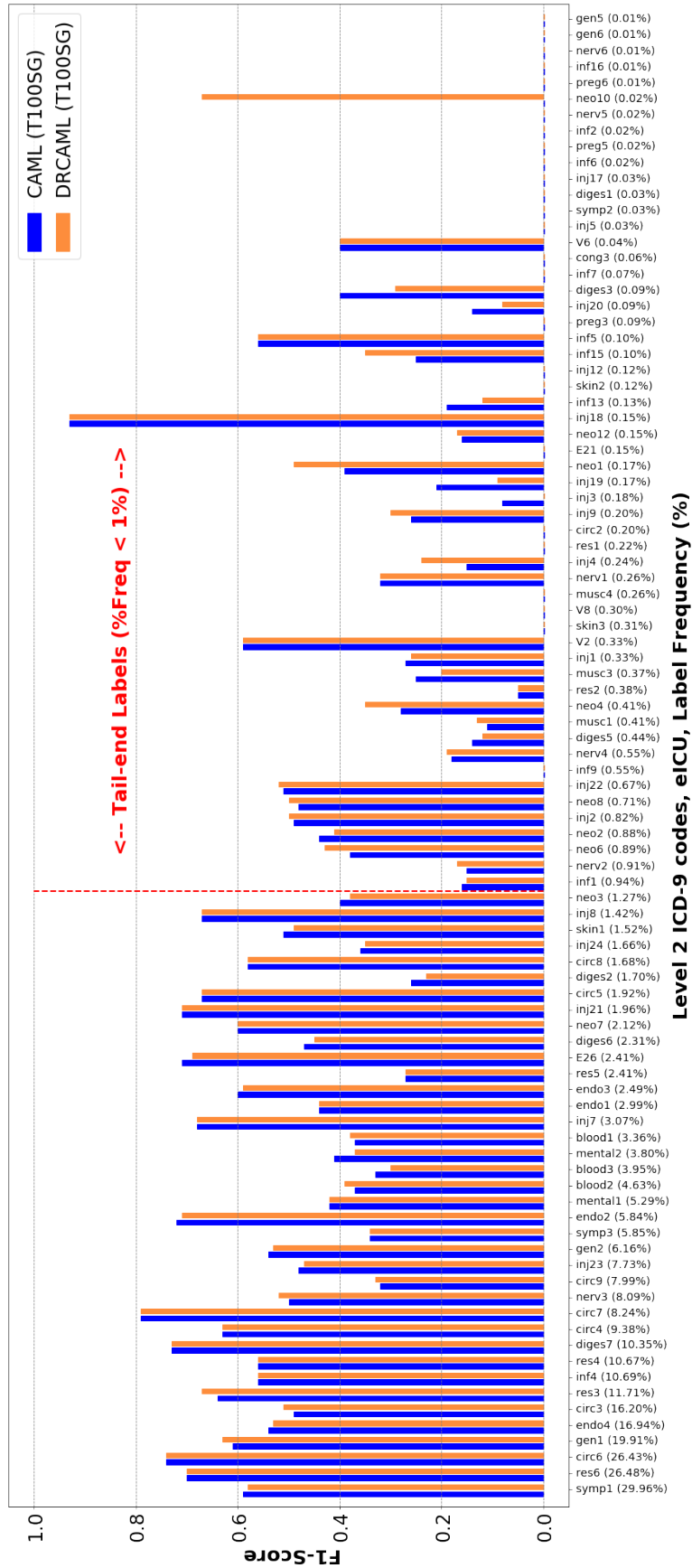


Figure 5.15: Comparison of F1 scores of Level 2 ICD-9 codes for eICU data between CAML (T100SG) and DRCAML (T100SG).

Table 5.9 presents overall comparison for both Level 2 and 3 of ICD-9 codes for eICU data. Input sequence length was limited to a maximum of 1,250 tokens. For eICU data, Level 2 ICD-9 codes include 93 labels, where T100SG performs better than T300SG for all classifiers. Micro and macro F1 scores for CAML and DRCAML are better than for other networks. Critical difference plots for T100SG and T300SG for multi-label classifiers are presented in Figure 5.16 also supports these observations.

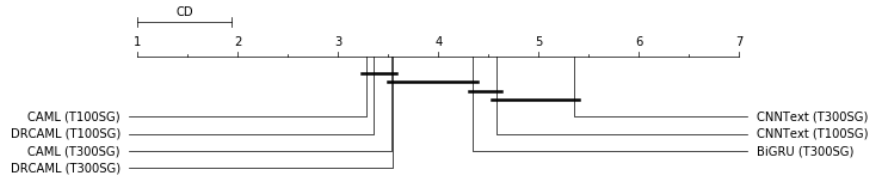


Figure 5.16: Critical difference plots for Level 2 ICD-9 codes, 93 labels. Nemenyi post-hoc test (95% confidence level), identifying statistical differences between multi-label neural networks presented in Table 5.9 for eICU data.

Figure 5.15 presents a comparison of F1 scores for Level 2 ICD-9 code labels for eICU data, where the frequency of occurrence of the labels is ordered from high (left) to low (right). CAML and DRCAML with T100SG embeddings are used for direct comparison. A label frequency of less than one percent is indicated. The eICU data has a higher percentage for tail-end labels at Level 2 than MIMIC III. As observed in MIMIC-III, for higher frequencies, both methods perform on par with each other. Tail-end labels include many labels with F1 scores of zero.

Table 5.9 also includes an overall comparison of individual F1 scores for Level 3 ICD-9 code labels for eICU data. In addition to CAML, BiGRU performs better overall in comparison to CNNTxt. Appendix Figure C.3 presents a comparison of F1 scores of individual 316 labels between BiGRU and CAML for ICD-9 Level 3 codes. Critical difference plots for T100SG and T300SG for multi-label classifiers are presented in Appendix Figure C.2.

5.2.1 Summary

This section presented experimental findings for Level 2 and 3 of ICD-9 codes for both MIMIC-III and eICU. Based on the findings in Section 5.1, only neural networks CAML, DRCAML, CNNTText and BiGRU were used for experiments with embeddings T100SG and T300SG. For both MIMIC-III and eICU, the best choice is CAML with T100SG for Level 2 and 3 of ICD-9 codes. In addition, 100-dimensional embeddings T100SG outperformed 300-dimensional T300SG. Tail-end labels were explicitly considered, where F1 scores of zero were observed across many labels.

5.3 Case Studies

This section considers embeddings-based multi-label neural networks for case studies presented in Table 2.4 for both MIMIC-III and eICU data. Predictions of medical codes are presented as a flat multi-label problem for codes associated with cardiovascular disease, COVID-19 patient shielding, and systemic fungal or bacterial infections. Comparison between label F1 scores are also presented for selected classifiers. Critical difference plots are used to determine the classifiers used for label F1 score comparisons, where only the low ranking models are presented.

Table 5.10 presents micro and macro F1 scores for the three case studies using MIMIC-III data, where classifiers embeddings and a maximum number of tokens in input text sequence is varied. Evidently, for all classifiers, increase in the number of tokens of the input text shows considerable improvement in performances.

Table 5.11 presents micro and macro F1 scores for the three case studies using eICU data, where classifiers embeddings and a maximum number of tokens in input text sequence is varied. The eICU data includes only 1% of text sequence with more than 512 tokens. Hence, when the maximum number of tokens in increased from 512 to 1,250, micro and macro F1 scores show

almost no improvements.

Classifier	Embeddings	Tokens	Cardiovascular		COVID-19		Fungal or Bacterial	
			Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
CAML	T100SG	512	0.71	0.20	0.41	0.23	0.41	0.26
CAML	T100SG	3,000	0.77	0.24	0.61	0.40	0.57	0.40
CAML	T300SG	3,000	0.77	0.24	0.54	0.37	0.62	0.37
DRCAML	T100SG	512	0.71	0.19	0.44	0.27	0.39	0.25
DRCAML	T100SG	3,000	0.77	0.24	0.60	0.39	0.62	0.38
DRCAML	T300SG	3,000	0.77	0.25	0.54	0.40	0.62	0.37
BiGRU	T100SG	512	0.68	0.19	0.39	0.19	0.44	0.27
BiGRU	T100SG	3,000	0.77	0.26	0.59	0.31	0.62	0.37
BiGRU	T300SG	3,000	0.74	0.24	0.52	0.33	0.57	0.36
CNNText	T100SG	512	0.72	0.22	0.49	0.31	0.41	0.24
CNNText	T100SG	3,000	0.74	0.30	0.58	0.42	0.59	0.32
CNNText	T300SG	3,000	0.75	0.31	0.59	0.41	0.57	0.40

Table 5.10: Comparison of micro-F1, macro-F1 of case studies among various multi-label neural classifiers with various embeddings and sequence length for MIMIC-III data. Bold is used to indicate the best results.

Classifier	Embeddings	Tokens	Cardiovascular		COVID-19		Fungal or Bacterial	
			Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
CAML	T100SG	512	0.68	0.41	0.60	0.32	0.64	0.25
CAML	T100SG	1,250	0.68	0.41	0.60	0.32	0.64	0.25
CAML	T300SG	1,250	0.68	0.40	0.61	0.30	0.63	0.22
DRCAML	T100SG	512	0.65	0.43	0.61	0.32	0.63	0.29
DRCAML	T100SG	1,250	0.65	0.42	0.61	0.32	0.63	0.29
DRCAML	T300SG	1,250	0.68	0.41	0.61	0.30	0.63	0.23
BiGRU	T100SG	512	0.67	0.43	0.58	0.35	0.62	0.27
BiGRU	T100SG	1,250	0.67	0.43	0.59	0.35	0.62	0.27
BiGRU	T300SG	1,250	0.65	0.41	0.58	0.35	0.63	0.24
CNNText	T100SG	512	0.68	0.40	0.59	0.36	0.63	0.25
CNNText	T100SG	1,250	0.68	0.40	0.59	0.36	0.63	0.25
CNNText	T300SG	1,250	0.64	0.39	0.58	0.34	0.61	0.24

Table 5.11: Comparison of micro-F1, macro-F1 of case studies among various multi-label neural classifiers with various embeddings and sequence length for eICU data. Bold is used to indicate the best results.

Critical difference plots for T100SG and T300SG for multi-label classifiers

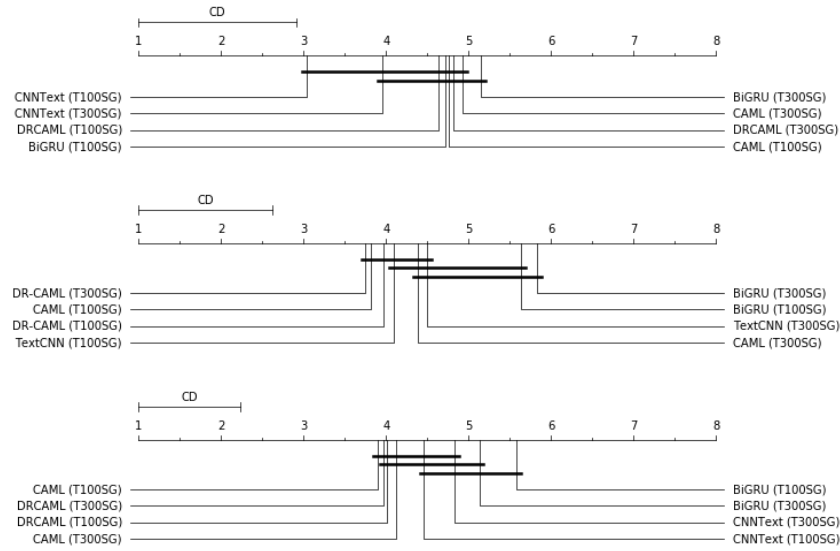


Figure 5.17: Critical difference plots for case studies - cardiovascular disease (top); COVID-19 (middle); and fungal or bacterial infection (bottom), for MIMIC-III data. Nemenyi post-hoc test (95% confidence level), identifying statistical differences between label F1 scores of multi-label neural networks presented in Table 5.10 for MIMIC-III data.

are presented in Figure 5.17 and Figure 5.18 for MIMIC-III and eICU respectively. As indicated before in this chapter, critical difference plots are used as additional supportive statistical analysis. For MIMIC-III, CNNTText (T100SG) is the preferred choice for cardiovascular disease, DRCAML (T300SG) for COVID-19 patient shielding and CAML (T100SG) for systemic fungal or bacterial infection. For eICU, the difference in average ranking is not statistically significant, as there is a bold line connecting across the classifiers.

5.3.1 Cardiovascular Disease

For cardiovascular disease, based on the results presented in Tables 5.10 and 5.11, micro F1 scores of CAML and DRCAML are better than that of other classifiers presented in this research for MIMIC-III and eICU data. This is similar to the observation in predictions of Level 1, 2, and 3 of ICD-9 codes. For eICU data, micro F1 scores of CNNTText are on par with CAML and DRCAML for cardiovascular disease. Macro F1 scores obtained from CNNTText performs

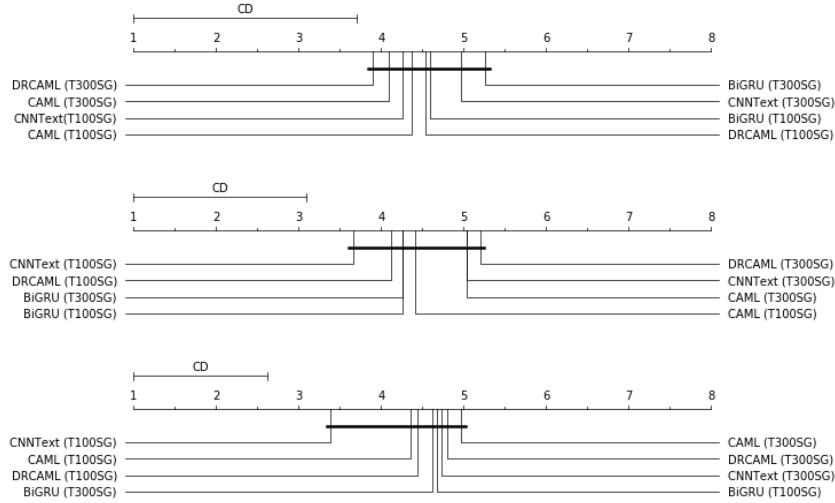


Figure 5.18: Critical difference plots for case studies - cardiovascular disease (top); COVID-19 (middle); and fungal or bacterial infection (bottom), for eICU data. Nemenyi post-hoc test (95% confidence level), identifying statistical differences between label F1 scores of multi-label neural networks presented in Table 5.11 for eICU data.

better for MIMIC-III in comparison to other multi-label classifiers presented in Table 5.10, and BiGRU performs on par with DRCAML for eICU data.

Figure 5.19 presents a comparison between label F1 scores for cardiovascular disease using MIMIC-III and eICU dataset. For MIMIC-III, CNNTText shows a clear improvement in performances, especially in tail-end labels compared to DRCAML.

5.3.2 COVID-19 Patient Shielding

For COVID-19 patient shielding, based on the results presented in Tables 5.10 and 5.11, micro F1 scores of CAML are better for MIMIC-III, and both CAML and DRCAML are better for eICU data in comparison, as noticed in other multi-label problems presented in this research. For macro F1 scores, CNNTText performs better for both MIMIC-III and eICU in comparison to other multi-label classifiers presented in Table 5.10.

Figure 5.20 presents a comparison between label F1 scores for COVID-19 patient shielding using MIMIC-III and eICU dataset. Each plot includes a

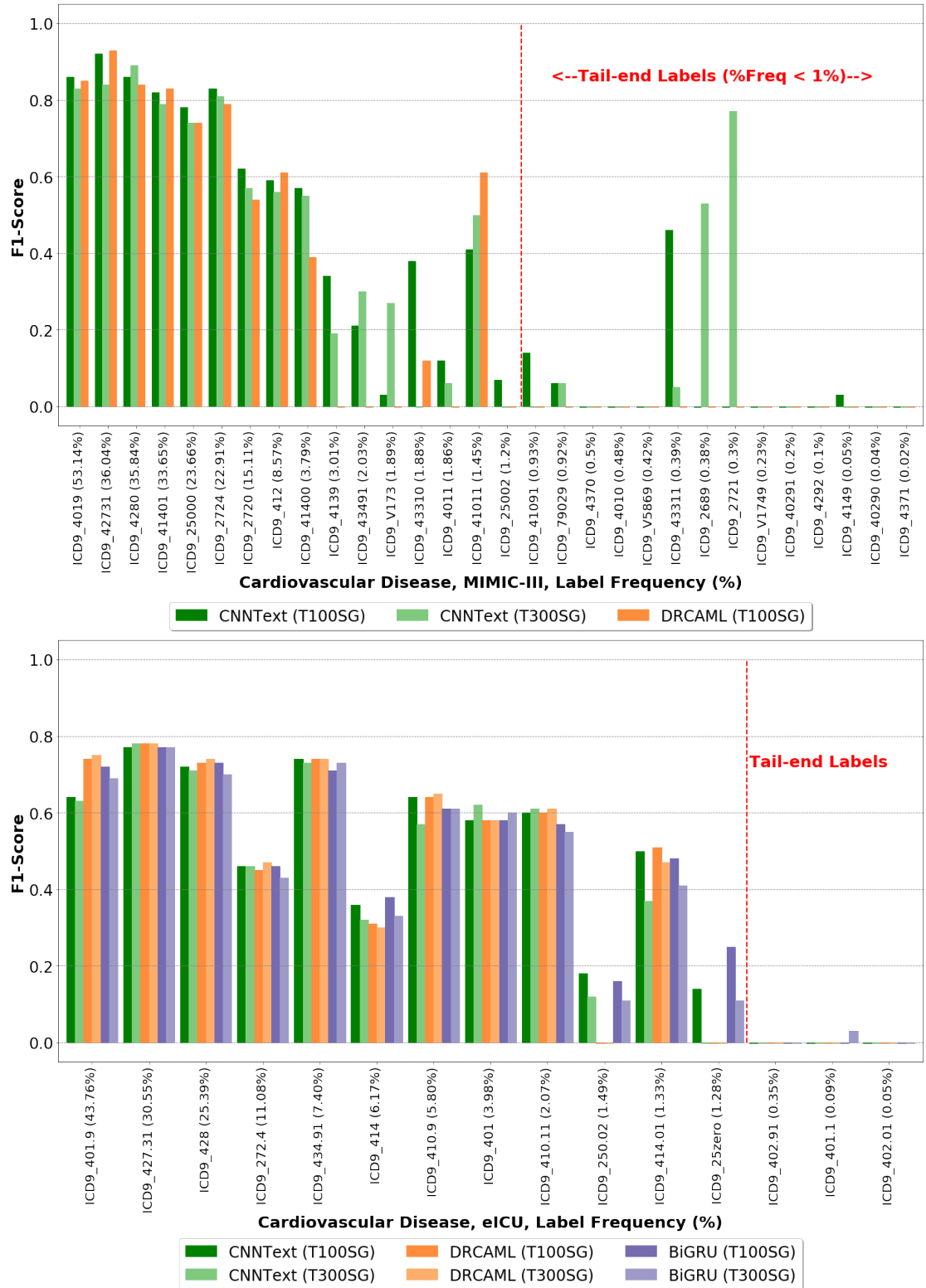


Figure 5.19: Comparison of F1 scores for cardiovascular disease with MIMIC-III (top) and eICU (bottom). The tail end labels are indicated. Each plot presents with individual legends for references.

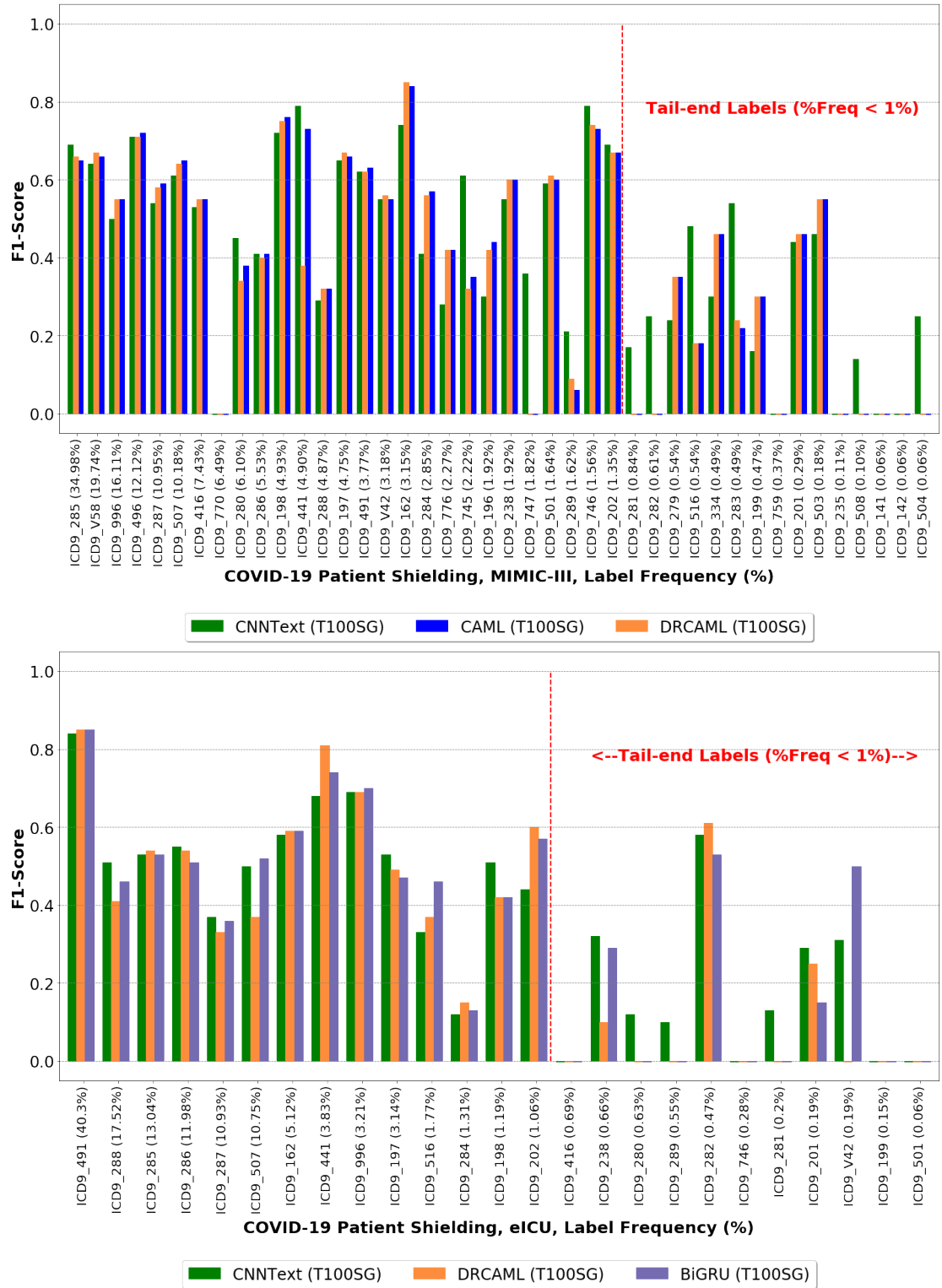


Figure 5.20: Comparison of F1 scores for COVID-19 patient shielding with MIMIC-III (top) and eICU (bottom). The tail end labels are indicated. Each plot presents with individual legends for references.

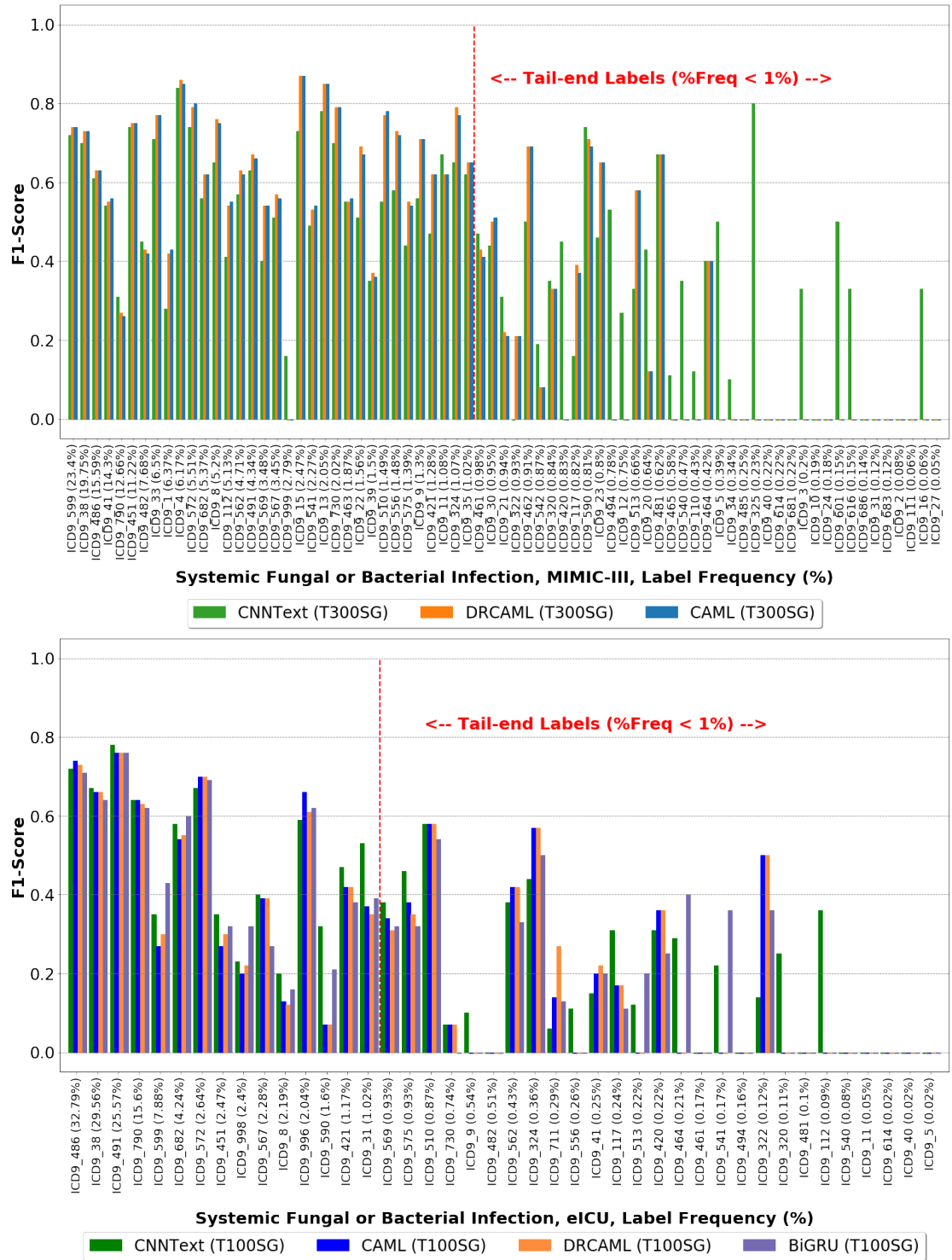


Figure 5.21: Comparison of F1 scores for systemic fungal or bacterial infection with MIMIC-III (top) and eICU (bottom). The tail end labels are indicated. Each plot presents with individual legends for references.

legend to indicate the classifier and embeddings used. Tail-end labels are also indicated. COVID-19 presents a similar trend to cardiovascular disease, where for labels with high frequency, all networks are on par, but for tail-end labels, CNNTText performs better than CAML or DRCAML for MIMIC-III data.

5.3.3 Systemic Fungal or Bacterial Infection

For fungal or bacterial infection, Tables 5.10 and 5.11, shows micro F1 scores of CAML, DRCAML and BiGRU are the same for MIMIC-III, and CAML is better for eICU data. Macro F1 scores of CAML and CNNTText are better for MIMIC-III, and DRCAML for eICU in comparison to other multi-label classifiers presented in Table 5.10.

Figure 5.21 presents a comparison between label F1 scores for fungal or bacterial infection using MIMIC-III and eICU dataset. Each plot includes a legend to indicate the classifier and embeddings used. Tail-end labels are also indicated. As observed in the COVID-19 patient shielding case, the trend of F1 scores is similar where for labels with high frequency, all networks are on par, but for tail end labels, CNNTText performs better for datasets.

5.4 Discussion

This chapter presented experimental outcomes of NLP techniques used for a multi-label medical text classification tasks. Predicting medical codes for patients with multi-morbidity for ICD-9 levels 1, 2 and 3 for both MIMIC-III and eICU are used as supervised multi-label classification tasks. Label-set sizes include 18, 93, 158, 316 and 923. Predicting medical codes related to case studies are also presented for both MIMIC-III and eICU data. Results and analysis are done at both individual label level and also at overall performances of multi-label classifiers. It is evident that variations in embeddings such as the skip-gram model over CBOW do result in improvements in F1 scores. In general, as observed in Chapter 4, health-related pre-trained embeddings do

achieve better results than general text pre-trained embeddings.

Neural networks provide better accuracy than traditional machine learning classifiers. The input features of classifiers such as ECC are static, whereas neural networks are dynamic. CAML and DRCAML and embeddings T100SG and T300SG perform better across all levels of ICD-9 codes. For case studies using MIMIC-III data, CNNTText provides an improvement in F1 scores, especially for tail-end labels. The general poor performances across the multi-label classifiers are evident for tail-end labels. This thesis presents a more detailed study and potential techniques to improve such poor tail-end label performances in Chapter 7.

Results presented in this chapter show that there is a definite benefit in exploring various techniques for representing the text features. However, not all techniques result in improvements of overall accuracy of predictions, i.e., POS or text split tags did not show consistent improvements across labels. An analysis of text representations using transformers for multi-label is presented in the next chapter, which presents more experimental evaluations of language models for multi-label medical text classifications.

Chapter 6

Transformers for Multi-label Classification of Medical Text

There has been a significant advance in NLP in the last couple of years, especially with transformer models. With minimum effort, transfer learning of pre-trained models by fine-tuning on downstream supervised tasks achieves very good results [16, 126]. This chapter fills the gap in the use of transformers in multi-label medical domain-specific tasks for highly imbalanced datasets. Predicting medical codes is used as one of the down-streaming tasks, where levels 1, 2 and 3 of ICD-9 codes are considered individual flat multi-label problems. The three case studies introduced in Section 2.8 are also used as down-streaming tasks. Both eICU and MIMIC-III data are used for experiments, and selected experimental results for embedding based neural networks from Chapter 5 are presented for direct comparison. Experimental results for multiple multi-label problems are presented to gain a better understanding of the effectiveness of transformers. Critical difference plots are presented as supportive statistical analysis. This chapter:

- analyses the effectiveness of using transformers for the task of automatically predicting medical codes of levels 1, 2, and 3 of ICD-9 codes, and predicting medical codes related to case studies from EHRs for multiple document lengths and several labels.

- demonstrates that for documents with sequence length truncated at 512 tokens, medical domain-specific transformer models outperform state-of-the-art (SOTA) methods for multi-label problems with 18, 93 and 158 labels for both datasets.
- demonstrates that for longer documents and larger multi-label problems, standard transformer models', with restricted text sequence length (i.e. 512 tokens), F1 scores are not as good as the traditional word-embeddings-based SOTA networks.
- demonstrates that transformer models such as TransformerXL with longer input sequences outperform SOTA networks.
- as with Chapter 5, tail-end labels were explicitly considered, where F1 scores of zero were observed across many labels.

6.1 ICD-9 Level 1

This section presents detailed experimental results for level 1 ICD-9 codes for both eICU and MIMIC-III data. First, a summary of text input sequence selection is presented, followed by a comparison of transformer models. An outline of the time required for each epoch using various transformers and traditional embeddings-based networks are also presented.

6.1.1 Medical Text Input Sequence Selections

MIMIC-III contains long documents with an average of more than 1,500 tokens. Standard transformer models can only handle a maximum sequence length of 512 tokens. Generally, if the sequence is long, it will be truncated to the first 512 tokens. Schäfer and Friedrich (2020) [15] observed that compared to the first 512 tokens (head-only), tokens from 'Brief Hospital Course' is better in discharge summary for predicting most frequent medical codes. Taking this into consideration, discharge summaries of MIMIC-III data are rearranged

Model	Tokens	Sequence	Micro-F1	Macro-F1
BERT-base	128	head-only	0.74	0.66
ClinicalBERT	128	head-only	0.77	0.68
BioMed-RoBERTa-base	128	head-only	0.77	0.66
PubMedBERT	128	head-only	0.77	0.68
BERT-base	512	head-only	0.79	0.71
ClinicalBERT	512	head-only	0.80	0.72
BioMed-RoBERTa-base	512	head-only	0.80	0.74
PubMedBERT	512	head-only	0.81	0.80
BERT-base	512	all text	0.79	0.70
ClinicalBERT	512	all text	0.80	0.72
BioMed-RoBERTa-base	512	all text	0.79	0.72
PubMedBERT	512	all text	0.81	0.80

Table 6.1: Comparison of overall performance of top level ICD-9 groupings among various multi-label transformer models where input text sequence tokens are varied. Tokens were counted from the start of the pre-processed discharge summary (head-only), or text rearranged using the seven logical sections as mentioned in Section 5.1.3 to ‘Brief Hospital Course’, ‘Discharge Diagnosis’, ‘Past Medical History’, ‘Admission Date’ and the rest (all text). Bold is used to indicate the best overall measure.

using the seven logical sections as mentioned in Section 5.1.3 to ‘Brief Hospital Course’, ‘Discharge Diagnosis’, ‘Past Medical History’, ‘Admission Date’ and the rest (referred to as **all text** in Table 6.1). Table 6.1 presents a comparison of level 1 ICD-9 codes for MIMIC-III data with various sequence length and segment selections. It shows that there is a clear improvement from 128 tokens to 512 tokens across all transformer models. However, there is no significant difference between **head-only** and **all text**. Hence, this thesis uses **head-only** for all experiments.

6.1.2 Comparison between Transformers

Table 6.2 considers the overall performance of top-level ICD-9 groups for several transformer models with various sequence length for MIMIC-III data. For comparison, results for DRCAML and CAML with T100SG are also presented. Experiments use a train-test hold-out set for validation, where data is split into 80/20. The results presented are obtained by averaging three independent runs.

Across standard transformer models with a sequence length of 512, PubMedBERT outperforms other models, with micro and macro F1 scores of MeDAL-Electra being the worst among those presented. TransformerXL (3,072) outperforms all other models. This is supported by critical difference plots presented in Figure 6.1, where TransformerXL (3,072) has the lowest rank, and MeDAL-Electra the highest rank with no bold line connecting the two, indicating that difference in average ranking is statistically significant. Compared to Longformer (512) and TransformerXL (512), micro and macro F1 scores of Longformer (3,000) and TransformerXL (3,072) show an improvement of 5 - 10%. Moreover, no bold lines are connecting Longformer (512) with Longformer (3,000) and TransformerXL (512) with TransformerXL (3,072) in the CD-plot presented in Figure 6.1, indicating that the difference in average ranking is statistically significant. Macro-F1 scores of SOTA models DRCAML and

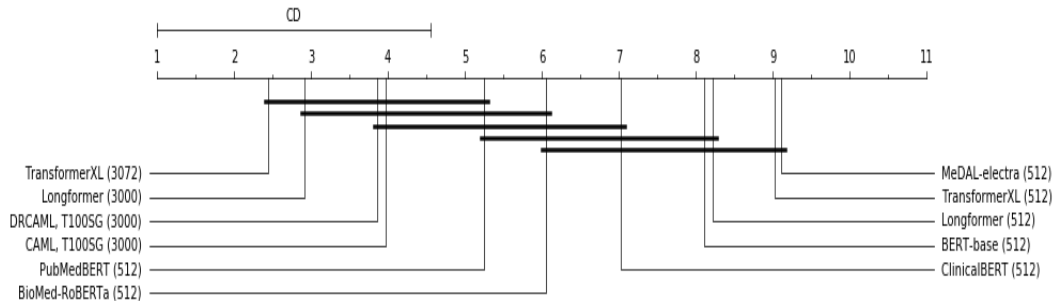


Figure 6.1: Critical difference plots for MIMIC-III, top level ICD-9 groupings. Nemenyi post-hoc test (95% confidence level), identifying statistical differences between multi-label transformers and embeddings-based neural networks presented in Table 6.2 for MIMIC-III data.

Level 1 ICD-9 codes, 18 labels, MIMIC III						
Models	Tokens	Micro -F1	Macro-F1	Hamming Loss	Rank loss	Jaccard Similarity
BERT-base	512	0.792	0.714	0.157	0.313	0.665
ClinicalBERT	512	0.796	0.725	0.157	0.308	0.670
PubMedBERT	512	0.807	0.746	0.148	0.296	0.685
BioMed-RoBERTa	512	0.798	0.736	0.154	0.305	0.673
MeDAL-electra	512	0.782	0.712	0.168	0.327	0.650
Longformer	512	0.796	0.733	0.155	0.310	0.666
TransformerXL	512	0.788	0.723	0.166	0.327	0.648
Longformer	3,000	0.837	0.795	0.123	0.249	0.725
TransformerXL	3,072	0.846	0.801	0.118	0.238	0.739
DRCAML, T100SG	3,000	0.843	0.733	0.145	0.270	0.722
CAML, T100SG	3,000	0.842	0.731	0.146	0.271	0.721

Table 6.2: Comparison of overall performance of top level ICD-9 groupings among various transformers and sequence length for MIMIC-III. CAML and DRCAML with T100SG embeddings are presented for comparison. Bold is used to indicate the best results for each token length group for transformers and underline is used to indicate best overall measure. For micro-F1, macro-F1 and Jaccard similarity, higher values are better; for Hamming loss and rank loss, lower values are better.

CAML are worse than PubMedBERT, BioMed-RoBERTa, Longformer (3,000) and TransformerXL (3,072), with almost 10% improvement observed between CAML/DRCAML and TransformerXL (3,072).

Figure 6.2 presents a comparison of F1 scores for level 1 ICD-9 codes for various transformer models and maximum sequence lengths for MIMIC-III data. Across transformers with maximum sequence length of 512, F1 scores of PubMedBERT are better or on par with other transformers except for categories **preg**, **pren** and **bld**. In contrast, compared to Longformer (3,000) and TransformerXL (3,072), where longer sequence text is used, F1 scores of PubMedBERT is either the same or worse. Generally, TransformerXL (3,072) outperforms other models, with the exception of categories **skin** and **preg** where F1 score of Longformer (3,000) is better.

Table 6.3 considers overall performance of top-level ICD-9 groups for several transformer models with various sequence length for eICU data. For com-

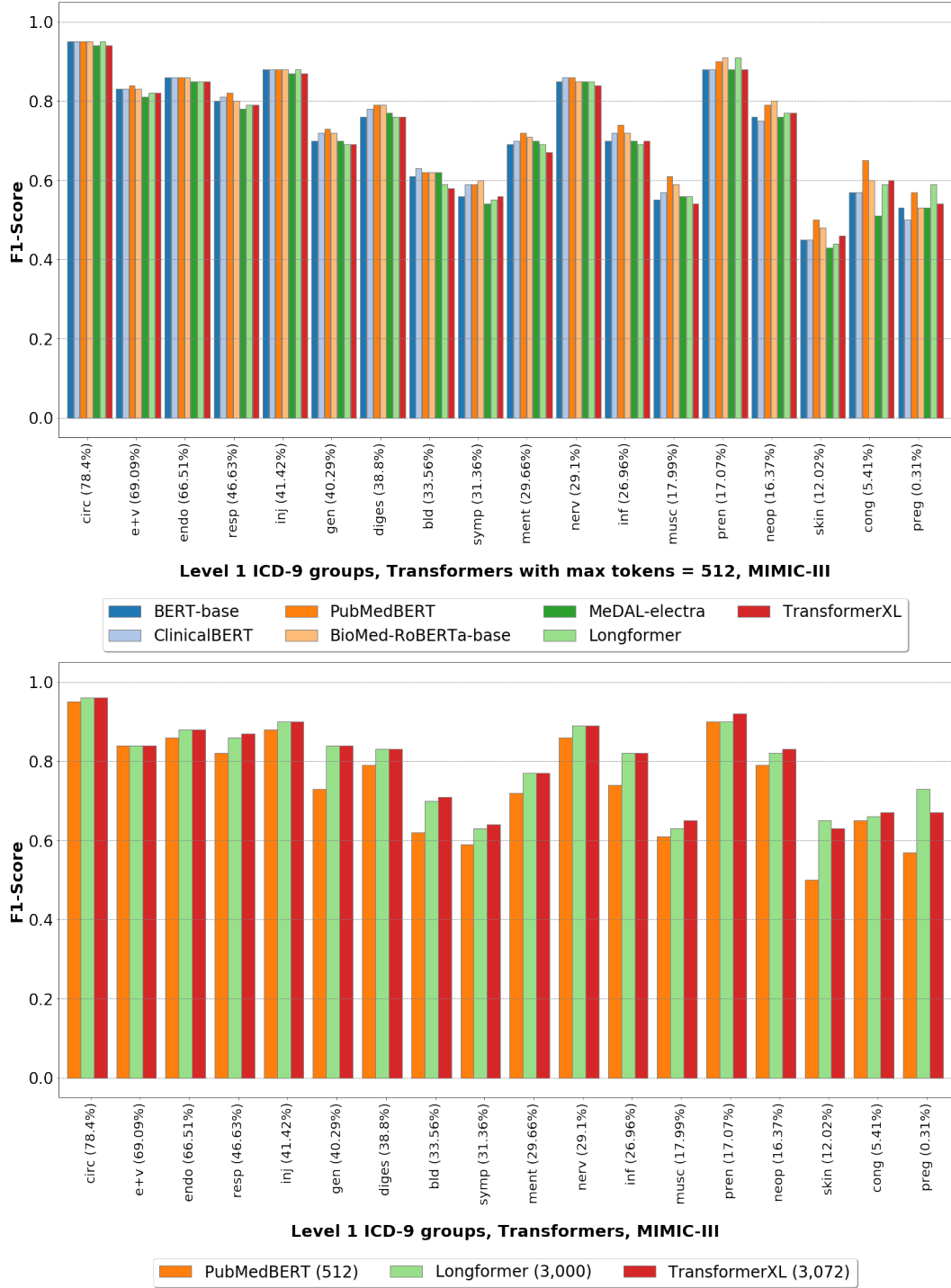


Figure 6.2: Comparison of individual F1 scores of top level ICD-9 groups with varied transformer models with max sequence length of 512 (top) and varied sequence lengths (bottom) for MIMIC-III data. Most frequent label on the left to the least frequent on the right.

parison, results for DRCAML and CAML with T100SG are also presented. Experiments use a train-test hold-out set for validation, where data is split into 80/20. The results presented are obtained by averaging three indepen-

Level 1 ICD-9 codes, 18 labels, eICU						
Models	Tokens	Micro -F1	Macro-F1	Hamming Loss	Rank loss	Jaccard Similarity
BERT-base	512	0.682	0.510	0.081	0.380	0.605
ClinicalBERT	512	0.685	0.530	0.081	0.342	0.609
PubMedBERT	512	0.682	0.520	0.082	0.347	0.602
BioMed-RoBERTa	512	0.689	0.536	<u>0.080</u>	0.333	0.613
MeDAL-electra	512	0.667	0.505	0.086	0.354	0.590
Longformer	512	0.688	0.533	0.081	0.332	0.614
TransformerXL	512	0.692	0.530	<u>0.080</u>	0.329	<u>0.619</u>
Longformer	1,250	0.690	0.538	0.081	0.331	0.615
TransformerXL	1,250	0.700	0.540	0.080	0.328	0.619
DRCAML, T100SG	1,250	0.647	0.512	0.123	0.268	0.526
CAML, T100SG	1,250	0.645	0.511	0.125	<u>0.267</u>	0.523

Table 6.3: Comparison of overall performance of top level ICD-9 groupings among various transformers and sequence length for eICU. CAML and DRCAML with T100SG embeddings are presented for comparison. Bold is used to indicate the best results for each token length groups for transformers and underline is used to indicate best overall measure. For micro-F1, macro-F1 and Jaccard similarity, higher values are better; for Hamming loss and rank loss lower values are better.

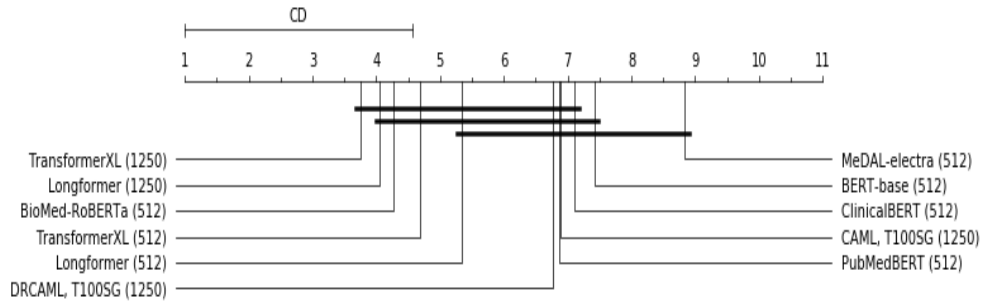


Figure 6.3: Critical difference plots of label F1 scores for eICU, top level ICD-9 groupings. Nemenyi post-hoc test (95% confidence level), identifying statistical differences between multi-label transformers and embeddings-based neural networks presented in Table 6.3 for eICU data.

dent runs.

Across standard transformer models with a sequence length of 512, the macro F1 score of BioMed-RoBERTa outperforms other models, with macro F1 scores of MeDAL-Electra being the worst among those presented. Trans-

formerXL (1,250) outperforms all other models. This is supported by critical difference plots presented in Figure 6.3, where TransformerXL (1,250) has the lowest rank and MeDAL-Electra the highest rank, with no bold line connecting the two, indicating that difference in average ranking is statistically significant. Compared to Longformer (512) and TransformerXL (512), micro and macro F1 scores of Longformer (1,250) and TransformerXL (1,250) show marginal improvements, with no statistically significant differences observed in the CD-plot. Moreover, micro and macro F1 scores of SOTA models DRCAML and CAML are worse than most transformer models, with 5-8% improvements observed between CAML/DRCAML and TransformerXL (1,250). This observation was only evident in macro-F1, not in the micro-F1 score for MIMIC-III.

Figure 6.4 presents comparison of F1 scores for level 1 ICD-9 codes for various transformer models and maximum sequence lengths for eICU data. Across transformers with a maximum sequence length of 512, F1 scores of BioMed-RoBERTa-base are better or on par with other transformers for ten of the 18 categories, with category **cong** being the most prominent. For a maximum sequence length of 1250, F1 scores of TransformerXL (1,250) are better or on par with other classifiers for nine of the 18 categories. F1 score of DRCAML (1,250) is better than the F1 scores of transformers for **pren**, **skin** and **musc**. For infrequent labels **preg** and **cong**, DRCAML performs worse on comparison.

6.1.3 Computational Time Requirements

As indicated through the results in this section, TransformerXL, with a longer sequence limit, 3,072 token for MIMIC-III and 1,250 for eICU, outperforms other transformer models and traditional word embeddings-based SOTA methods. Table 6.1.3 presents an overview of the runtime required per epoch in seconds across a range of models. Traditional embeddings-based neural networks require significantly less time than transformers, with MeDAL-Electra needing the less time per epoch within the transformer models. On the other

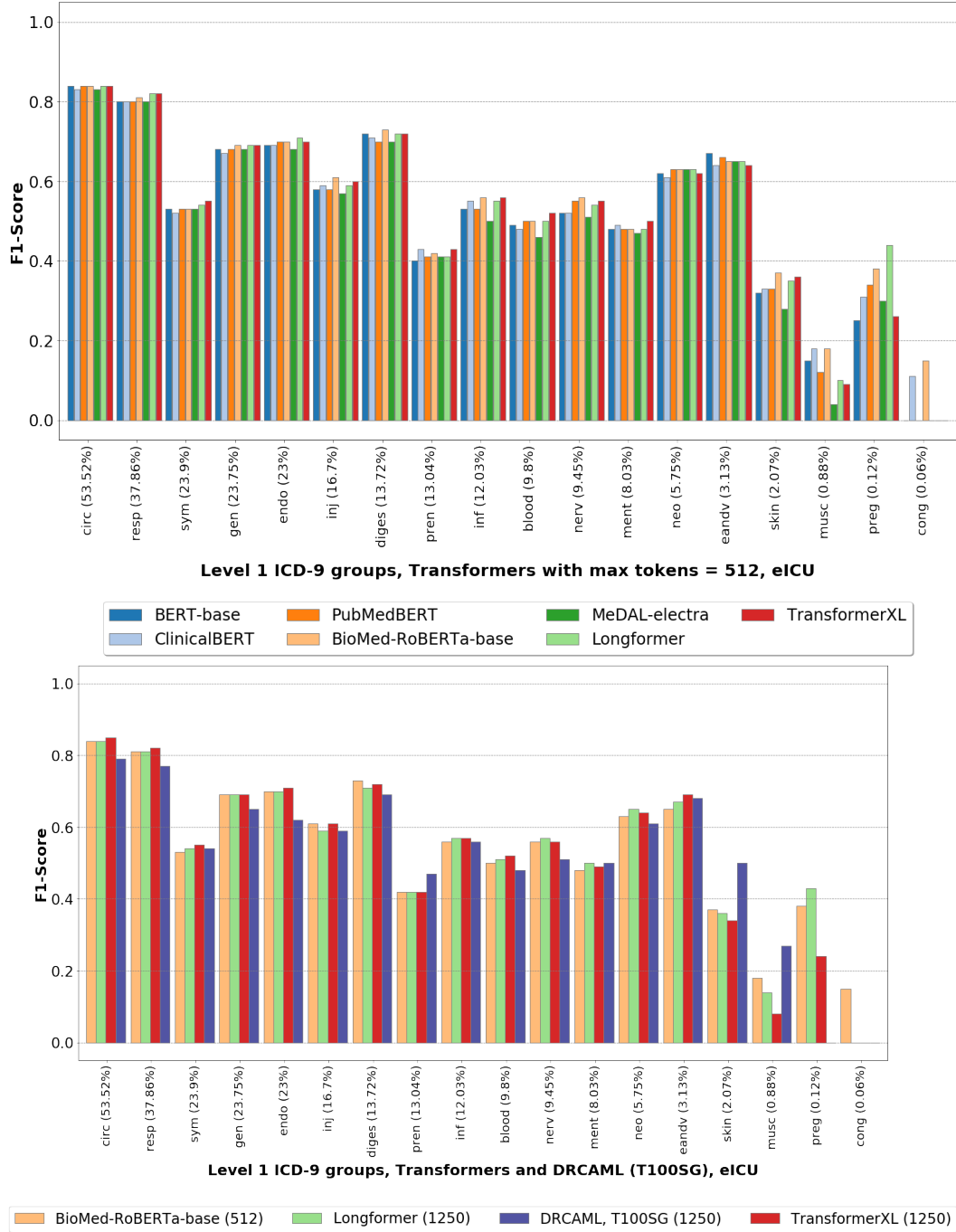


Figure 6.4: Comparison of individual F1 scores of top level ICD-9 groups with varied transformer models with max sequence length of 512 (top) and with varied sequence length and models (bottom) for eICU data. Most frequent label on the left to the least frequent on the right.

hand, TransformerXL (3,072) and TransformerXL (1,250) require considerably more computational time than other models. For example, the computational time required is about 50 times more than BERT-base for MIMIC-III with TransformerXL (3,072), and 17 times more than BERT-base for eICU with

Level 1 ICD-9 codes, 18 labels				
Models	MIMIC-III		eICU	
	Tokens	Time per epoch	Tokens	Time per epoch
MeDAL-electra	512	1,321 sec	512	2,998 sec
PubMedBERT	512	1,475 sec	512	4,024 sec
BERT-base	512	1,502 sec	512	4,918 sec
ClinicalBERT	512	3,280 sec	512	5,879 sec
BioMed-RoBERTa	512	3,221 sec	512	8,827 sec
Longformer	512	7,123 sec	512	21,205 sec
TransformerXL	512	7,844 sec	512	21,791 sec
Longformer	3,000	31,901 sec	1,250	32,238 sec
TransformerXL	3,072	73,655 sec	1,250	83,335 sec
CNNText, T100SG	3,000	42 sec	1,250	52 sec
CAML, T100SG	3,000	45 sec	1,250	57 sec
DRCAML, T100SG	3,000	55 sec	1,250	58 sec
BiGRU, T100SG	3,000	215 sec	1,250	240 sec

Table 6.4: Time required (in seconds) for language models presented in Table 6.2 and Table 6.3 for Level 1 ICD-9 grouping².

TransformerXL (1,250). Compared to TransformerXL, Longformer, with identical/similar maximum sequence length requires less than half the time for long sequences. However, for text truncated to 512 tokens, both Longformer and TransformerXL require very similar time. This behaviour is partly explained by the fact that the Longformer architecture is designed to scale linearly with sequence length, instead of quadratically as in the original transformer models [82].

²Average times (in seconds) based on experiments run on 12 core Intel(R) Xeon(R) W-2133 CPU @ 3.60GHz, GPU device GV100GL [Quadro GV100].

Level 2 ICD-9 codes, 158 labels, MIMIC III						
Models	Tokens	Micro -F1	Macro-F1	Hamming Loss	Rank loss	Jaccard Similarity
BERT-base	512	0.623	0.386	0.051	0.412	0.475
ClinicalBERT	512	0.627	0.387	0.051	0.405	0.477
PubMedBERT	512	0.647	0.424	0.048	0.385	0.502
BioMed-RoBERTa	512	0.635	0.405	0.049	0.402	0.487
MeDAL-electra	512	0.591	0.286	0.050	0.489	0.439
Longformer	512	0.633	0.377	0.048	0.417	0.486
Longformer	3,000	<u>0.716</u>	<u>0.444</u>	<u>0.040</u>	<u>0.317</u>	<u>0.576</u>
CAML, 100SG	3,000	<u>0.716</u>	0.428	0.047	0.303	0.560
DRCAML, 100SG	3,000	<u>0.716</u>	0.415	0.047	0.305	0.559
Level 3 ICD-9 codes, 923 labels, MIMIC III						
ClinicalBERT	512	0.526	0.166	0.012	<u>0.377</u>	0.540
PubMedBERT	512	0.547	0.184	0.012	0.520	0.400
BioMed-RoBERTa	512	0.533	0.175	0.012	0.536	0.386
Longformer	3,000	0.602	0.168	<u>0.010</u>	0.477	0.450
CAML, T100SG	3,000	<u>0.635</u>	<u>0.264</u>	0.012	0.392	<u>0.474</u>
DRCAML, T100SG	3,000	0.633	0.241	0.012	0.403	0.471

Table 6.5: Comparison of the overall performance of Levels 2 and 3 of ICD-9 codes among various transformers and sequence lengths for MIMIC-III. CAML and DRCAML with T100SG embeddings are presented for comparison. Bold is used to indicate the best results for each token length groups for transformers, and underline is used to indicate the best overall measure. For micro-F1, macro-F1 and Jaccard similarity, higher values are better; for Hamming loss and rank loss, lower values are better.

6.2 ICD-9 Levels 2 and 3

In this section, levels 2 and 3 ICD-9 codes are considered flat multi-label problems for MIMIC-III and eICU data. Table 6.5 presents overall comparison for both levels 2 and 3 of ICD-9 codes for MIMIC-III data. Only Longformer is used for a longer sequence of text, as the required resource for TransformerXL is restrictive (see section 7.7 for detailed explanation of computer resource restrictions). Embeddings-based SOTA methods CAML and DRCAML are also presented for direct comparison. For level 2 ICD-9 codes, Longformer (3,000) outperforms other transformer models. Micro-F1 scores of CAML and DRCAML are the same as Longformer (3,000). Critical difference plots

Level 2 ICD-9 codes, 93 labels, eICU						
Models	Tokens	Micro -F1	Macro-F1	Hamming Loss	Rank loss	Jaccard Similarity
BERT-base	512	0.567	0.293	0.026	0.428	0.478
ClinicalBERT	512	0.577	0.322	0.026	0.411	0.487
PubMedBERT	512	0.574	0.303	0.026	0.419	0.490
BioMed-RoBERTa	512	0.578	<u>0.327</u>	0.026	0.409	0.423
MeDAL-electra	512	0.574	0.248	<u>0.023</u>	0.454	0.482
Longformer	512	0.585	0.311	0.024	0.414	0.495
Longformer	1,250	<u>0.589</u>	0.320	0.024	0.407	<u>0.499</u>
DRCAML, T100SG	1,250	0.572	0.320	0.033	<u>0.294</u>	0.434
BiGRU, T100SG	1,250	0.570	0.307	0.031	0.335	0.440
Level 3 ICD-9 codes, 316 labels, eICU						
ClinicalBERT	512	0.504	0.171	<u>0.009</u>	0.508	0.416
PubMedBERT	512	0.502	0.167	<u>0.009</u>	0.515	0.405
BioMed-RoBERTa	512	0.501	0.180	<u>0.009</u>	0.506	0.412
Longformer	1,250	<u>0.508</u>	0.178	<u>0.009</u>	0.506	<u>0.417</u>
CAML, T100SG	1,250	0.504	<u>0.216</u>	0.012	<u>0.362</u>	0.375
BiGRU, T100SG	1,250	0.487	0.215	0.010	0.458	0.391

Table 6.6: Comparison of overall performance of Levels 2 and 3 of ICD-9 codes among various transformers and sequence length for eICU. CAML, DRCAML and BiGRU with T100SG embeddings is presented for comparison. Bold is used to indicate the best results for each token length groups for transformers and underline is used to indicate best overall measure. For micro-F1, macro-F1 and Jaccard similarity, higher values are better; for Hamming loss and rank loss, lower values are better.

presented in Appendix Figure D.1 indicate no statistically significant differences between Longformer, CAML and DRCAML. For level 3 of ICD-9 codes, CAML, T100SG outperforms other models, including Longformer (3,000).

Figure 6.5 presents a comparison of F1 scores for level 2 ICD-9 code labels for MIMIC-III data, where the frequency of occurrence of the labels is ordered from high (left) to low (right) between PubMedBERT and Longformer, and CAML and Longformer. Label frequency less than one percent is indicated. For higher frequencies, all three methods perform similar to each other. However, tail-end labels include multiple labels with F1 scores of zero. Overall for tail-end labels, PubMedBERT performs better than both CAML and Long-

former. One possible reason for this may be because the vocabulary used to train PubMedBERT is health-related, whereas Longformer is a general text pre-trained transformer.

Table 6.6 presents overall comparison for both levels 2 and 3 of ICD-9 codes for eICU data. As with MIMIC-III data, only Longformer is used for a longer sequence of text. Embeddings-based methods CAML, DRCAML and BiGRU, are also presented for direct comparison. For level 2 ICD-9 codes, the micro F1 score of Longformer (1,250) and macro F1 of BioMed-RoBERTa outperforms all other models. The critical difference plots presented in Appendix Figure D.2 indicate BioMed-RoBERTa with the lowest rank, and no statistically significant differences between Longformer, DRCAML, BioMed-RoBERTa and ClinicalBERT. For level 3 of ICD-9 codes, macro-F1 scores of CAML and BiGRU, outperforms macro-F1 scores of other models, including Longformer (1,250), by almost 20%. The micro-F1 score of Longformer (1,250) is marginally ($\approx 1\%$) better than that of CAML.

Figure 6.6 presents a comparison of F1 scores for level 2 ICD-9 code labels for eICU data, where the frequency of occurrence of the labels is ordered from high (left) to low (right) between BioMed-RoBERTa, Longformer, and DRCAML. Label frequency less than one percent is indicated. For higher frequencies, all three methods perform similar to each other. However, tail-end labels include multiple labels with F1 scores of zero. A closer look at these labels is presented in Chapter 7.

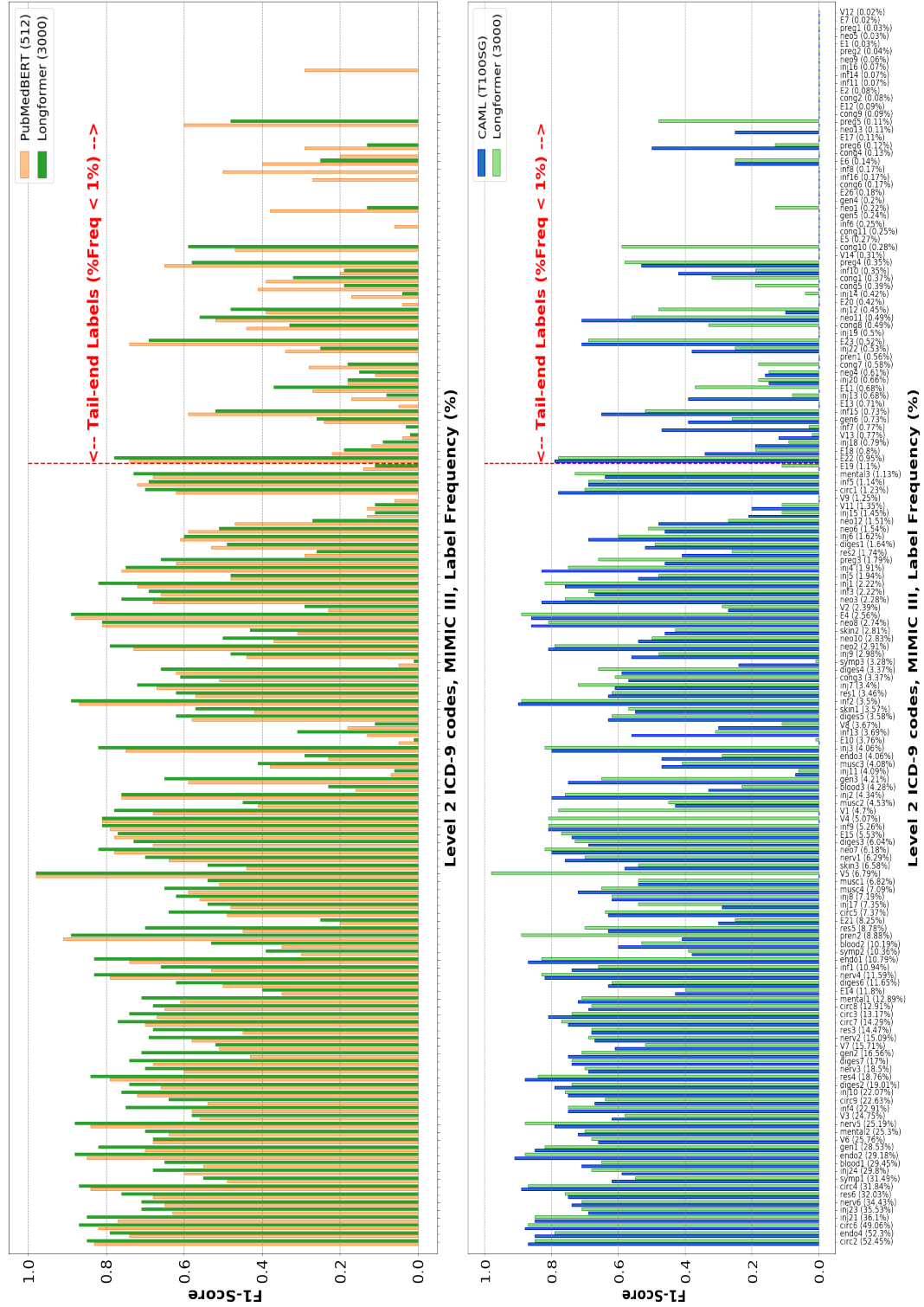
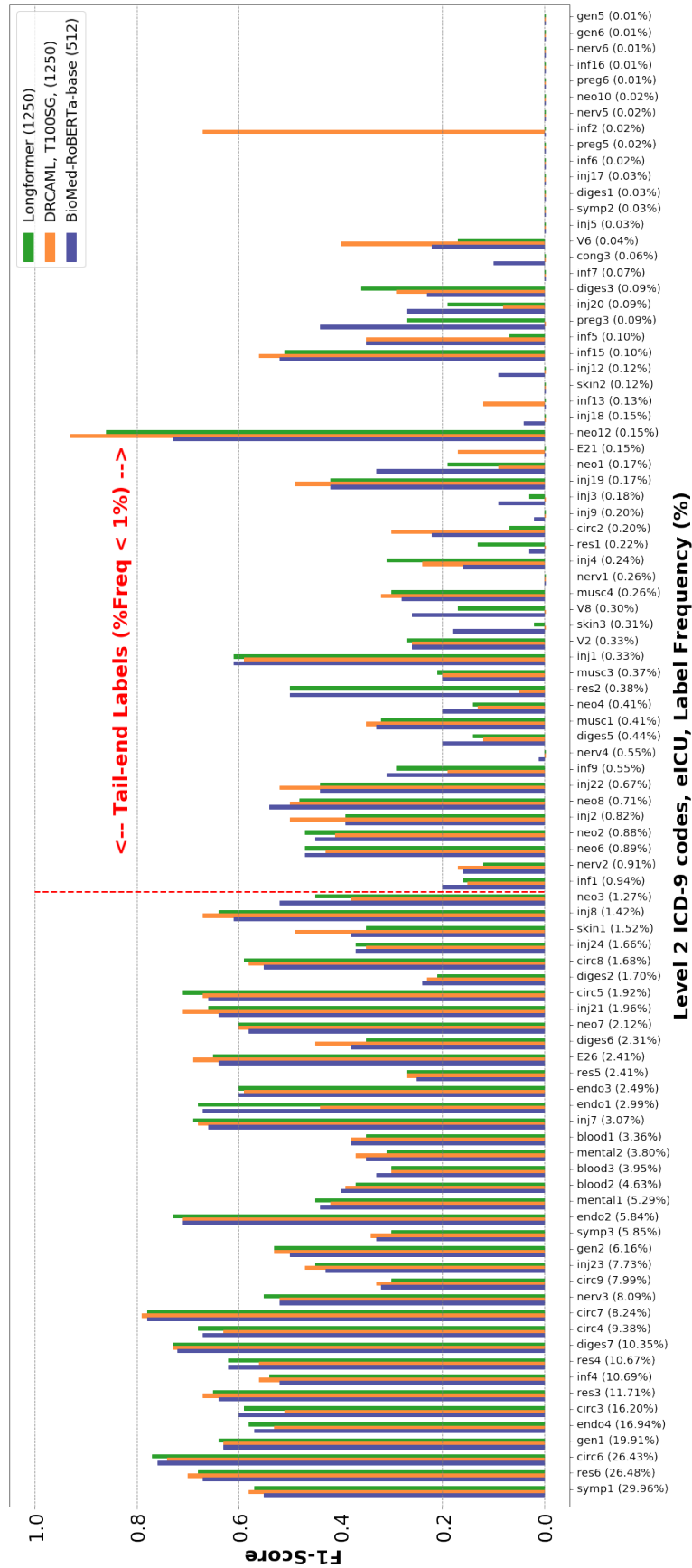


Figure 6.5: Comparison of F1 scores of level 2 ICD-9 codes for MIMIC-III data between PubMedBERT and Longformer (top) and Longformer and CAML (T100SG) (bottom).



6.3 Case Studies

This section considers transformer models for case studies presented in Table 2.4 for both MIMIC-III and eICU data. This research considers predictions of medical codes as a flat multi-label problem for codes associated with cardiovascular disease, COVID-19 patient shielding and systemic fungal or bacterial infections. In addition to the number of labels, the number of instances is different for each case study and data. Only a subset of the available text data is used, due to the availability associated with the relevant case study. Critical difference plots are presented in Appendix D.6 and D.7 for MIMIC-III and eICU respectively.

Table 6.7 presents micro and macro F1 scores for the three case studies using MIMIC-III data, for various different transformer models, and for different limits on the maximum input length. Evidently, across all three case studies, TransformerXL (3,072) outperformed all other models.

Table 6.8 presents micro and macro F1 scores for the three case studies using eICU data. An increase in maximum token sizes of the input sequence has less influence on the overall performance, as observed with other multi-label problems using eICU. This behaviour is explained by the average sequence

Classifier	Tokens	Cardiovascular		COVID-19		Fungal or Bacterial	
		Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
BioMed-RoBERTa	512	0.69	0.30	0.53	0.45	0.45	0.39
PubMedBERT	512	0.70	0.30	0.54	0.48	0.48	0.39
TransformerXL	512	0.68	0.28	0.53	0.45	0.47	0.39
Longformer	512	0.69	0.29	0.52	0.45	0.45	0.36
TransformerXL	1,536	0.73	0.32	0.58	0.49	0.55	0.44
TransformerXL	3,072	0.77	0.32	0.65	0.51	0.64	0.46
Longformer	3,000	0.74	0.30	0.58	0.50	0.58	0.43
CAML (T100SG)	3,000	0.77	0.24	0.61	0.40	0.57	0.40
CNNText (T100SG)	3,000	0.74	0.30	0.58	0.42	0.57	0.32

Table 6.7: Comparison of micro-F1, macro-F1 of case studies among various transformer models and sequence length for MIMIC-III data. Bold is used to indicate the best results.

Classifier	Tokens	Cardiovascular		COVID-19		Fungal or Bacterial	
		Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
BERT-base	512	0.67	0.40	0.61	0.37	0.62	0.28
BioMed-RoBERTa	512	0.66	0.39	0.61	0.37	0.62	0.30
PubMedBERT	512	0.67	0.41	0.64	0.39	0.63	0.30
Longformer	512	0.67	0.39	0.61	0.38	0.62	0.29
Longformer	1,250	0.69	0.42	0.61	0.40	0.62	0.29
DRCAML (T100SG)	1,250	0.65	0.42	0.61	0.32	0.63	0.29
BiGRU (T100SG)	1,250	0.67	0.43	0.59	0.35	0.62	0.27

Table 6.8: Comparison of micro-F1, macro-F1 of case studies among various transformer models and sequence length for eICU data. Bold is used to indicate the best results.

length of eICU data, which is less than 512 tokens. There is no clear overall winner among the models presented across the three case studies. This observation is supported by the critical difference plots presented in Appendix Figure D.7, where no statistically significant differences were found among the models presented here across all three case studies.

6.3.1 Cardiovascular Disease

Figure 6.7 presents a comparison between label F1 scores for cardiovascular disease using MIMIC-III and eICU datasets. Each plot includes a legend to indicate the different models used. For MIMIC-III, both CNNTxt and TransformerXL (3,072) show better F1 scores for most labels, with PubMedBERT performing better than Longformer at the tail-end labels. For eICU, with a total of 15 labels and only three tail-end labels, as the frequency of occurrences decreases, F1 scores of BiGRU are better than the transformer model scores, except for label ICD9_402.91.

6.3.2 COVID-19 Patient Shielding

Figure 6.8 presents a comparison between label F1 scores for COVID-19 patient shielding using MIMIC-III and eICU dataset. For MIMIC-III data, Trans-

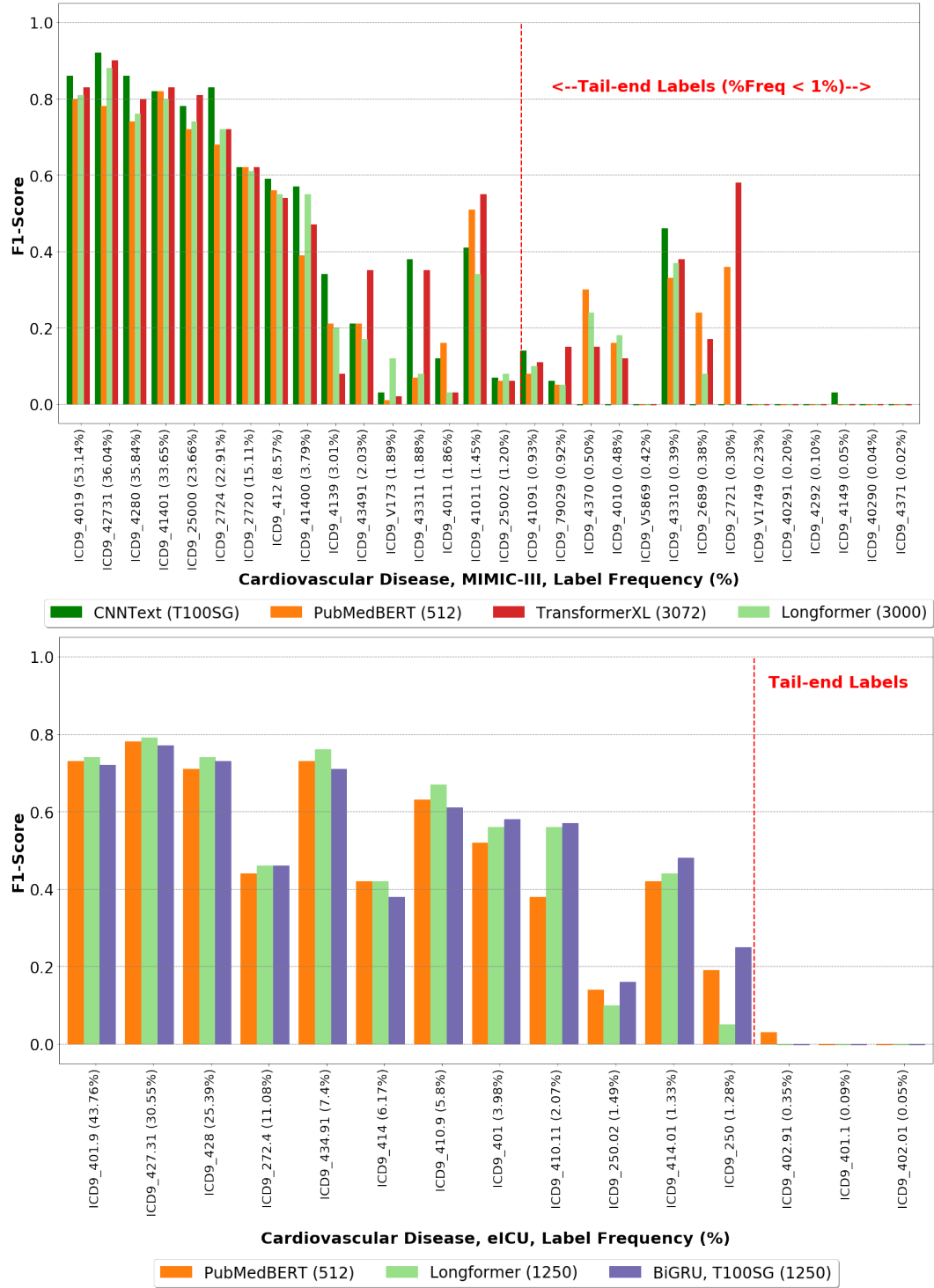


Figure 6.7: Comparison of F1 scores for cardiovascular disease with MIMIC-III (top) and eICU (bottom) using transformers and embeddings-based neural networks. The tail end labels are indicated.

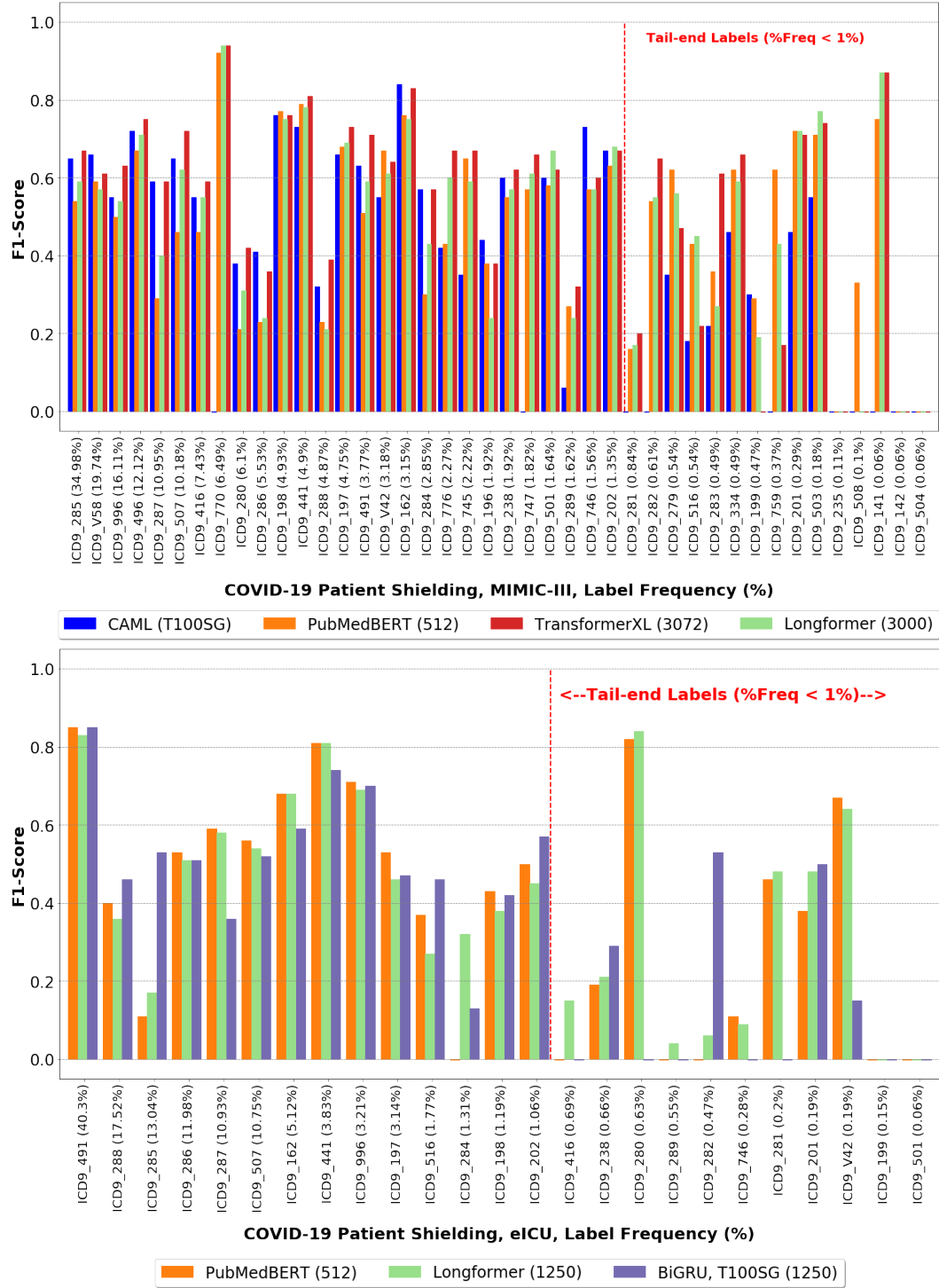


Figure 6.8: Comparison of F1 scores for COVID-19 patient shielding with MIMIC-III (top) and eICU (bottom) using transformers and embeddings-based neural networks. The tail end labels are indicated.

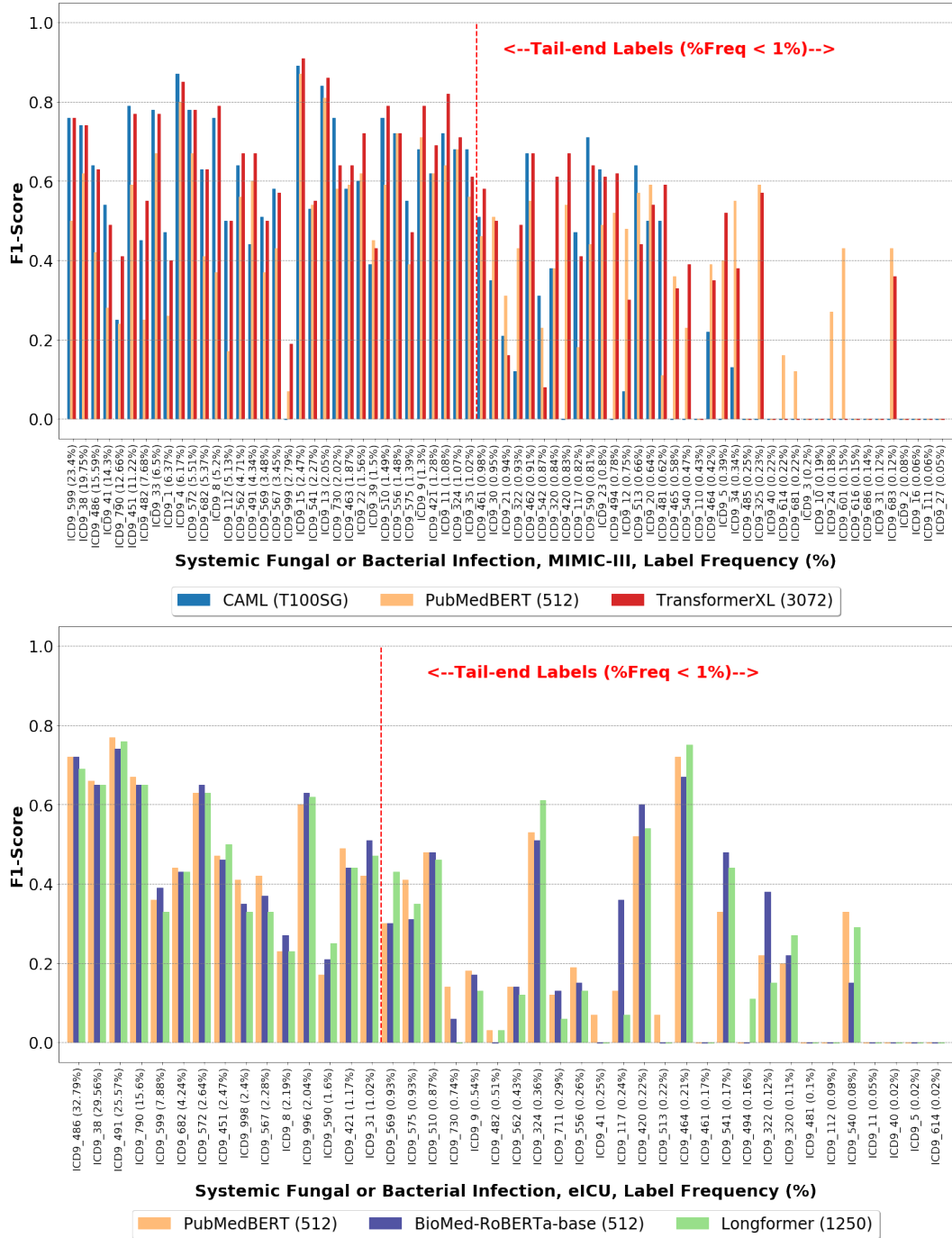


Figure 6.9: Comparison of F1 scores for systemic fungal or bacterial with MIMIC-III (top) and eICU (bottom) using transformers and embeddings-based neural networks. The tail end labels are indicated.

formerXL (3,072) shows better F1 scores than other models across all labels for COVID-19 patient shielding. F1 scores of CAML for labels with less frequency of occurrence are being the worst among those presented. For eICU, Longformer (1,250) shows improvements for tail-end labels.

6.3.3 Systemic Fungal or Bacterial Infection

Figure 6.9 presents a comparison between label F1 scores for systemic fungal or bacterial infection using MIMIC-III and eICU dataset. For both datasets, there are more extended tail-end labels. For MIMIC-III, F1 scores of TransformerXL (3,072) are better for higher frequent labels. However, for lower frequent labels, F1 scores of PubMedBERT are better than both CAML and TransformerXL. For eICU data, F1 scores of all three methods compared are similar across the labels.

6.4 Discussion

This chapter presented a study analysing the effectiveness of transformers for several multi-label problems. It has been shown that using transformers, especially domain-specific pre-trained models can be highly beneficial in multi-label medical text classifications. This chapter has presented new SOTA results for predicting level 1 and level 2 ICD-9 codes from EHRs using a fixed sequence length. However, some issues with using transformers have also been pointed out throughout this thesis. For standard transformers, the input sequence length cannot exceed 512 tokens, impacting the accuracy of predictions, especially for MIMIC-III data. The results presented in this chapter show that for longer documents, such as the MIMIC-III data and multi-label problems with more labels, standard transformer models with restricted text sequence length (i.e. 512 tokens) are not as good as the traditional word-embeddings-based SOTA networks.

As a solution to this issue with input token limitations, TransformerXL

and Longformer were considered. Both Longformer and TransformerXL can be beneficial for long medical documents. There is an increase in performance compared to standard transformer models, which can only handle sequences of at most 512 tokens. TransformerXL (3,072) outperformed all other networks compared in this chapter. However, the resources required for TransformerXL (3,072) present a new limitation. For example, for level 1 ICD-9 codes with 18 labels, TransformerXL (3,072) requires almost 50 times more computational time per epoch than the standard BERT-base for MIMIC-III data. If processing time is not an issue, then TransformerXL with a more extended sequence will be the best model to use. It will also be interesting to try continuously training both Longformer and TransformerXL with health-related text.

As with Chapter 5, this chapter also highlighted the F1 scores of the tail-end labels. Chapter 7 presents a more detailed study of tail-end labels, where it is shown that concatenating transformers improves the F1 scores of tail-end labels.

Chapter 7

Language Model Variations

Recent developments in NLP, especially transformers' success, motivated this research to explore the opportunity to further improve performance for medical-domain specific tasks by exploiting models pre-trained on health data. Chapters 5 and 6 presented an extensive study on various domain-specific language models for various multi-label medical text classification problems. However, most transformer models are limited to a maximum sequence length of 512 tokens, with some exceptions, such as Longformer and TransformerXL, which can process long documents. For a dataset such as MIMIC-III, where the average number of tokens of individual documents are greater than 512, results presented in Chapters 5 and 6 show longer sequences improves predictive accuracy of models. Although Longformer and TransformerXL perform well compared to other language models for long documents, these models require considerable resources such as GPU, CPU, memory and process time in comparison to other models. Moreover, results presented in Chapter 6 show that for larger multi-label problems and infrequent labels, transformer models' F1 scores are not as good as the traditional word-embeddings-based SOTA neural networks.

This chapter considers the option of using multiple variations of language models for multi-label classification with extended text and multi-sources of texts while using fewer resources in comparison to TransformerXL and Long-

former. This chapter presents various options where language models are concatenated and compares them with the results presented in Chapters 5 and 6. This chapter also takes a closer look at the infrequent tail-end labels and explores the option of improving performances when variations of language models are considered. Results show improvements to overall micro and macro F1 scores when concatenated BioMed-Transformer models are considered, even for multi-label problems with more than 100 labels. In addition, there is evidence of improvements in F1 scores for tail-end labels.

The maximum sequence length of eICU dataset is very short compared to MIMIC-III. Hence this chapter only focuses on MIMIC-III data. It is essential to point out that some preliminary experiments were conducted using eICU for cardiovascular disease and systemic fungal or bacterial infections. However, as expected, due to the lack of long text sequences, the language model variations did not show any performance improvements. Examples of results for eICU are presented in Appendix E.

Three variations of concatenated language models are presented: multi-CNNText, multi-BioMed-Transformers and CNNText with Transformers. A brief overview of input text data is also provided. Cardiovascular disease is used to present overall results of significant variations of language models, followed by results for systemic fungal or bacterial infection, COVID-19 patient shielding, and levels 2 and 3 of ICD-9 codes. Micro and macro F1 scores for many language models for epochs 1 to 30 are also presented. The improvement of F1 scores in tail-end labels is presented by considering the differences of F1 scores for each label. For comparison, differences in F1-scores of labels with the frequency of occurrence $> 1\%$ is also presented. The primary focus here is to show the differences when compared to Longformer and TransformerXL. Due to the extreme amount of resources required by these methods, our experiments are limited in terms of the use of TransformerXL.

7.1 Multi-BioMed-Transformers

Multi-BioMed-Transformers refer to an architecture where more than one domain-specific transformer model is concatenated together to enable the option of multiple text inputs. Figure 7.1 presents an example architecture of three PubMedBERT models concatenated together. Chalkidis et al. (2020) [149] proposed a similar architecture called HIER-SCI-BERT for processing long MIMIC-III data. HIER-SCI-BERT encoded consecutive segments on texts with a maximum sequence length of 512 using SCI-BERT [149]. Results obtained using HIER-SCI-BERT outperformed SCI-BERT. However, it still performed poorly compared to other networks [149]. One possible reason for poor results is the use of a continuously pre-trained BERT model instead of a BERT model trained from scratch using biomedical related text [149]. For this thesis, PubMedBERT, a BERT model trained from scratch using biomedical related

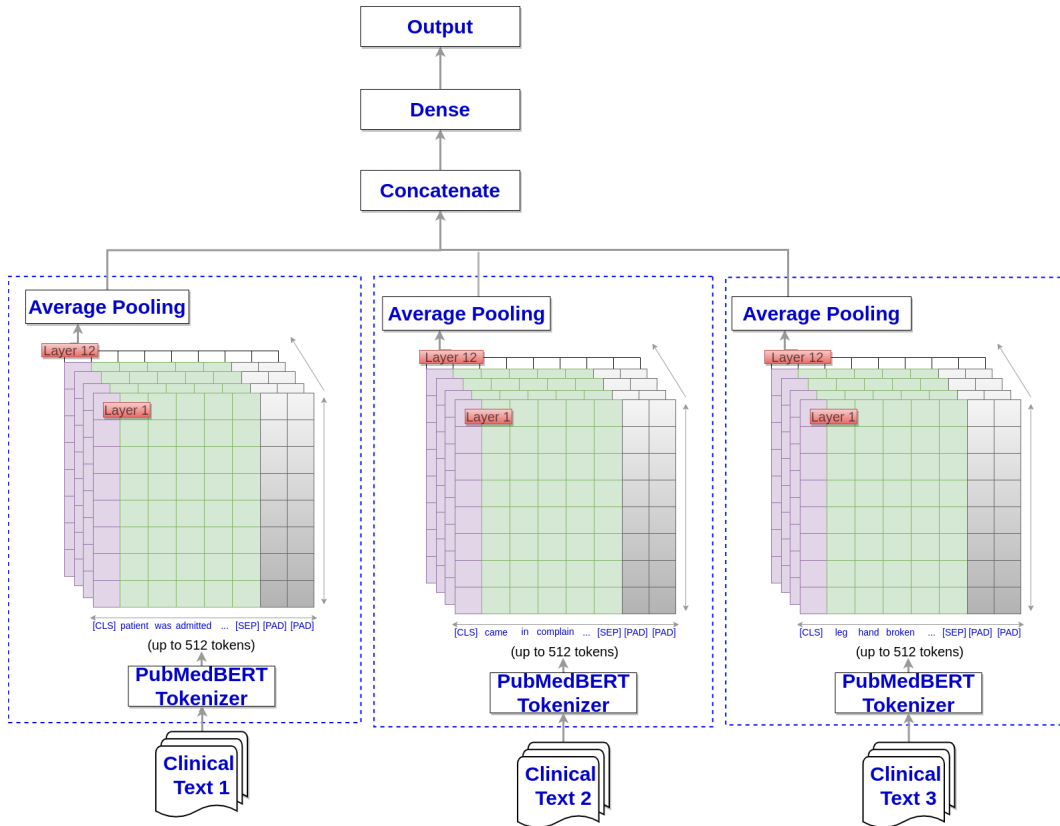


Figure 7.1: TriplePubMedBERT architecture, where each PubMedBERT architecture, as indicated by blue dotted lines, is concatenated together. Three separate input text is indicated by clinical text 1, 2 and 3.

text, is used. Results presented in Chapter 6 also suggests that PubMedBERT performs better than other transformer models with fixed length restrictions.

This research explores the options of two to four PubMedBERT models that are concatenated together. Concatenated transformer models enable the processing of longer input sequences, where the longer input sequence is split into multiple smaller segments with a maximum sequence length of 512 tokens. The average length of discharge summaries in MIMIC-III is approximately 1,500 tokens, hence the choice to concatenate two to four PubMedBERT models. Moreover, as indicated in Chapter 2, MIMIC-III contains text from other categories, such as text summaries of ECG reports and radiology reports. Multi-BioMed-Transformers provides the option to explore using these other available texts as additional input text. More details on input text sequences is provided in Section 7.4.

This research focusses mainly on PubMedBERT, but some experiments also include BioMed-RoBERTa. However, this concept can be expanded using multiple transformer models and different models, including Longformer or TransformerXL. Due to the restrictions of computational resources, this thesis focuses only on the variations mentioned above. This selection of specific transformer models is supported by the results presented in Chapter 6, where PubMedBERT and BioMed-RoBERTa are two models that consistently perform well. Micro and macro F1 scores are better for PubMedBERT and BioMed-RoBERTa than for the other transformer models presented in this research.

7.2 Multi-CNNText

Multi-CNNText adopts the same idea as multi-BioMed-Transformers, where more than one CNNText model is concatenated together. Figure 7.2 presents an example of DualCNNText where two CNNText models are concatenated together. Although CNNText can handle longer sequence length as input text,

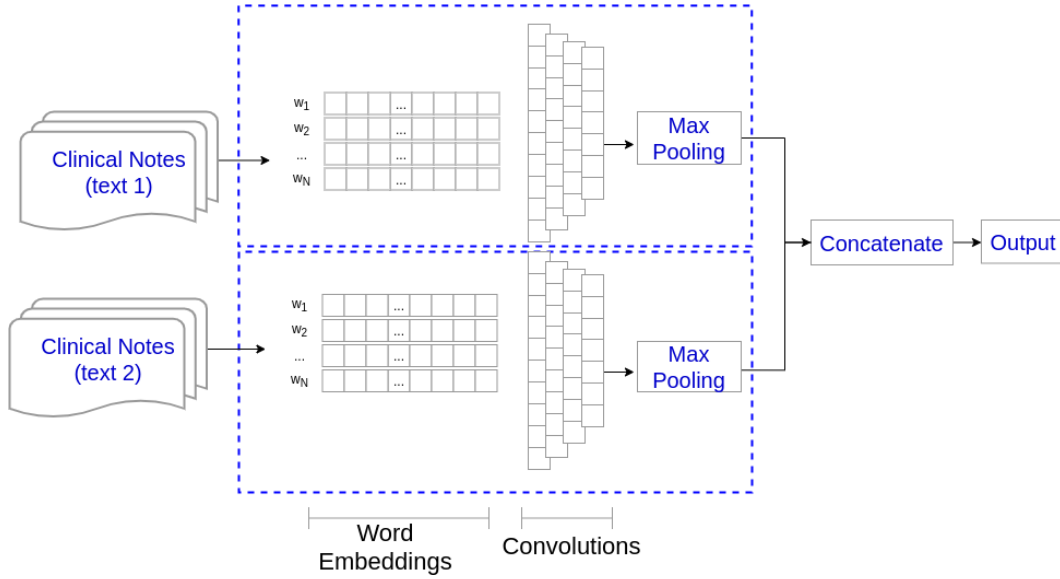


Figure 7.2: DualCNNText architecture, where two identical CNNTText [1], indicated by blue dotted lines, are concatenated together.

concatenating multiple CNNTText models provides the option of using input text from different categories such as ECG and radiology, as mentioned before. As the features of different categories can be captured separately. Embeddings used for CNNTText is a fastText pre-trained model using health-related data with 100-dimensions (T100SG).

7.3 CNNTText with Transformers

The third variation is combining CNNTText with transformers. Although many variations can be considered, this research only considers a couple of variations. BERT-base and PubMedBERT are the two transformers that are used with CNNTText. These choices are because PubMedBERT performs well across multi-label problems as shown in Chapter 6, and BERT-base provides a baseline. However, multiple transformer models can be used and variations to CNNTText, such as with embeddings dimensions. It is also important to point out that CNNTText is just one possible choice, and there are many other deep learning models that can be used instead of CNNTText.

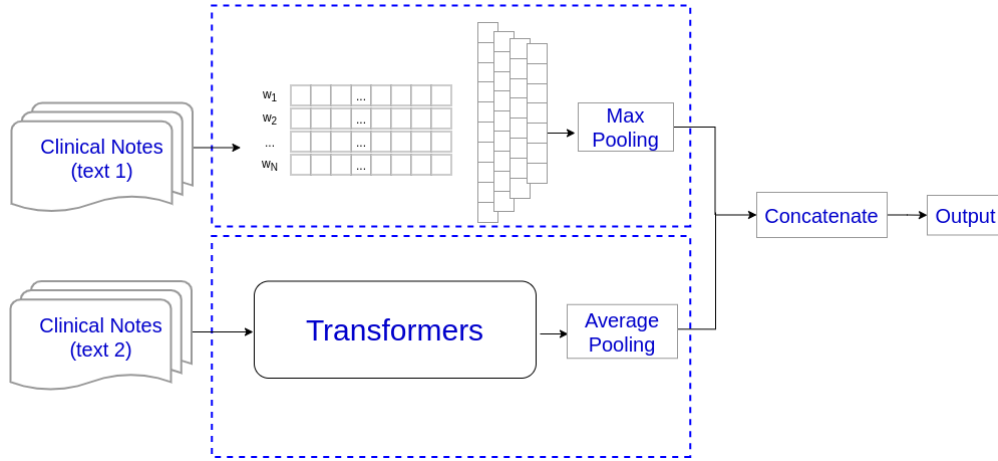


Figure 7.3: CNNTText with Transformer architecture with different input text, where CNNTText [1] and transformer model (see Figures 3.7 and 3.8), indicated by blue dotted lines, are concatenated together.

7.4 Input Text Data

This section presents a brief overview of the variations to the input text used in experiments. Figure 2.2 presented an overview of the frequency of words in MIMIC-III data for categories: discharge summary (**dis**), text summary of ECG reports (**ecg**) and text summary of radiology reports (**rad**). As more than 90% of the patient records in MIMIC-III contain discharge summaries, they have been used as the primary text data for MIMIC-III throughout this research. However, this chapter also explores the option of using **ecg** and **rad**.

As indicated earlier, multiple language models are concatenated to enable the option of using multiple input texts. Discharge summary contains text sequences with an average of 1,500 tokens and a range of 60 to 9,500 tokens. The discharge summary is split into equal segments for a given hospital admission, and each section is labelled text 1, ..., 4. For example, for two splits, if a given discharge summary is 700 tokens long, text 1 is the first 350 tokens, and text 2 is the last 350 tokens. In the case of a lengthy document, if the discharge summary is 2500 tokens long, text 1 is the first 1,250 tokens, and text 2 is the last 1,250 tokens. For multi-BioMed-Transformers where the maximum sequence length is 512, each of text 1, ..., 4 is truncated to 512 tokens. There

are many other ways to split the text, including sequential splits. For instance, with the first example above, text 1 being the first 512 tokens, and text 2 being the remainder 238 tokens. Each of these decisions has some advantages and disadvantages. After preliminary experiments and also based on results presented in [149], the decision was made to split the discharge summary into equal sections.

Another option is using **dis** as text 1, and **ecg** and **rad** as text inputs 2 and 3. This research presents results for the following options:

Option 0: **dis**_{1 of 2} + **dis**_{2 of 2}

Option 1: **dis**_{1 of 3} + **dis**_{2 of 3} + **dis**_{3 of 3}.

Option 2: **dis**_{1 of 2} + **dis**_{2 of 2} + **ecg**.

Option 3: **dis**_{1 of 2} + **dis**_{2 of 2} + **rad**.

Option 4: **dis**_{1 of 2} + **dis**_{2 of 2} + **ecg** + **rad**.

Option 5: **dis**_{1 of 4} + **dis**_{2 of 4} + **dis**_{3 of 4} + **dis**_{4 of 4}.

Option 6: **dis** + **ecg**.

Option 7: **dis** + **rad**.

7.5 Overall Performances of Language Models

This section presents overall micro and macro F1 scores for language model variations with several input text sequences. Critical difference plots are presented as supportive statistical analysis. The Nemenyi posthoc test (95% confidence level) identifies statistical differences between learning methods. CD graphs show the average ranking of individual F1 scores obtained using various language models. The lower the rank, the better it is. The difference in average ranking is statistically significant if there is no bold line connecting the two settings. This section also presents micro and macro F1 scores for each

Model Variations	Input	CVD	COVID-19	Infections	Level 2	Level 3
BioMed-RoBERTa	-	c6	c6	c6	c6	c6
PubMedBERT	-	c6	c6	c6	c6	c6
TransformerXL (3,072)	-	c6	c6	c6	-	-
Longformer (3,000)	-	c6	c6	c6	c6	c6
CAML	-	c5	c5	c5	c5	c5
CNNText	-	c5	c5	c5	c5	c5
Dual-Bio-RoBERTa	Option 0	✓	-	-	✓	✓
DualPubMedBERT	Option 0	✓	✓	✓	✓	✓
Triple-BioMed-RoBERTa	Option 1	✓	-	-	-	-
TriplePubMedBERT	Option 1	✓	✓	✓	✓	-
TriplePubMedBERT	Option 2	✓	✓	-	-	-
TriplePubMedBERT	Option 3	✓	✓	-	-	-
QuadruplePubMedBERT	Option 4	✓	✓	-	-	-
QuadruplePubMedBERT	Option 5	-	✓	✓	-	-
DualCNNText	Option 0	✓	✓	-	-	-
TripleCNNText	Option 2	✓	-	-	-	-
TripleCNNText	Option 3	✓	-	-	-	-
TripleCNNText	Option 1	-	✓	-	-	-
QuadrupleCNNText	Option 5	✓	✓	-	-	-
CNNText + BERT-base	Option 6	✓	-	-	-	-
CNNText + PubMedBERT	Option 6	✓	-	-	-	-
CNNText + PubMedBERT	Option 7	✓	-	-	-	-

Table 7.1: Summary checklist of the experimental results presented for various models. For the purpose of comparison, some results from Chapter 5 (c5) and Chapter 6 (c6) are also presented. Cardiovascular disease (CVD) presents experimental results with the most variations and other multi-label problems present selective model variations. T100SG is used for traditional embeddings-based neural networks and concatenations that use CNNText. Various input options are also indicated. See Tables 7.2 and 7.3 for details on input text options.

epoch for a range of one to 30, where selected language models are directly compared. Table 7.1 provides a summary checklist of various experimental results presented for language models. Cardiovascular disease is used in the

most variety of models. However, due to resource restrictions, only selective concatenated models are used for other multi-label problems.

7.5.1 Cardiovascular Disease

Table 7.2 presents results for various language model variations for cardiovascular disease, using MIMIC-III data where the number of instances is 28,154, and the number of labels is 30. Selected results from Chapters 5 and 6 are also presented for direct comparison. Multi-PubMedBERT options and multi-BioMed-RoBERTa options show a consistent improvement of 3 to 7% in micro-F1 scores over single PubMedBERT and BioMed-RoBERTa, respectively. The macro-F1 score of TriplePubMedBERT (option 2) is better than other language models presented with at least 3% improvement, except for TransformerXL with 3,072 tokens. Macro F1 scores of multi-CNNText and CNNText with transformers perform poorly compared to all other language models presented. For cardiovascular disease, incorporating `ecg` and `rad` does show some improved overall results, especially with TriplePubMedBERT options. Critical difference plots for individual label F1 scores obtained using various language models in Table 7.2 is presented in Figure 7.4. Both Table 7.2 and Figure 7.4 show TransformerXL with `dis` 3,072 tokens is the best option. However, multi-BioMed-Transformers show improvements, especially when compared to single-BioMed-Transformers.

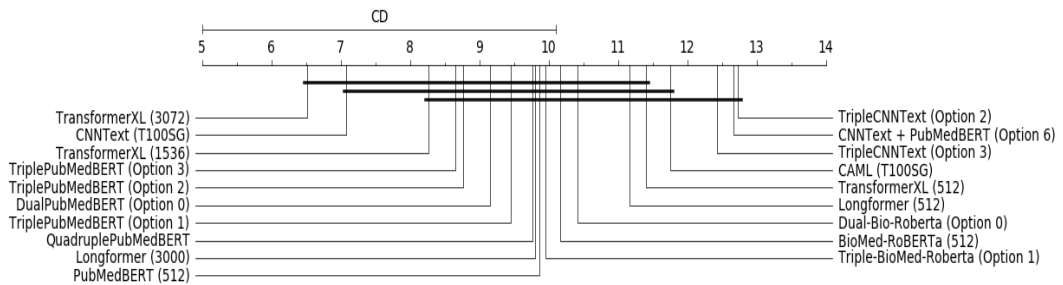


Figure 7.4: Critical difference plots. Nemenyi post-hoc test (95% confidence level), identifying statistical differences between language models for cardiovascular disease presented in Table 7.2.

Neural Network Details	Input Text Options	Micro-F1	Macro-F1
BioMed-RoBERTa	(c6) dis 512	0.69	0.30
PubMedBERT	(c6) dis 512	0.70	0.30
TransformerXL	(c6) dis 1,536	0.75	0.28
TransformerXL	(c6) dis 3,072	<u>0.77</u>	<u>0.32</u>
Longformer	(c6) dis 3,000	0.74	0.30
Dual-Bio-RoBERTa	Option 0: 512	0.72	0.28
DualPubMedBERT	Option 0: 512	0.72	0.30
Triple-BioMed-RoBERTa	Option 1: 512	0.72	0.29
TriplePubMedBERT	Option 1: 512	0.73	0.29
TriplePubMedBERT	Option 2: 512	0.73	0.31
TriplePubMedBERT	Option 3: 512	0.73	0.30
QuadruplePubMedBERT	Option 4: 512	0.74	0.28
CNNText (T100SG)	dis 512	0.72	0.23
DualCNNText (T100SG)	Option 0: 1,000	0.73	0.22
TripleCNNText (T100SG)	Option 2: 1,000	0.74	0.24
TripleCNNText (T100SG)	Option 3: 1,000	0.75	0.25
QuadrupleCNNText (T100SG)	Option 4: 1,000	0.74	0.22
CNNText (T100SG) + BERT-base	Option 6: dis 3,000 + ecg 512	0.75	0.20
CNNText (T100SG) + PubMedBERT	Option 6: dis 3,000 + ecg 512	0.76	0.22
CNNText (T100SG) + PubMedBERT	Option 7: dis 3,000 + rad 512	0.75	0.21
CAML (T100SG)	(c5) dis 3,000	<u>0.77</u>	0.24
CNNText (T100SG)	(c5) dis 3,000	0.74	0.30

Table 7.2: Comparison of micro-F1, macro-F1 of cardiovascular disease among various language models and input text for MIMIC-III data. The number of instances is 28,154, and the number of labels is 30. Input text options include the maximum sequence length, reference to the option, and where appropriate, a reference to the previous chapter the results were originally presented. Bold is used to indicate the best results for each grouping in the table, and underline is used for overall best results. Results are averaged over three runs.

Figure 7.5 presents a comparison of micro and macro F1 scores after each epoch for a range of 1 to 30 for selected language models presented in Table 7.2 for cardiovascular disease using MIMIC-III data. For TransformerXL (3,072), the experiments were only performed for 20 epochs as the required computational resources for these experiments are very high. Across all micro

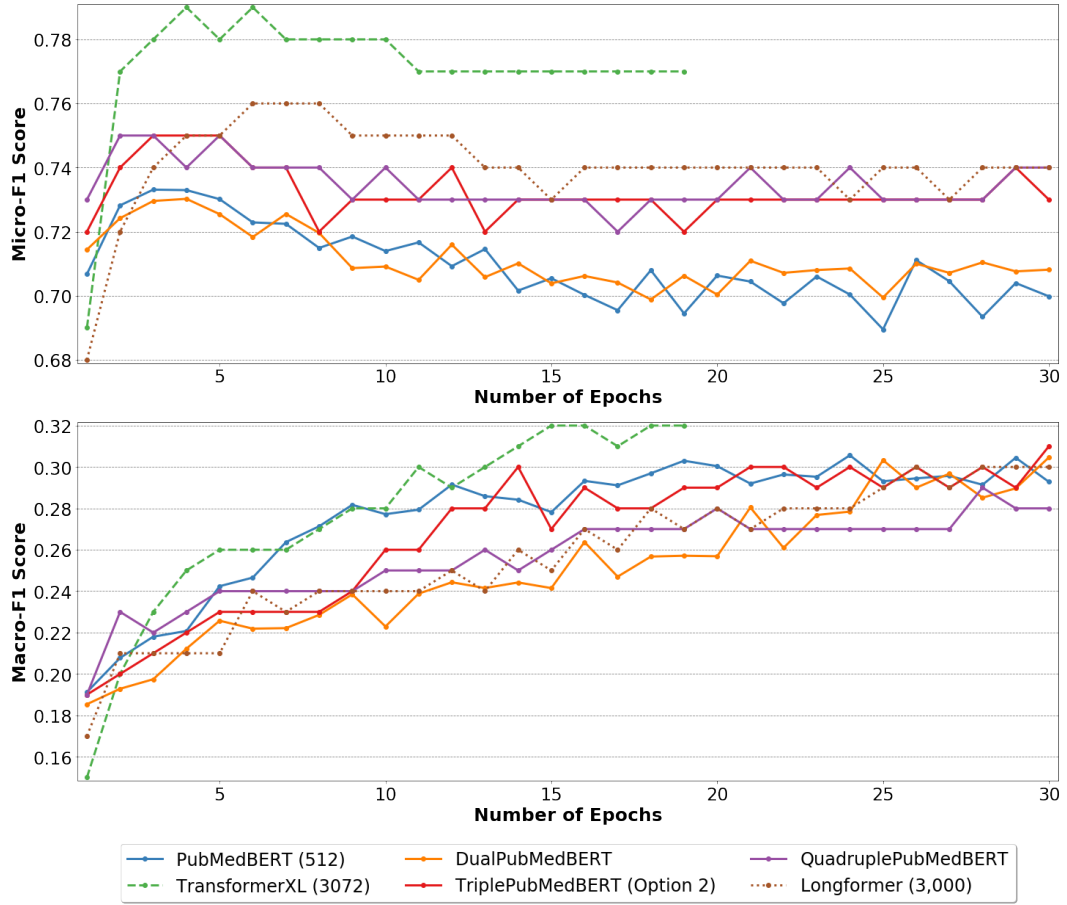


Figure 7.5: micro (top) and macro (bottom) F1 scores of selected language models for cardiovascular disease with varied epochs. Due to resource restrictions, TransformerXL (3,072) was only experimented for 20 epochs.

F1 scores, the rate of increase is very high from one to two epochs, with a peak of the best score after four to seven epochs, followed by a drop. Overall micro-F1 scores settle to a more stable measure after 15 epochs. As anticipated, TransformerXL (3,072) is considerably better than other methods. However, Longformer is not much different to triple and quadruple PubMedBERT. Compared to single PubMedBERT, dual is slightly better and with triple and quadruple showing more significant improvements. For macro-F1 scores, there is a direct proportionality to the increase in the epoch. Although, in general, TransformerXL is better than other models, Longformer (3,000) is not. Single PubMedBERT shows closer associations with the macro-F1 scores of TransformerXL from seven to 12 epochs. Dual and triple PubMedBERT show a higher rate of increase over the increase in epochs. There is a clear

difference in the behaviour of language models between micro and macro F1 scores. For this research, macro-F1 scores were prioritised, as the individual labels directly influence them.

7.5.2 Systemic Fungal or Bacterial Infection

Table 7.3 presents results for various language model variations for systemic fungal or bacterial infection, using MIMIC-III data where the number of in-

Neural Network Details	Input Text Options	Micro-F1	Macro-F1
PubMedBERT	(c6) dis 512	0.48	0.39
TransformerXL	(c6) dis 3,072	<u>0.64</u>	<u>0.46</u>
Longformer	(c6) dis 3,000	0.58	0.43
DualPubMedBERT	Option 0: 512	0.57	0.43
TriplePubMedBERT	Option 1: 512	0.56	0.40
TriplePubMedBERT	Option 2: 512	0.54	0.39
TriplePubMedBERT	Option 3: 512	0.54	0.40
QuadruplePubMedBERT	Option 5: 512	0.57	0.40
QuadruplePubMedBERT	Option 4: 512	0.54	0.40
DualCNNTText (T100SG)	Option 0: 2,000	0.54	0.27
TripleCNNTText (T100SG)	Option 1: 2,000	0.50	0.27
QuadrupleCNNTText (T100SG)	Option 5: 1,000	0.47	0.24
CAML (T100SG)	(c5) dis 3,000	0.62	0.38
CNNTText (T100SG)	(c5) dis 3,000	0.57	0.40

Table 7.3: Comparison of micro-F1, macro-F1 of systemic fungal or bacterial infection among various language models and input text for MIMIC-III data. Number of instances is 30,814 and number of labels is 73. Bold is used to indicate the best results for each groupings in the table, and underline is used for overall best results. Results are averaged over three runs. Maximum length of sequence is also indicated next to input text. Reference to options are also presented.

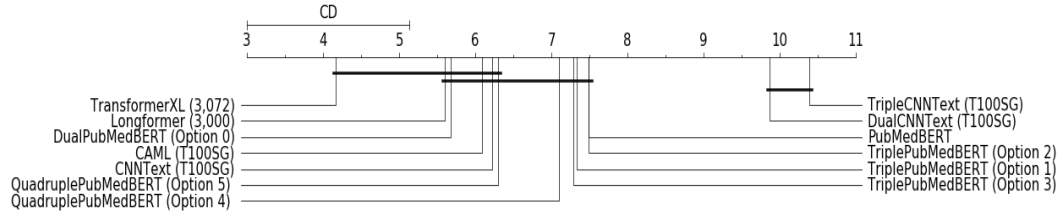


Figure 7.6: Critical difference plots. Nemenyi post-hoc test (95% confidence level), identifying statistical differences between language models for systemic fungal or bacterial infection presented in Table 7.3.

Neural Network Details	Tokens	Time per epoch
CNNTText (T100SG)	3,000	43 sec
CAML (T100SG)	3,000	47 sec
PubMedBERT	512	2,940 sec
TransformerXL	512	2,921 sec
TransformerXL	1,536	15,000 sec
TransformerXL	3,072	43,200 sec
Longformer	3,000	13,500 sec
DualPubMedBERT	1,024	4,020 sec
TriplePubMedBERT	1,536	5,580 sec
QuadruplePubMedBERT	2,048	7,080 sec
DualCNNTText (T100SG)	4,000	1,606 sec
TripleCNNTText (T100SG)	6,000	1,669 sec

Table 7.4: Systemic fungal or bacterial infection, time per epoch using MIMIC-III. Results are averaged over three runs.

stances is 30,814, and the number of labels is 73. Multi-PubMedBERT options show improvements of 12 to 19 % in micro-F1, and 2 to 10 % in macro-F1 scores over single PubMedBERT, except for TriplePubMedBERT (option 2), where the macro-F1 score is on par with single PubMedBERT. Contrary to the case of cardiovascular disease, additional input texts **ecg** and **rad** here do not result in better performance. One possible reason is that labels for systemic fungal or bacterial infection are not directly associated with **ecg** and **rad**. Macro F1 scores of multi-CNNTText perform poorly in comparison to all other

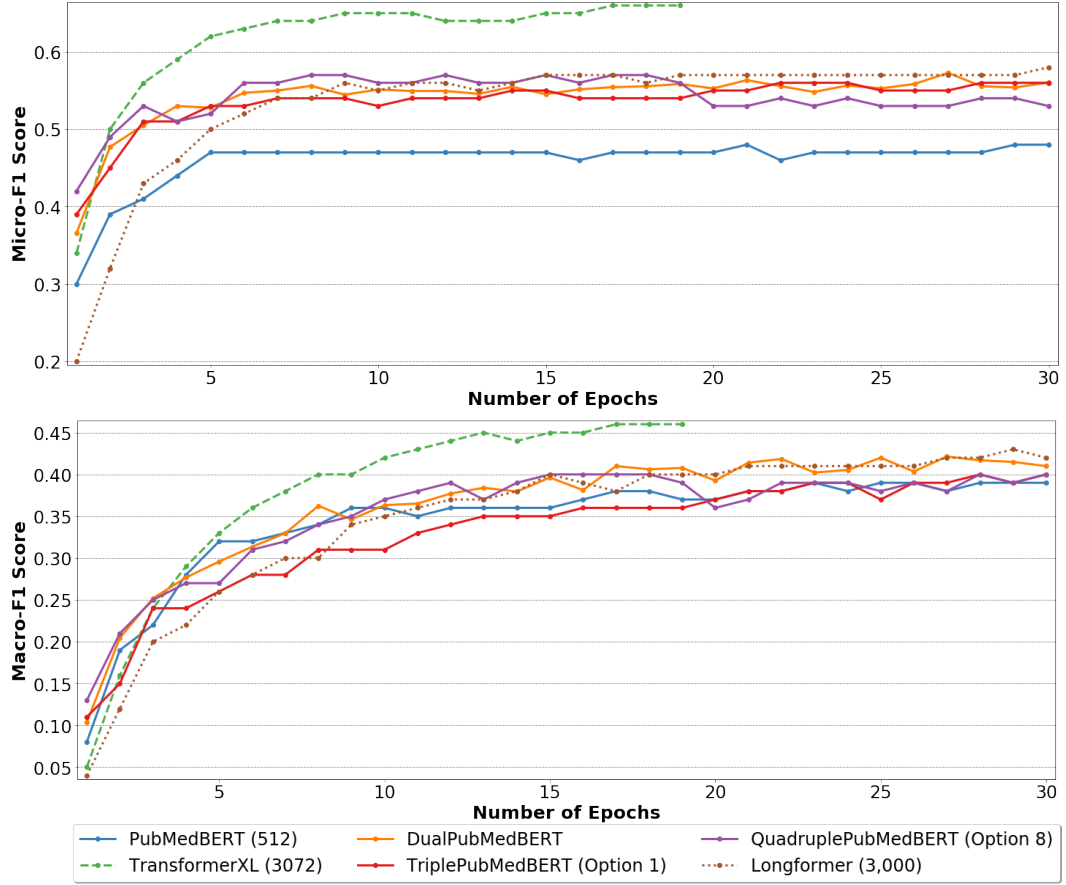


Figure 7.7: micro (top) and macro (bottom) F1 scores of selected language models for systemic fungal or bacterial infection with varied epochs. Due to resource restrictions TransformerXL (3,072) was only experimented for 20 epochs.

variations of language models presented. Critical difference plots for label F1 scores obtained using various language models in Table 7.3 are presented in Figure 7.6. Both Table 7.3 and Figure 7.6 show TransformerXL with dis 3,072 tokens to be the best option as observed in cardiovascular disease. However, multi-BioMed-Transformers show improvements, especially when compared to single-BioMed-Transformers.

Table 7.4 presents time per epoch in seconds for systemic fungal or bacterial infection to provide a direct comparison among the language models. TransformerXL (3,072) requirements are much higher than that of other language models, including multi-PubMedBERT. For example, if CNNTText is directly compared to TransformerXL (3,072), the time required per epoch is

1,000 times more.

Figure 7.7 presents a comparison of micro and macro F1 scores after each epoch for a range of one to 30 for selected language models presented in Table 7.3 for systemic fungal or bacterial infection using MIMIC-III data. For TransformerXL (3,072), the experiments were only performed for 20 epochs as the required computational resource for these experiments are very high, which is also evident from Table 7.4. For micro F1 scores, initially, the rate of increase is very high, followed by long, consistently stable measures. As anticipated, TransformerXL (3,072) is better than other methods. Longformer starts with lower measures followed by improved micro-F1 scores after 15 epochs. In comparison to single PubMedBERT, all variations of multi-PubMedBERT models show significant improvements. For Macro-F1 scores, all language model variations present a similar pattern where, as the number of epochs increases, the micro-F1 scores are increasing, with the rate of increase slowing down over later epochs. TransformerXL is better than the other models, with dualPubMedBERT performing better than Longformer for the initial ten epochs and following a similar path to Longformer after 20 epochs. Compared to cardiovascular disease, the trend of language models over epochs is similar.

7.5.3 COVID-19 Patient Shielding

Table 7.5 presents results for various language model variations for cardiovascular disease, using MIMIC-III data where the number of instances is 35,458,

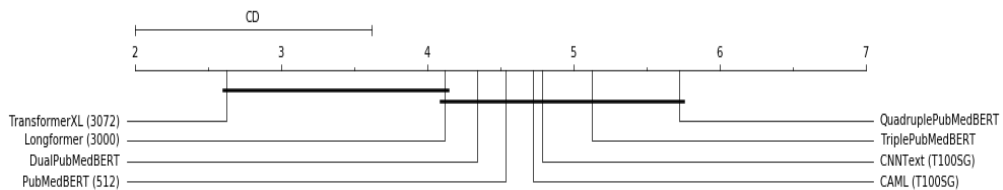


Figure 7.8: Critical difference plots. Nemenyi post-hoc test (95% confidence level), identifying statistical differences between language models for COVID-19 patient shielding presented in Table 7.5.

Neural Network Details	Input Text Options	Micro-F1	Macro-F1
PubMedBERT	(c6) dis 512	0.54	0.48
TransformerXL	(c6) dis 3,072	<u>0.65</u>	<u>0.51</u>
Longformer	(c6) dis 3,000	0.58	0.50
DualPubMedBERT	Option 0: 512	0.58	0.49
TriplePubMedBERT	Option 1: 512	0.54	0.46
QuadruplePubMedBERT	Option 5: 512	0.52	0.46
CAML (T100SG)	(c5) dis 3,000	0.64	0.40
CNNText (T100SG)	(c5) dis 3,000	0.58	0.42

Table 7.5: Comparison of micro-F1, macro-F1 of COVID-19 patient shielding among various language models and input text for MIMIC-III data. Number of instances is 35,458 and number of labels is 42. Bold is used to indicate the best results for each grouping in the table, and underline is used for overall best results. Results are averaged over three runs. Maximum length of sequence is also indicated next to input text.

and the number of labels is 42. As observed with the other case studies, both Table 7.5 and Figure 7.8 show TransformerXL with dis 3,072 tokens to be the best option. DualPubMedBERT shows improvements over single PubMedBERT and other variations of multi-PubMedBERT.

Figure 7.9 presents a comparison of micro and macro F1 scores after each epoch for a range of one to 30 for selected language models presented in Table 7.5 for COVID-19 patient shielding using MIMIC-III data. For TransformerXL (3,072), the experiments were only performed for 15 epochs. For micro F1 scores, initially, the rate of increase is very high, followed by long, consistently stable measures, with the exception of dualPubMedBERT, which drops slightly at 12 epochs. TransformerXL (3,072) is much better than other methods after a couple of epochs. Although dualPubMedBERT performs better than other variations of PubMedBERT, micro and macro F1 scores of single

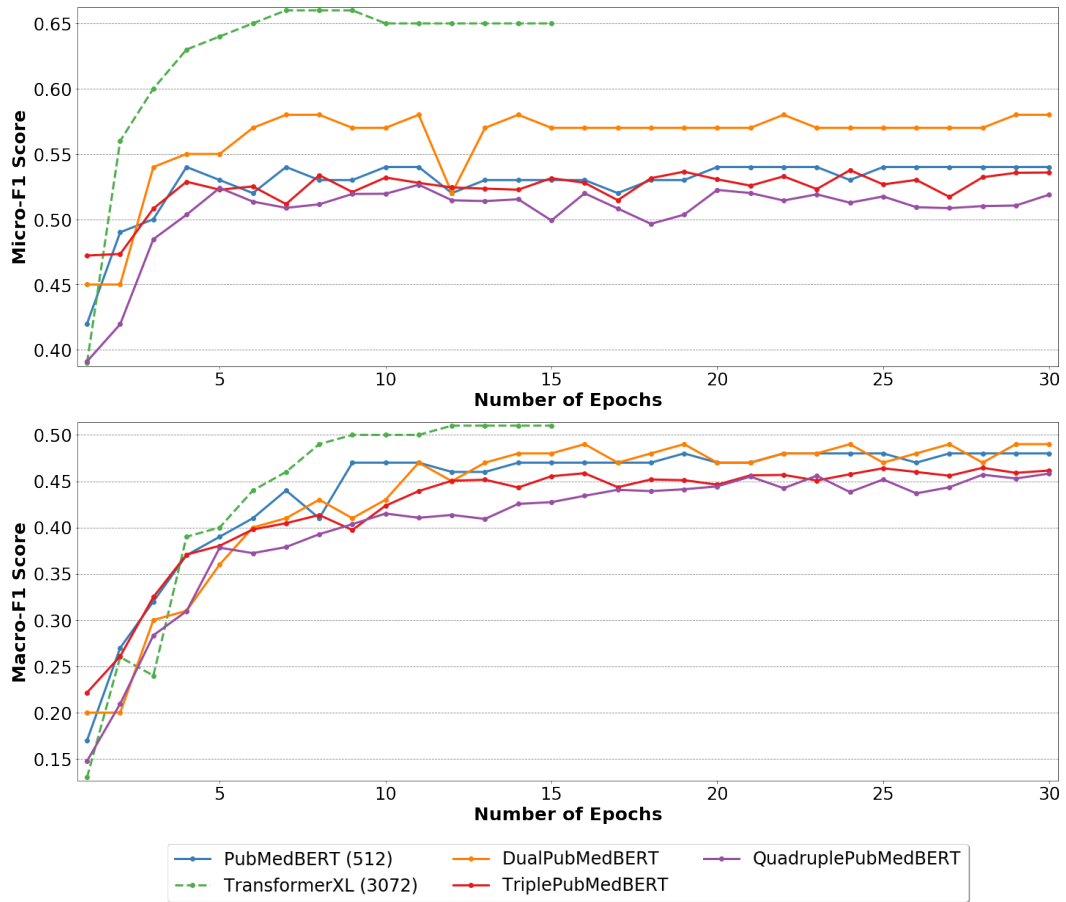


Figure 7.9: micro (top) and macro (bottom) F1 scores of selected language models for COVID-19 patient shielding with varied epochs. Due to resource restrictions, TransformerXL (3,072) was only experimented for 15 epochs.

PubMedBERT are better than that of triple and quadruple PubMedBERT.

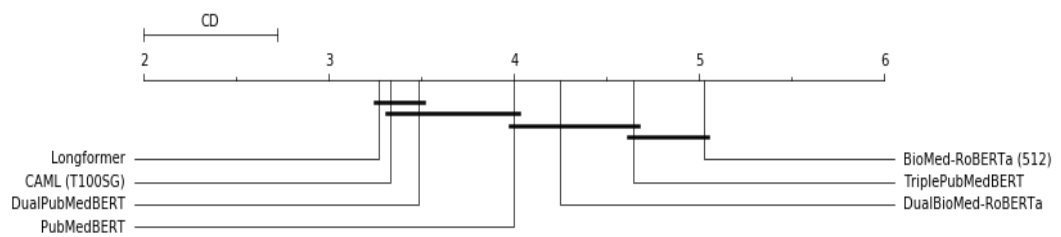


Figure 7.10: Critical difference plots. Nemenyi post-hoc test (95% confidence level), identifying statistical differences between language models for levels 2 of ICD-9 codes presented in Table 7.6.

Level 2 ICD-9 codes, 158 labels, MIMIC-III			
Neural Network Details	Input Text Options	Micro-F1	Macro-F1
CAML, T100SG	(c5) dis 3,000	<u>0.72</u>	0.43
PubMedBERT	(c6) dis 512	0.65	0.43
BioMed-RoBERTa	(c6) dis 512	0.64	0.41
Longformer	(c6) dis 3,000	<u>0.72</u>	<u>0.45</u>
DualPubMedBERT	Option 0: 512	0.68	<u>0.45</u>
DualBioMed-RoBERTa	Option 0: 512	0.66	0.43
TriplePubMedBERT	Option 1: 512	0.66	0.43
Level 3 ICD-9 codes, 923 labels, MIMIC-III			
CAML, T100SG	(c5) dis 3,000	<u>0.64</u>	<u>0.26</u>
PubMedBERT	(c6) dis 512	0.55	0.18
BioMed-RoBERTa	(c6) dis 512	0.53	0.18
Longformer	(c6) dis 3,000	0.60	0.17
DualPubMedBERT	Option 0: 512	0.57	0.20
DualBioMed-RoBERTa	Option 0: 512	0.56	0.19

Table 7.6: Comparison of micro-F1, macro-F1 of levels 2 and 3 of ICD-9 codes among various language models and input text for MIMIC-III data. The number of instances is 52,722 for both levels, and the number of labels is 158 for level 2 and 923 for level 3. Bold is used to indicate the best results for each grouping in the table, and underline is used for overall best results. Results are averaged over three runs. A maximum length of the sequence is also indicated next to input text.

7.5.4 Levels 2 and 3 ICD-9 codes

This section compares the overall performances of language models for levels 2 and 3 of ICD-9 codes. Due to the computational restrictions, TransformerXL (3,072) was not used for long documents. Table 7.6 presents results for various language model variations for levels 2 and 3 of ICD-9 codes, using MIMIC-III data where the number of instances is 52,722, and the number of labels is 158

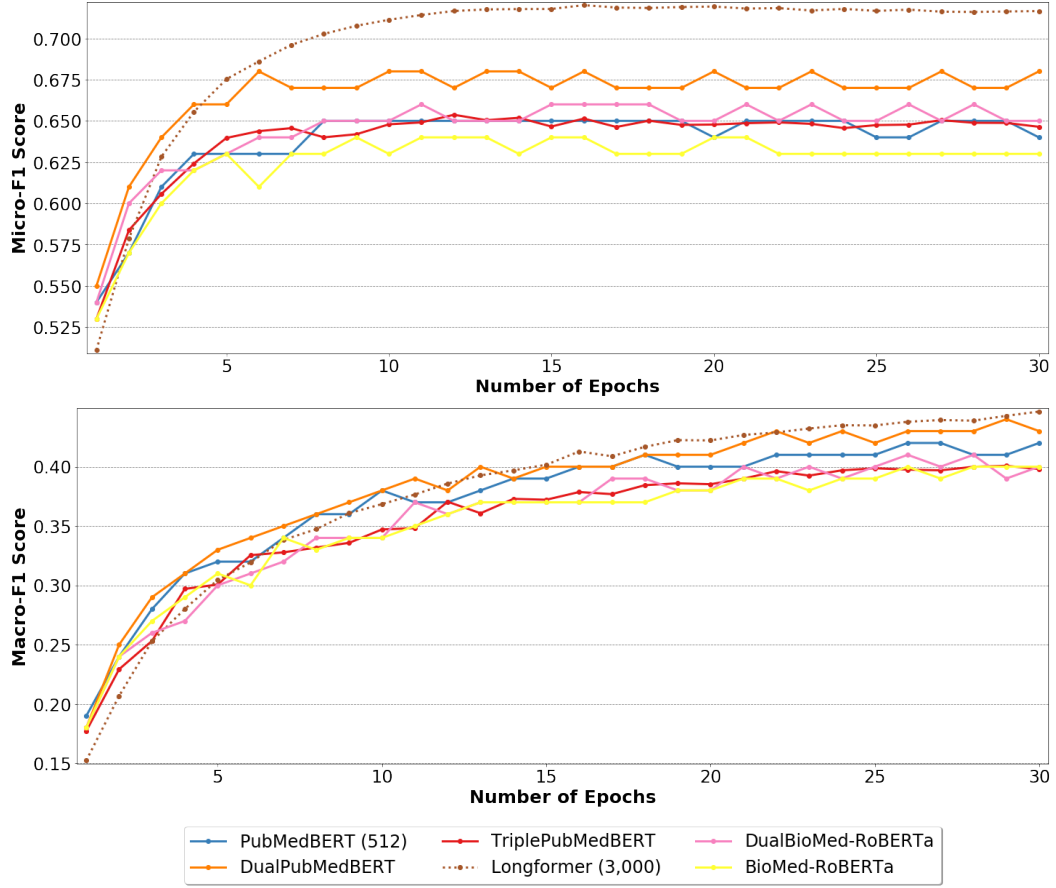


Figure 7.11: Micro (top) and macro (bottom) F1 scores of selected language models for level 2 ICD-9 codes with varied epochs.

for level 2 and 923 for level 3. Selected results from Chapters 5 and 6 are also presented for direct comparison. For level 2 ICD-9 codes, Longformer with dis 3,000 tokens is the best option, as indicated in Table 7.6 and Figure 7.11 - with one exception, where macro-F1 of DualPubMedBERT is equal to that of Longformer. DualPubMedBERT shows improvements in both micro and macro F1 scores by 3 to 5% over other PubMedBERT variations. For level 3 ICD-9 codes, CAML (T100SG) is the best option with the best micro and macro F1 scores. However, it is essential to point out that the macro-F1 score of DualPubMedBERT is better than other transformer models, including Longformer, where the macro-F1 score improves from 0.17 to 0.20.

Figure 7.11 presents a comparison of micro and macro F1 scores after each epoch for a range of one to 30 for selected language models presented in Table 7.6 for level 2 ICD-9 codes using MIMIC-III data. For micro F1 scores,

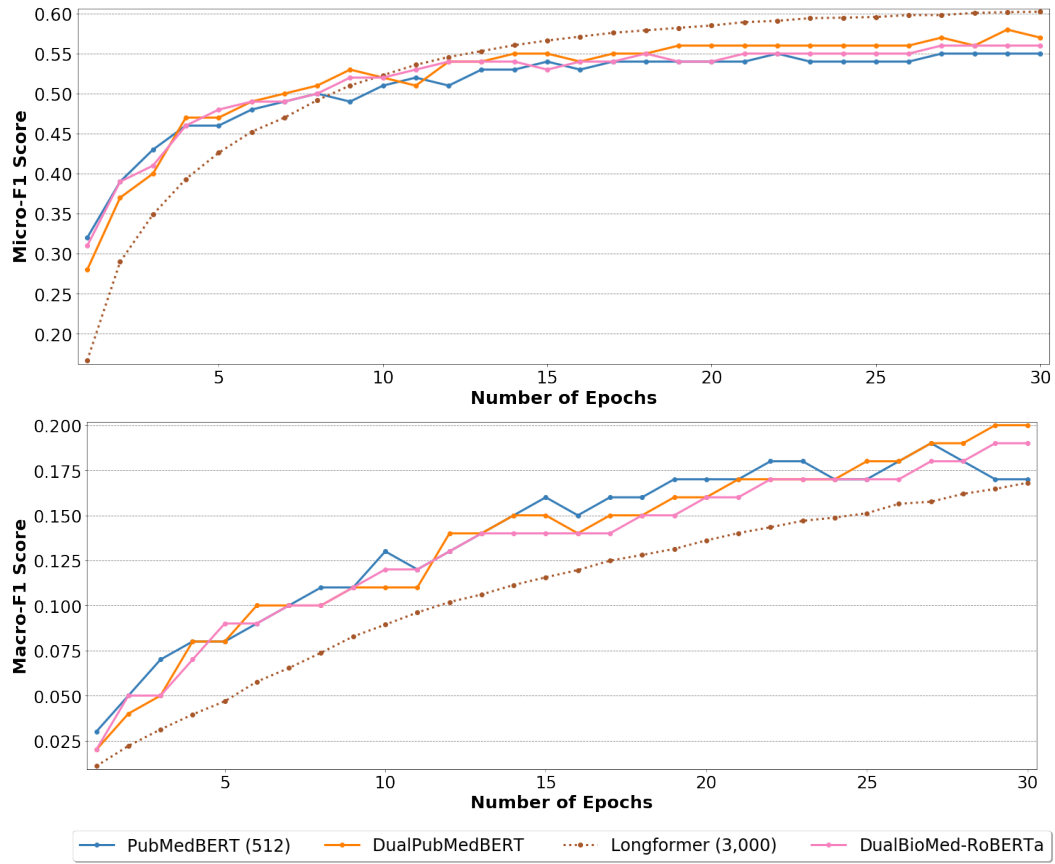


Figure 7.12: Micro (top) and macro (bottom) F1 scores of selected language models for level 3 ICD-9 codes with varied epochs.

initially, the rate of increase is very high, followed by long, consistently stable measures. Longformer (3,000) is much better than other methods after the first five epochs. DualPubMedBERT performs better than other variations of PubMedBERT and BioMed-RoBERTa, with the lowest micro-F1 scores from single BioMed-RoBERTa. For Macro-F1 scores, all language model variations present a similar pattern where, as the number of epochs increases, the micro-F1 scores are increasing, with the rate of increase slowing down over epochs. Initially, DualPubMedBERT performs better than other models, with Longformer overtaking macro-F1 scores of all language models after 15 epochs. However, in general, the macro-F1 scores of language models over epochs are closely related to each other, with BioMed-RoBERTa variations at the lower end of the scale.

Figure 7.12 presents a comparison of micro and macro F1 scores after each

epoch for a range of one to 30 for selected language models presented in Table 7.6 for level 3 ICD-9 codes using MIMIC-III data. Micro-F1 scores of language models are closely related to each other, with Longformer starting with the lowest micro-F1 scores and becoming the model with the best micro-F1 scores after ten epochs. For Macro-F1 scores, Longformer performs consistently worse than all other models across all epochs.

7.5.5 Summary

This section compared multiple language models for MIMIC-III data for the number of labels being 30, 42, 73, 158 and 923. TransformerXL (3,072) consistently outperformed other language models. Compared to single BioMed-Transformers, multi-BioMed-transformers outperforms, with a more noticeable improvement in micro-F1 scores for cardiovascular disease and systemic fungal or bacterial infection. Only Longformer was used to handle long text sequences for levels 2 and 3 of ICD-9 codes due to computational restrictions. For level 2 macro-F1 score of ICD-9 codes, dualPubMedBERT was the same as Longformer, and for level 3 with 923 labels, the macro-F1 score was better than Longformer for dual-BioMed-transformer models. The trend of micro-F1 and macro-F1 scores after each epoch was also presented. For micro-F1, in general, there is an initial rapid growth in scores, followed by stable measures. Macro-F1 scores improve as the number of epochs increases, with the rate of increase slowing down over epochs. Macro-F1 scores are calculated by averaging the F1 scores of labels, and as the number of labels increases, the required epochs for a more stable macro-F1 score also increases. For example, COVID-19 with 42 labels the change in macro-F1 score is minimum after 15 epochs. However, for level 3 ICD-9 codes with 923 labels, there is an increase in macro-F1 scores with the number of epochs, where Longformer could overtake the other methods over a longer run.

7.6 Tail-end Labels

This section presents a comparison of individual label F1 scores for the multi-label problems presented in Section 7.5. The focus here is on showing the differences and the improvements in F1 scores of tail-end labels with multi-BioMed-Transformers compared to Longformer and TransformerXL. Labels with frequency $\geq 1\%$ are grouped, and tail-end labels (with label frequency $< 1\%$) are grouped for clear differentiation. Throughout Chapters 5 and 6, tail-end label F1 scores were highlighted for reference. Differences are calculated by taking the actual differences of F1 scores of Longformer or TransformerXL from a particular language model variation, i.e., F1 scores of a label (multi-BioMed-Transformers) - F1 scores of the same label (Longformer (3,000)). Hence, negative values indicate Longformer or TransformerXL with the better F1 score.

In order to make it easier to follow, this section starts with the two multi-label problems, levels 2 and 3 of ICD-9 codes, where only Longformer (3,000) was used for extended sequences of clinical text. This is followed by the three case studies, where the differences between TransformerXL (3,072)/Longformer (3,000) and selected variations of language models are presented.

Heat maps illustrating the number of label wins, losses and draws are also included for each multi-label problem. This provides an additional demonstration of the performances of concatenated language models compared to TransformerXL (3,072)/Longformer (3,000). For multi-label problems, F1-scores of many infrequent labels are zero. This observation is also evident in F1 scores presented in Chapter 5 and 6. In order to quantify, the observations in the differences of F1 scores are presented as wins, draws and losses. For most cases, draws occur when both F1 scores for a label are zero.

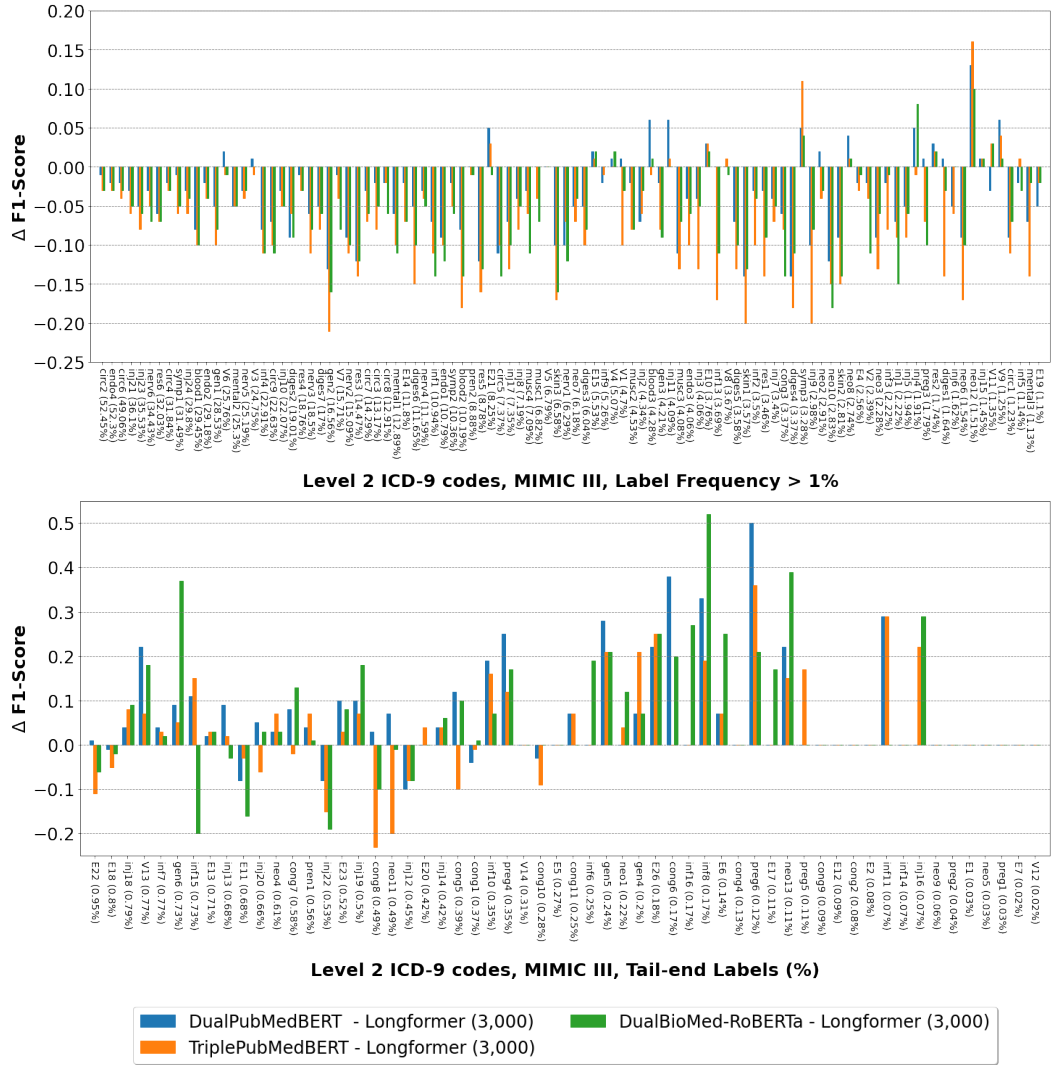


Figure 7.13: Level 2 ICD-9 codes, where the difference between F1 scores of dual/triple language model variations and Longformer (3,000 tokens) is presented. Top plot: Label frequency $\geq 1\%$ where labels are ordered based on frequency with most frequent label at the left of the plot. Bottom plot: Tail-end labels (frequency $< 1\%$) with labels ordered based on frequency with the least infrequent label at the right end. Legend is presented for references. Negative F1 scores indicated better F1 scores for Longformer (3,000).

7.6.1 Levels 2 and 3 ICD-9 codes

Figure 7.13 presents the difference in F1 scores for three combinations: dualPubMedBERT - Longformer (3,000), triplePubMedBERT - Longformer (3,000) and dualBioMed-RoBERTa - Longformer (3,000). Two separate graphs are presented for label frequency $\geq 1\%$ and tail-end labels. For frequent labels, the F1 scores of Longformer are better than the other three models. Smaller

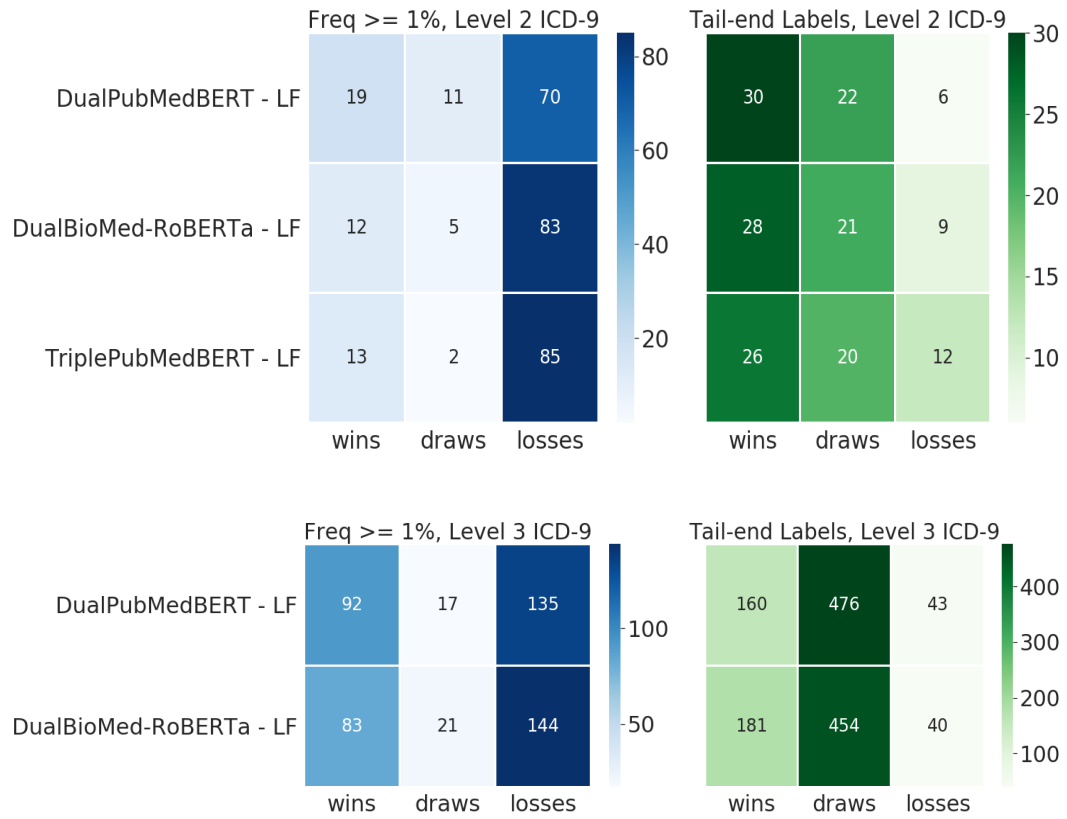


Figure 7.14: Heat maps for levels 2 and 3 of ICD-9 codes illustrating the number of wins, draws and losses of concatenated language models compared to Longformer (LF). Label Frequency $\geq 1\%$ is in blue, and tail-end labels with frequency $< 1\%$ is in green. Number of labels that win, loss and draw with LF are also indicated.

differences in F1 scores are noticed among the most frequent labels, and occasional dual and triple models perform slightly better than Longformer for specific labels. In general, Longformer has the most wins over other models for label frequency $\geq 1\%$. This pattern is reversed for tail-end labels, with Longformer losing more to the dual and triple models where the difference in F1 scores is noted. For some labels these differences can be rather large, reaching a maximum value of 0.52 for label `inf8`.

Figure 7.14 presents heat maps for the number of label wins, draws and losses for Level 2 and 3 ICD-9 codes. For tail-end labels there are more wins achieved by concatenated models. DualPubMedBERT is the best performing option with the least number of losses among the more frequent label group and most wins among the tail-end labels.



Figure 7.15: Level 3 ICD-9 codes, where the difference between F1 scores of dual language model variations and Longformer (3,000 tokens) is presented. Top two plots: Label frequency $\geq 1\%$ where labels are ordered based on frequency, with most frequent label at the left of the plot (only selected frequencies are indicated on x-axis). Bottom two plots: Tail-end labels (frequency $< 1\%$) with labels ordered based on frequency, with the least frequent label at the right end. Considering the large number of tail-end labels (more than 650 labels), individual labels are not indicated in x-axis. Legend is presented for references. Negative F1 scores indicated better F1 scores for Longformer (3,000).

Figure 7.15 presents the difference in F1 scores for dual transformers and Longformer (3,000). Level 3 contains 923 labels, with more than 650 labels being infrequent. Figure 7.15 also shows Longformer losing more to dual transformers at tail-end labels. This observation is further illustrated through the heat maps presented in Figure 7.14, where for tail-end labels, the majority of F1 scores obtained using dual transformers wins or draws with Longformers' F1 scores.

7.6.2 Case Studies

Figure 7.16 and Figure 7.17 present the difference in F1 scores for systemic fungal or bacterial infection between multi-PubMedBERT and TransformerXL, and multi-PubMedBERT and Longformer (3,000) respectively. For frequent labels, the F1 scores of TransformerXL are consistently better than the PubMedBERT variations and a clear winner. For infrequent labels, multi-PubMedBERT variations perform better than TransformerXL and Longformer for many labels. This observation is further illustrated by the heat map presented in Figure 7.18, where for tail-end labels, concatenated transformer models show improvements.

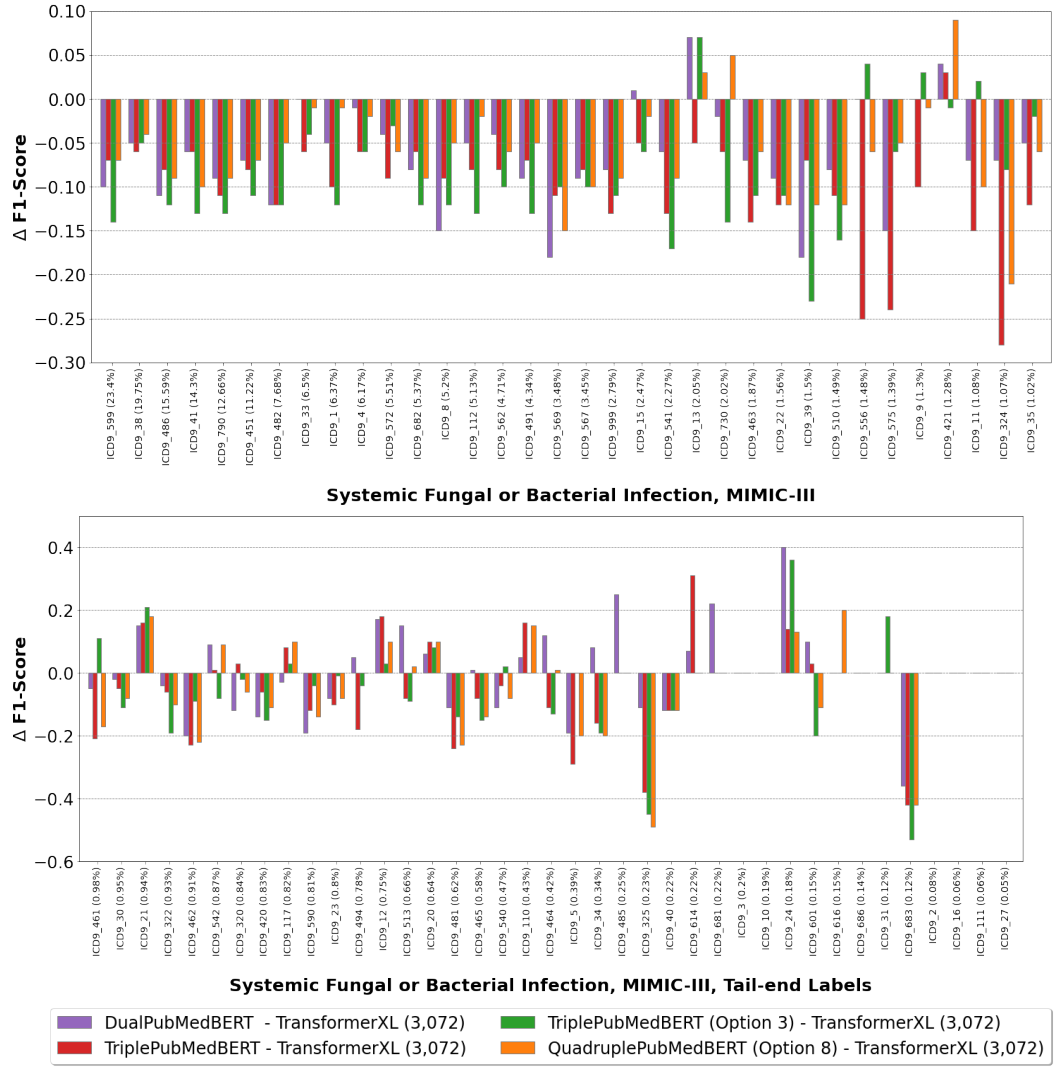


Figure 7.16: Systemic fungal or bacterial infection, where the difference between F1 scores of dual/triple language model variations and TransformerXL (3,072 tokens) is presented. Top plot: Label frequency $\geq 1\%$ where labels are ordered based on frequency, with most frequent label at the left of the plot. Bottom plot: Tail-end labels (frequency $< 1\%$) with labels ordered based on frequency, with the least frequent label at the right end. Legend is presented for references. Negative F1 scores indicated better F1 scores for TransformerXL (3,072 tokens).

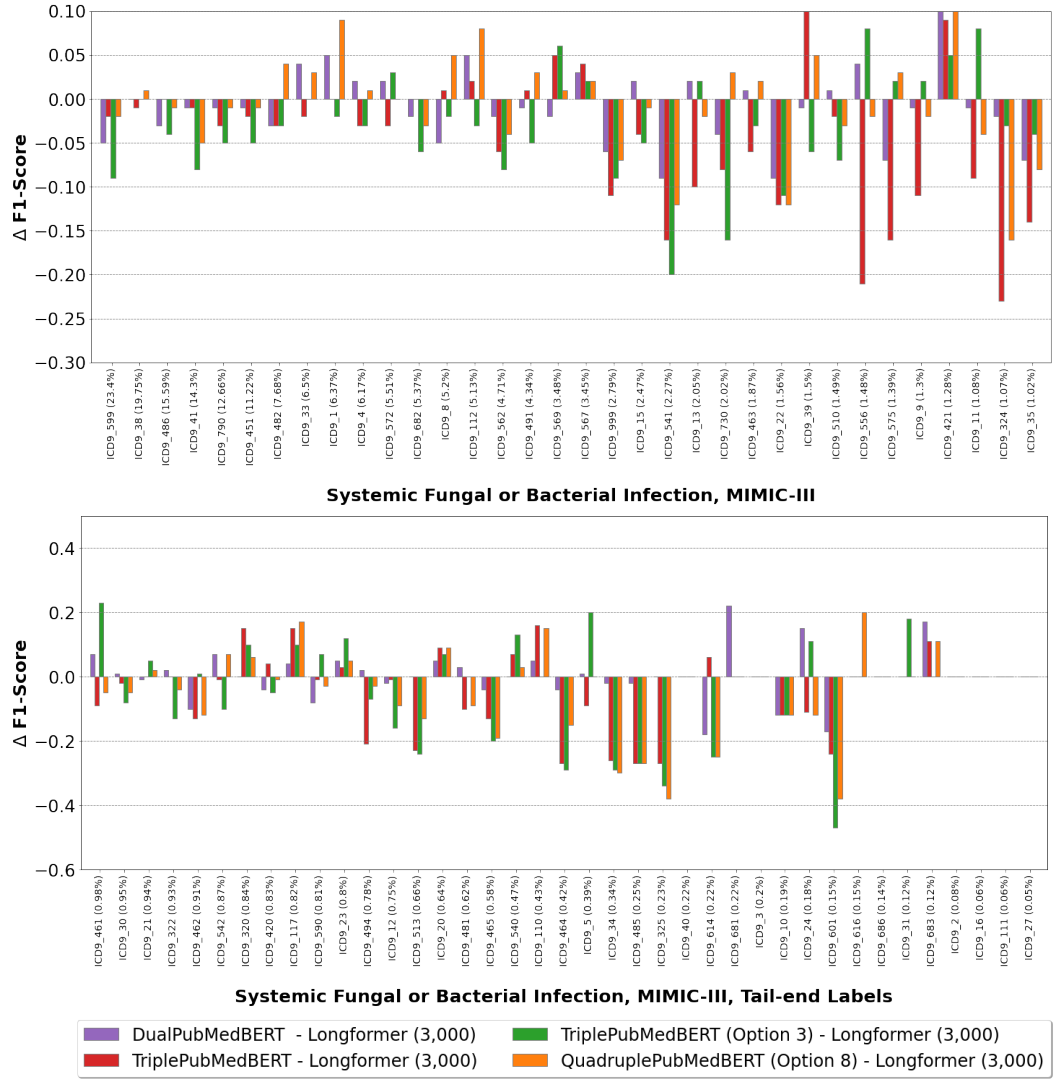


Figure 7.17: Systemic fungal or bacterial infection, where the difference between F1 scores of dual/triple language model variations and Longformer (3,000 tokens) is presented. Top plot: Label frequency $\geq 1\%$ where labels are ordered based on frequency with most frequent label at the left of the plot. Bottom plot: Tail-end labels (frequency $< 1\%$) with labels ordered based on frequency with the least frequent label at the right end. Negative F1 scores indicated better F1 scores for Longformer (3,000 tokens).

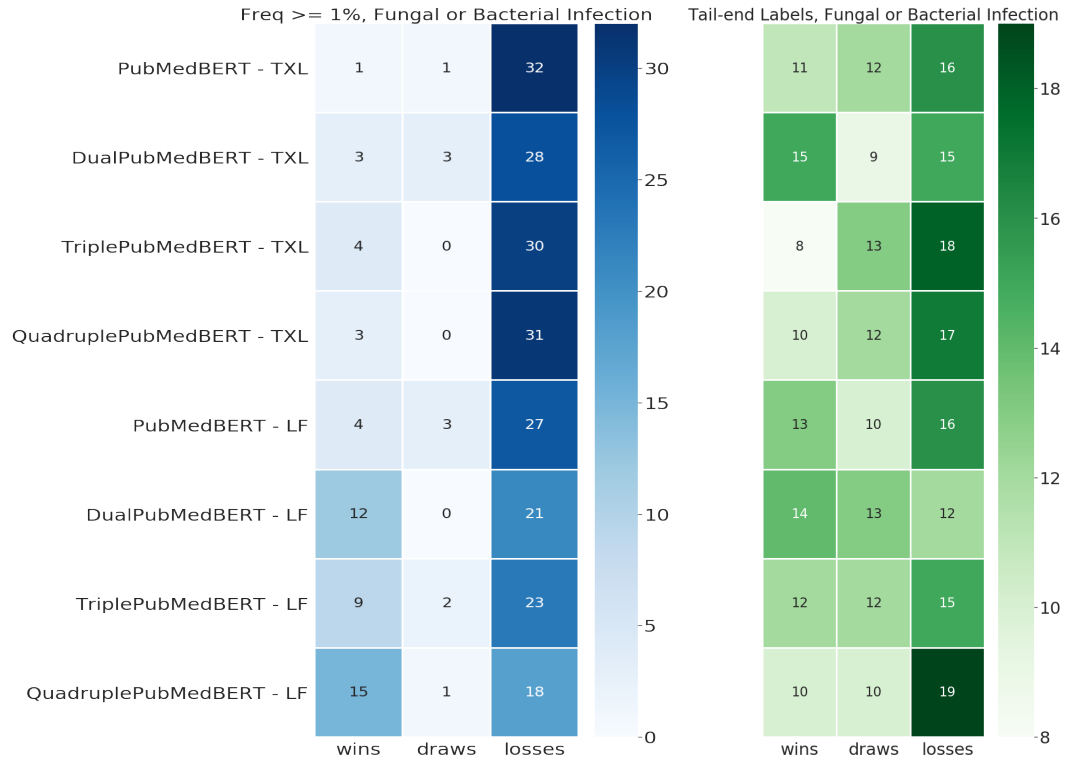


Figure 7.18: Heat maps for systemic fungal or bacterial infections illustrating the number of wins, draws and losses of concatenated language models compared to Longformer (LF) and TransformerXL (TXL). Label Frequency $\geq 1\%$ is in blue, and tail-end labels with frequency $< 1\%$ is in green. The number of labels that win, loss and draw with LF/TXL is also indicated.

Figure 7.19 presents the difference in F1 scores for COVID-19 patient shielding between multi-PubMedBERT and TransformerXL (3,072)/Longformer (3,000). Across all the label frequencies, F1 scores of Longformer or TransformerXL are generally better than other language models, with a few exceptions, including label ‘ICD9_508’. Heat maps as presented in Figure 7.20 provide further evidence, where for COVID-19 with only 42 labels in MIMIC-III, TransformerXL (3,072) is better than concatenated PubMedBERT models. The number of wins and losses for tail-end labels are between 4 - 6 labels.

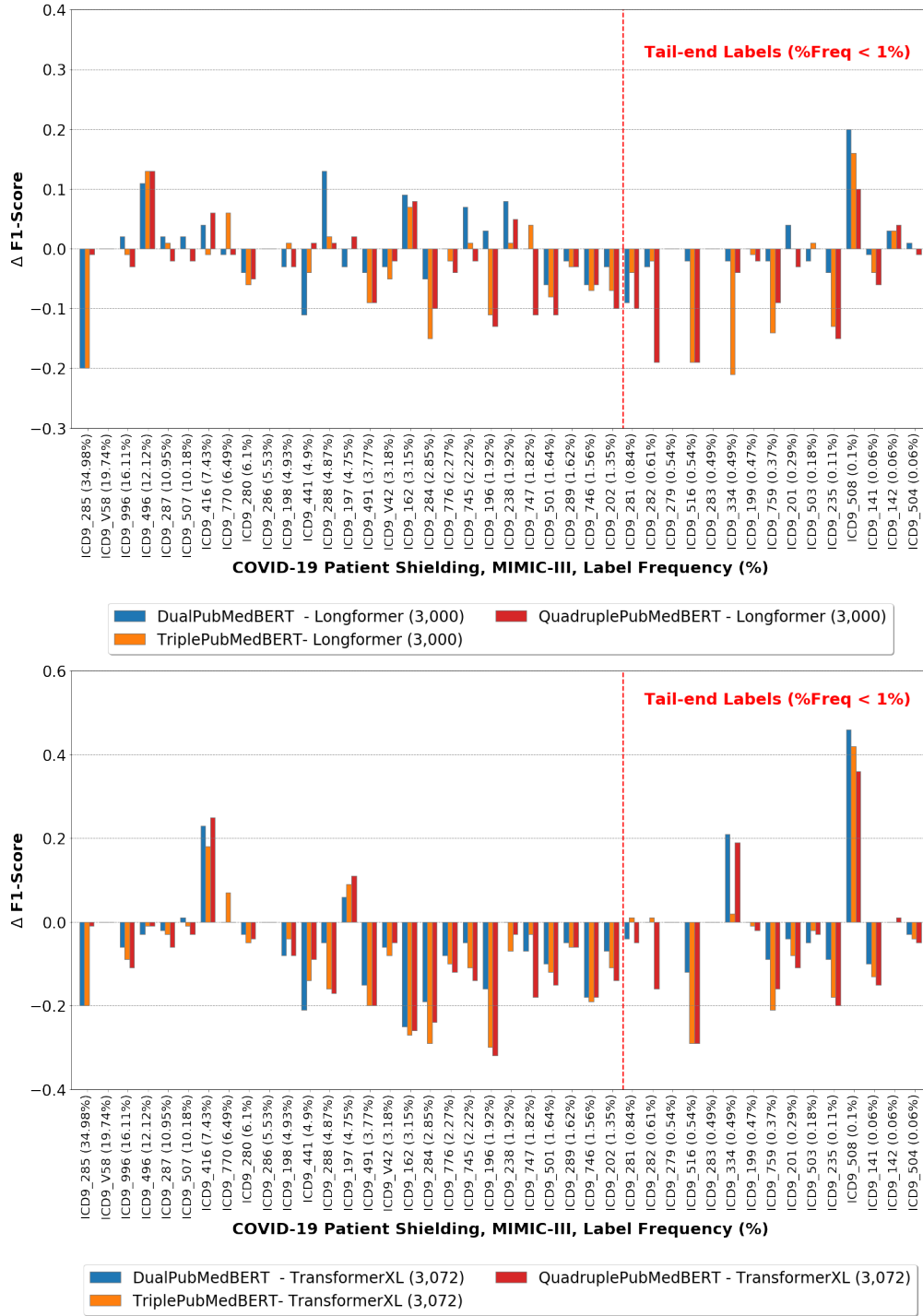


Figure 7.19: COVID-19 patient shielding, where the difference between F1 scores of multi-PubMedBERT and Longformer (3,000 tokens) (top plot) and the difference between F1 scores of multi-PubMedBERT and TransformerXL (3,072 tokens) (bottom plot) is presented. Legend is presented for references. Negative F1 scores indicated better F1 scores for Longformer (3,000 tokens) or for TransformerXL (3,072). tail-end labels are indicated on the plots.

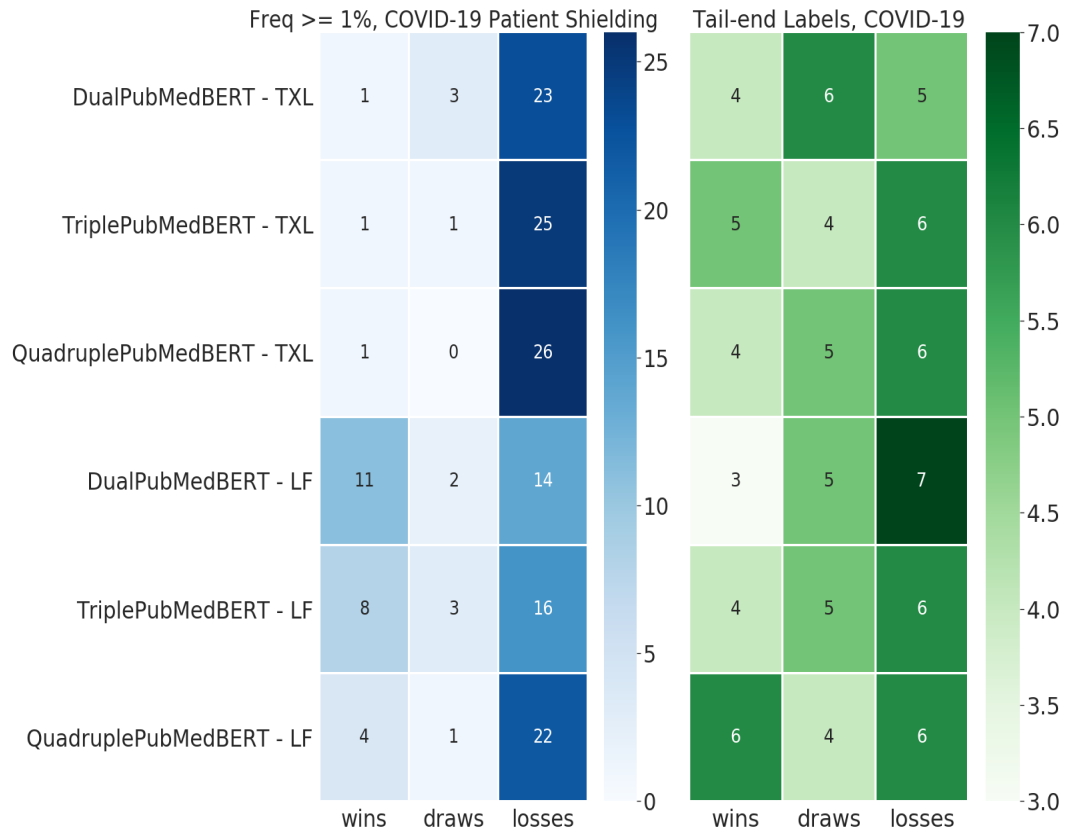


Figure 7.20: Heat maps for COVID-19 patient shielding illustrating the number of wins, draws and losses of concatenated language models compared to Longformer (LF) and TransformerXL (TXL). Label Frequency $\geq 1\%$ is in blue, and tail-end labels with frequency $< 1\%$ is in green. The number of labels that win, loss and draw with LF/TXL is also indicated.

Figure 7.21 presents the difference in F1 scores for cardiovascular disease between various language model variations and TransformerXL (3,072)/Longformer (3,000). The cardiovascular disease domain only incorporates 30 labels. Across the label frequencies, F1 scores of Longformer or TransformerXL are better than for the other language models, with only a few exceptions, including label ‘ICD9_40291’. Heat map presented in Figure 7.22 provides further evidence to this observation.

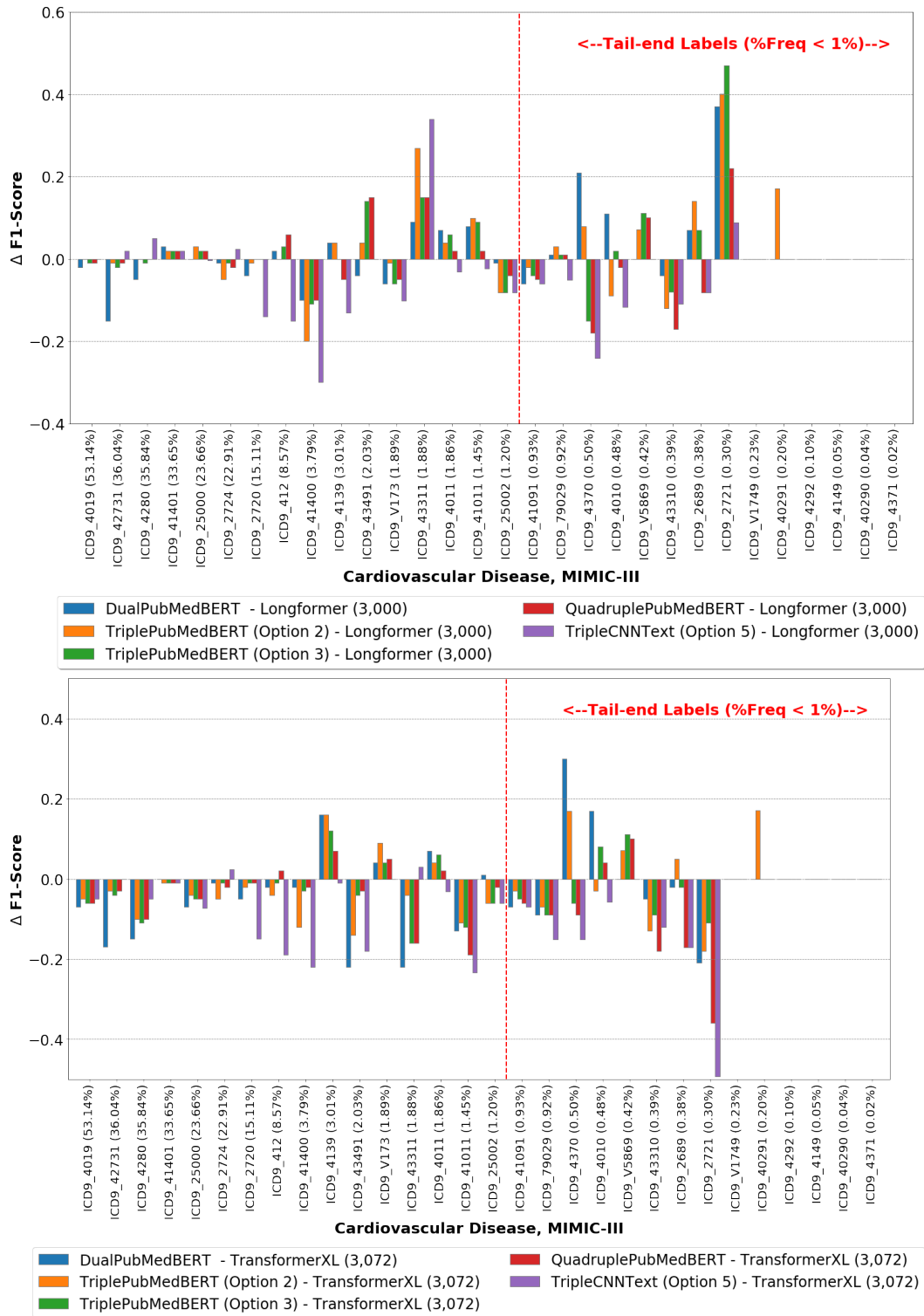


Figure 7.21: Cardiovascular disease, where the difference between F1 scores of language model variations and Longformer (3,000 tokens) (top plot) and the difference between F1 scores of language model variations and TransformerXL (3,072 tokens) (bottom plot) is presented. Legend is presented for references. Negative F1 scores indicated better F1 scores for Longformer (3,000 tokens) or for TransformerXL (3,072). tail-end labels are indicated on the plots.

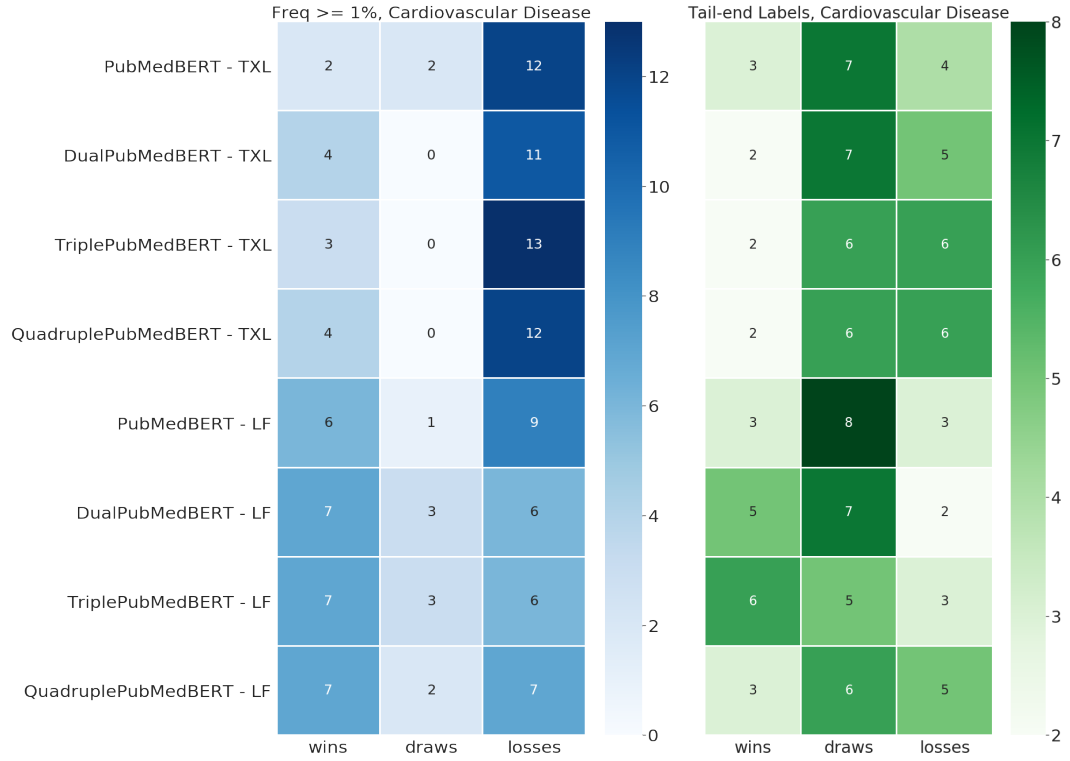


Figure 7.22: Heat maps for cardiovascular disease illustrating the number of wins, draws and losses of concatenated language models compared to Longformer (LF) and TransformerXL (TXL). Label Frequency $\geq 1\%$ is in blue, and tail-end labels with frequency $< 1\%$ is in green. The number of labels that win, loss and draw with LF/TXL is also indicated.

7.7 Discussions

This chapter presented concatenated language model variations, especially transformer-based ones, to improve the overall performance of infrequent multi-label problems where the input text data sequence is long. Although TransformerXL and Longformer can encode long sequences, and in general, TransformerXL outperforms other models, the required computational resources are restrictive. As indicated in several subsections of the thesis TransformerXL requires considerable GPU/CPU resources and time. For example, for MIMIC-III level 1 ICD-9 codes, TransformerXL for a sequence length of 3,072 required 73,655sec (or 20.45 hours) per epoch, which means for 15 epochs and 3 separate runs $= 20.45 \times 15 \times 3 = 921$ hours of processing time. Such long run times

rendered large-scale TransformerXL experiments infeasible.

Three main concatenated variations were presented where multi-PubMedBERT models outperformed other variations and also single BioMed-Transformers. There was a noticeable improvement in micro-F1 for multi-BioMed-transformers with cardiovascular disease and systemic fungal or bacterial infection. For larger multi-label problems, compared to Longformer, the macro-F1 score of dualPubMedBERT is the same for level 2 ICD-9 codes and better than Longformer for level 3 ICD-9 codes.

This research also studies the impact on predictive performance for less frequent labels. Label frequency is highly biased to the hospital/department the data were collected. If the data were from a fertility ward, the label frequency of pregnancy-related medical codes would be high, while for the cardiovascular ward this may not be the case. However, only being able to predict highly frequent labels well, pose risks to the patients' health and well being. Hence, this research also compared individual label F1 scores for multi-label problems focusing on tail-end labels. For larger multi-label problems with long tail-end labels, such as level 2 and 3 ICD-9 codes, multi-BioMed-transformers had more wins than Longformer.

This chapter has provided experimental evidence shows that, with fewer resources, concatenated BioMed-Transformers can improve overall micro and macro F1 scores for multi-label problems with long medical text. In addition, for multi-label problems with many tail-end labels, multi-BioMed-Transformers outperform other language models when F1 scores of tail-end labels are compared directly.

Chapter 8

Conclusions

This thesis started by explaining the advantages of machine learning techniques in healthcare and considers predicting medical codes as a multi-label classification problem. Moreover, this thesis argues that infrequent labels are as important as frequent labels when a patient’s health is concerned. Several avenues were explored to achieve two main goals:

1. Improving SOTA for predicting medical codes from EHRs as multi-label classification problems.
2. Improving the accuracy of predictions of infrequent labels for long medical documents.

In regards to the first point, Chapters 4 and 5 present domain-specific multi-sourced fastText pre-trained word embeddings with various dimensionalities. Experimental results show that changes to dimensionalities and training methods— skip-gram in comparison to CBOW, and health-related data compared to general text— are all factors that influence the overall predictive accuracy of classifiers. Neural networks provide better accuracy than traditional machine learning classifiers such as ECC. Among embeddings-based neural networks, SOTA methods CAML and DRCAML with health-related text pre-trained embeddings T100SG and T300SG perform better across all levels of ICD-9 codes. Chapter 6 focuses on the effectiveness of transformers. Domain-specific pre-trained transformer models show benefits over general text

pre-trained transformers among the same type of models. However, the limitation of maximum sequence length for standard transformers impacts predictive accuracy for data such as MIMIC-III with long documents. It is shown that for longer documents, such as the MIMIC-III data and multi-label problems with more labels, standard transformer models with restricted text sequence length (i.e. 512 tokens) are not as good as the traditional word-embeddings-based SOTA networks. Using TransformerXL and Longformer overcomes the limitation of sequence length, with TransformerXL (3,072) outperforming all other networks. TransformerXL (3,072) requires considerable computational time, and if processing time is not an issue, then TransformerXL with a more extended sequence will be the best model to use.

The second point is mentioned throughout Chapters 5 and 6, and is addressed in Chapter 7. TransformerXL and Longformer can handle long documents and outperform SOTA for predicting medical codes from EHRs. However, the tail-end performance across many language models is poor. Chapter 7 presents concatenated language models for domain-specific transformers. Experimental results show that multi-BioMed-Transformers outperform other language models, including TransformerXL and Longformer, when F1 scores of tail-end labels are compared directly for multi-label problems with long-tail labels. In addition, results also show improvements in overall micro and macro F1 scores and are achieved with fewer resources.

8.1 Future work

The research presented in this thesis opens directions for future research. This section discusses some avenues and possible extensions to the research presented in this thesis.

8.1.1 Domain-specific continuous training of TransformerXL and Longformer

Given the evidence presented in this thesis regarding the benefits of domain-specific language models, continuous training of TransformerXL and Longformer are the most obvious next steps. The training would initialize with the standard TransformerXL and Longformer model, pre-trained using general text. It then continues the pre-training process using domain-specific data. Alternatively, TransformerXL and Longformer models can be trained from scratch using domain-specific data. Domain-specific TransformerXL and Longformer can also be used as part of concatenated models, as presented in Chapter 7.

8.1.2 Hierarchical Multi-label Classifications

Medical codes such as ICD-9 codes have a tree hierarchy nature. The most obvious extension to the research presented in this thesis is to consider predicting ICD-9 codes as a hierarchical multi-label classification problem. The information of ancestor classes could be incorporated for each label. This will also be possible for the case studies, as each label belongs to a parent class and ancestor class as shown in Figures 2.9, 2.10 and 2.11 for cardiovascular disease, COVID-19 patient shielding and systemic fungal or bacterial infections, respectively. Moreover, using transformers in a hierarchical multi-label setting is a relatively new area [150]. Hence, the benefits of incorporating the advancements of transformers in the hierarchical setting is an open and exciting research avenue to explore.

8.1.3 HAN with Transformers and Word Embeddings

This thesis has outlined the benefits and limitations of both domain-specific word embeddings and transformer models. An alternative avenue is to combine both transformers and word embeddings. Based on the original HAN [3] archi-

ture and the recent modification presented by Si and Roberts (2021) [129], an alternative is to use domain-specific transformer-based encoders to learn features from words to sentences and use word embeddings to encode sentences to build the document representations. This means at the first level from word to sentence a domain-specific pre-trained transformer model is directly applied, and aggregate the representation of the words to form a sentence vector. The number of tokens in any given sentence is generally a lot less than 512 tokens and hence provides an alternative to overcoming the limitations of domain-specific transformers.

8.2 Final Remarks

This thesis presents an extensive research study on domain-specific language models for multi-label classification of medical text; where predicting ICD-9 codes from EHRs is the primary application. The research shows that domain-specific multi-sourced fastText pre-trained embeddings improve predictive accuracy; and that transformer models, such as TransformerX, outperform SOTA results for predicting medical codes from EHRs. In addition, this research explored ways to improve tail-end label predictions for multi-label classification. Whereas multi-label classifications generally focus on improving overall predictive accuracy, this leads to most frequent labels having greater influence on model inference. For EHRs, however, accuracy for low-frequency labels is of considerable importance; and this research has shown how concatenated models yield improvement to tail-end label predictions for multi-label classification.

Appendix A

Additional Data Information

Patient ID	Admission
242070	admission diagnosis Non-operative Organ Systems Organ System Cardiovascular admission diagnosis Was the patient admitted from the O.R. or went to the O.R. within 4 hours of admission? No admission diagnosis All Diagnosis Non-operative Diagnosis Cardiovascular Rhythm disturbance (conduction defect)
Patient ID	Care General
242070	Proton pump inhibitor
242070	Combined device and drug therapy
242070	Not intubated/normal airway
242070	Parenteral - bolus pm
242070	Spontaneous - adequate
242070	H-2 antagonists
242070	Full therapy
242070	Intermediate
242070	NPO
Patient ID	History
242070	peripheral vascular disease
242070	Performed
242070	procedural coronary intervention - date unknown
242070	hypertension requiring treatment
Patient ID	Patient
242070	Rhythm disturbance (conduction defect)
Patient ID	Treatment
242070	gastrointestinal medications antiemetic serotonin antagonist ondansetron
242070	cardiovascular consultations Cardiology consultation
242070	cardiovascular vascular disorders VTE prophylaxis compression boots
242070	cardiovascular vascular disorders VTE prophylaxis conventional heparin therapy
242070	gastrointestinal medications stress ulcer prophylaxis esomeprazole
242070	neurologic pain / agitation / altered mentation analgesics
242070	gastrointestinal endoscopy/gastric instrumentation nasogastric tube with suction
242070	cardiovascular vascular disorders VTE prophylaxis compression stockings
242070	pulmonary ventilation and oxygenation oxygen therapy (< 40%) nasal cannula
242070	renal urinary catheters foley catheter

Figure A.1: Sample eICU Data as obtained from the Database.

Appendix B

Additional Results for Binary Classification

ICD-9 groups	Models	LR, Ridge value				
		1e-8	1e-4	1e-2	1	100
circ (78.4%)	W300	0.932	0.932	0.932	0.932	0.933
	M300	0.931	0.931	0.931	0.931	0.931
	T300	0.933	0.933	0.933	0.933	0.933
	TM300	0.932	0.932	0.932	0.932	0.932
resp (46.6%)	W300	0.774	0.774	0.774	0.774	0.774
	M300	0.771	0.771	0.771	0.771	0.768
	T300	0.779	0.779	0.779	0.779	0.778
	TM300	0.776	0.776	0.776	0.776	0.774
skin (12.0%)	W300	0.336	0.336	0.336	0.335	0.303
	M300	0.347	0.347	0.347	0.345	0.281
	T300	0.350	0.350	0.349	0.349	0.311
	TM300	0.344	0.344	0.344	0.342	0.295

Table B.1: F-measure for selected ICD-9 groups for word embeddings with varied dimensions using LR with varied ridge values for MIMIC III data is presented. Refer to Table 4.3 for word embeddings models details. 10-fold cross validation was used.

Random Forest parameters			W300	M300	T300	TM300
depth = 50	I = 100	K = 0	0.915	0.918	0.917	0.918
depth = 200	I = 100	K = 0	0.915	0.918	0.917	0.918
depth = 0	I = 100	K = 0	0.914	0.917	0.915	0.916
depth = 0	I = 100	K = 20	0.916	0.920	0.918	0.918
depth = 0	I = 100	K = 50	0.918	0.920	0.919	0.919
depth = 0	I = 300	K = 0	0.915	0.918	0.917	0.918
depth = 0	I = 300	K = 20	0.916	0.920	0.918	0.909
depth = 0	I = 300	K = 50	0.918	0.921	0.919	0.919
depth = 0	I = 600	K = 0	0.914	0.917	0.915	0.916
depth = 0	I = 600	K = 50	0.916	0.920	0.918	0.918

Table B.2: F-measure for circ(78.9%) for word embeddings using random forest for MIMIC III data is presented. Refer to Table 5.5 for word embeddings models details. 10-fold cross validation was used.

Random Forest parameters			W300	M300	T300	TM300
depth = 50	I = 100	K = 0	0.712	0.711	0.703	0.703
depth = 200	I = 100	K = 0	0.712	0.711	0.703	0.703
depth = 0	I = 100	K = 0	0.707	0.709	0.701	0.701
depth = 0	I = 100	K = 20	0.715	0.713	0.703	0.708
depth = 0	I = 100	K = 50	0.715	0.718	0.703	0.707
depth = 0	I = 300	K = 0	0.712	0.711	0.703	0.703
depth = 0	I = 300	K = 20	0.720	0.716	0.708	0.711
depth = 0	I = 300	K = 50	0.718	0.720	0.709	0.710
depth = 0	I = 600	K = 0	0.707	0.709	0.701	0.701
depth = 0	I = 600	K = 50	0.715	0.713	0.703	0.708

Table B.3: F-measure for resp(46.6%) for word embeddings using random forest for MIMIC III data is presented. Refer to Table 5.5 for word embeddings models details. 10-fold cross validation was used.

Random Forest parameters			W300	M300	T300	TM300
depth = 50	I = 100	K = 0	0.003	0.021	0.009	0.011
depth = 200	I = 100	K = 0	0.003	0.021	0.009	0.011
depth = 0	I = 100	K = 0	0.004	0.013	0.006	0.006
depth = 0	I = 100	K = 20	0.010	0.039	0.017	0.022
depth = 0	I = 100	K = 50	0.020	0.053	0.033	0.029
depth = 0	I = 300	K = 0	0.003	0.021	0.009	0.011
depth = 0	I = 300	K = 20	0.008	0.031	0.014	0.013
depth = 0	I = 300	K = 50	0.015	0.047	0.030	0.027
depth = 0	I = 600	K = 0	0.04	0.013	0.006	0.006
depth = 0	I = 600	K = 50	0.010	0.039	0.017	0.022

Table B.4: F-measure for skin(12.0%) for word embeddings using random forest for MIMIC III data is presented. Refer to Table 4.3 for word embeddings models details. 10-fold cross validation was used.

Appendix C

Additional Results for Multi-label Classification

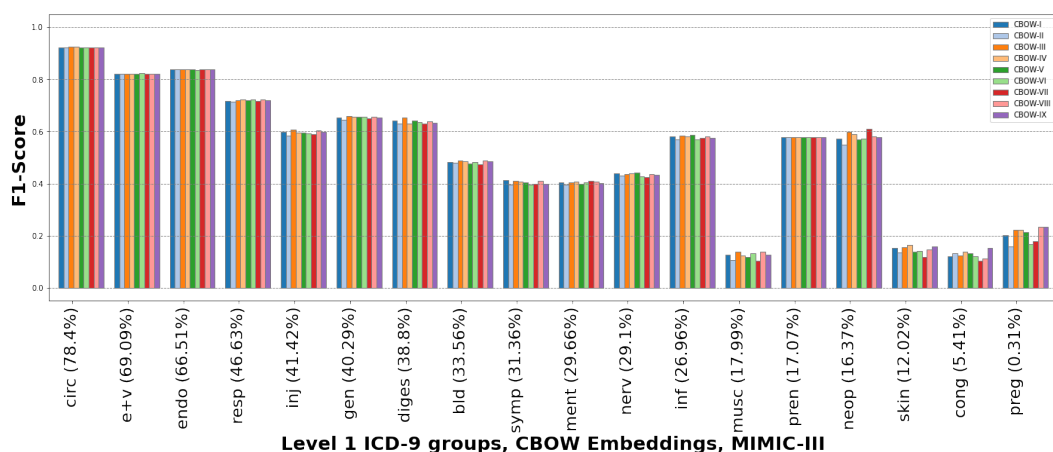


Figure C.1: Comparison of F1-scores for top level ICD-9 groups for embeddings obtained by varying fastText parameters for CBOW model, as presented in Table 5.4.

Tokens	Filters	Filter Size	Micro-F1	Macro-F1
2,500	100	[1,2,3,4]	0.81	0.73
2,500	300	[2,4,6,8]	0.81	0.73
2,500	600	[2,4,6,8]	0.81	0.74
2,500	600	[4,6,8,10,12]	0.81	0.74
5,000	300	[2,4,6,8]	0.82	0.76

Table C.1: Overall performance of top level ICD-9 groupings, 50 dimensional embeddings for varied parameters of CNNTText.

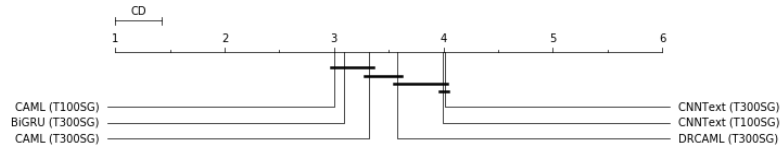


Figure C.2: Critical difference plots for Level 3 ICD-9 codes, 316 labels. Nemenyi post-hoc test (95% confidence level), identifying statistical differences between multi-label neural networks presented in Table 5.9 for eICU data.

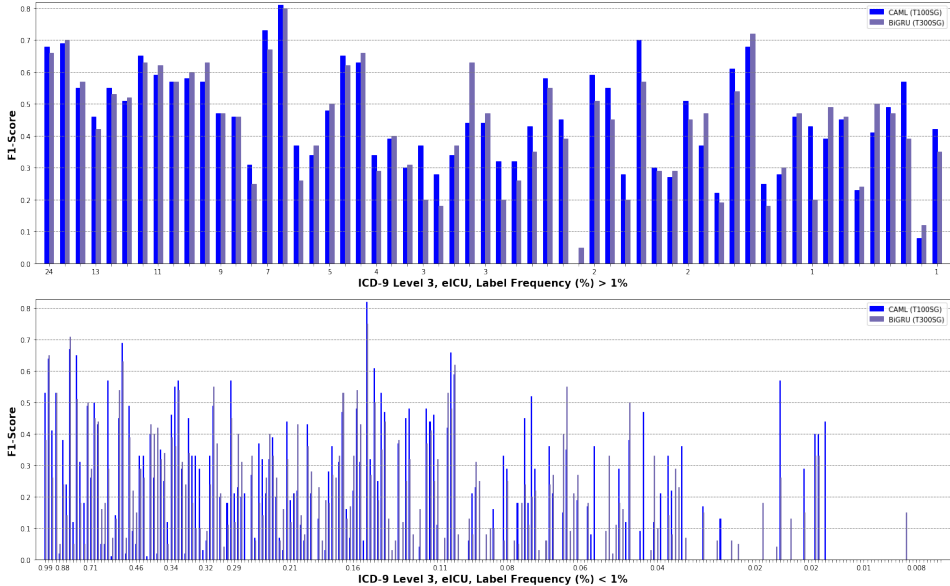


Figure C.3: Comparison of individual label F1-score of ICD-9 level 3 between BiGRU (T300SG) and CAML (T100SG) embeddings.



Figure C.4: Comparison of individual label F1-score of ICD-9 level 3 between CNNTxt and CAML with T100SG embeddings.

Appendix D

Additional Results for Multi-label Classification using Transformers

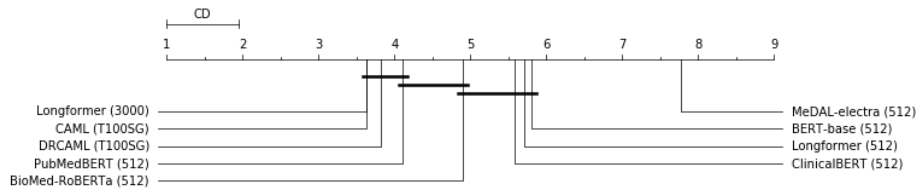


Figure D.1: Critical difference plots for Level 2 ICD-9 codes, MIMIC-III data, 158 labels. Nemenyi post-hoc test (95% confidence level), identifying statistical differences between multi-label transformers and embeddings-based neural networks presented in Table 6.5 for MIMIC-III data.

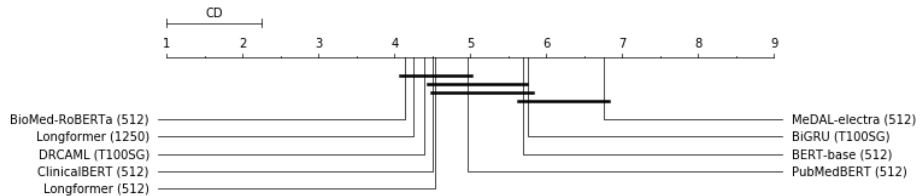


Figure D.2: Critical difference plots for Level 2 ICD-9 codes, eICU data, 93 labels. Nemenyi post-hoc test (95% confidence level) identifying statistical differences between multi-label transformers and embeddings-based neural networks presented in Table 6.6 for eICU data.

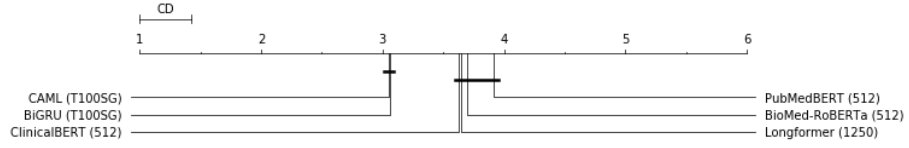


Figure D.3: Critical difference plots for Level 3 ICD-9 codes, 316 labels. Nemenyi post-hoc test (95% confidence level) identifying statistical differences between multi-label neural networks presented in Table 6.6 for eICU data.



Figure D.4: Comparison of individual label F1-score of ICD-9 level 3 between Longformer and ClinicalBERT.

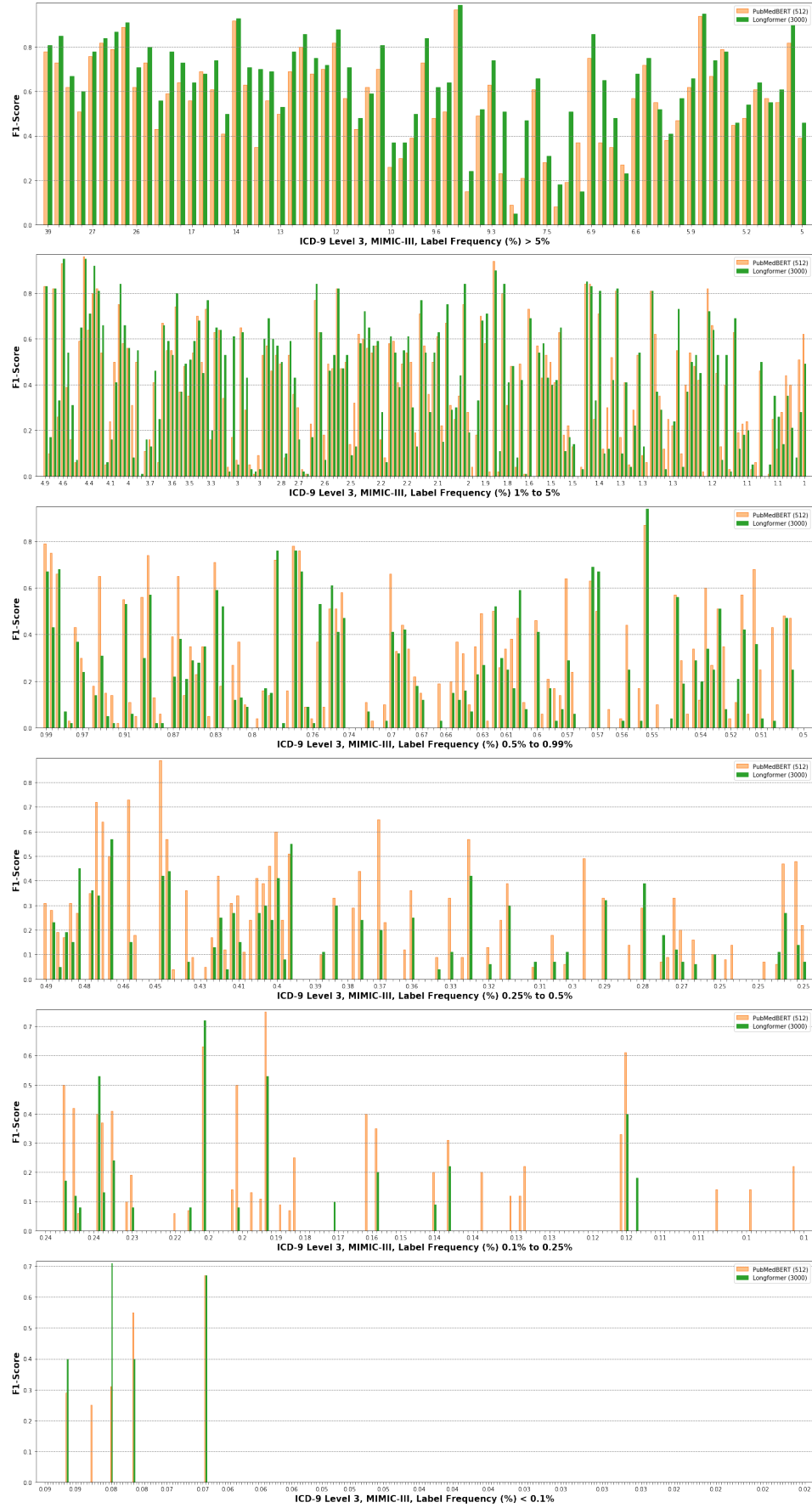


Figure D.5: Comparison of individual label F1-score of ICD-9 level 3 between PubMedBERT and Longformer.

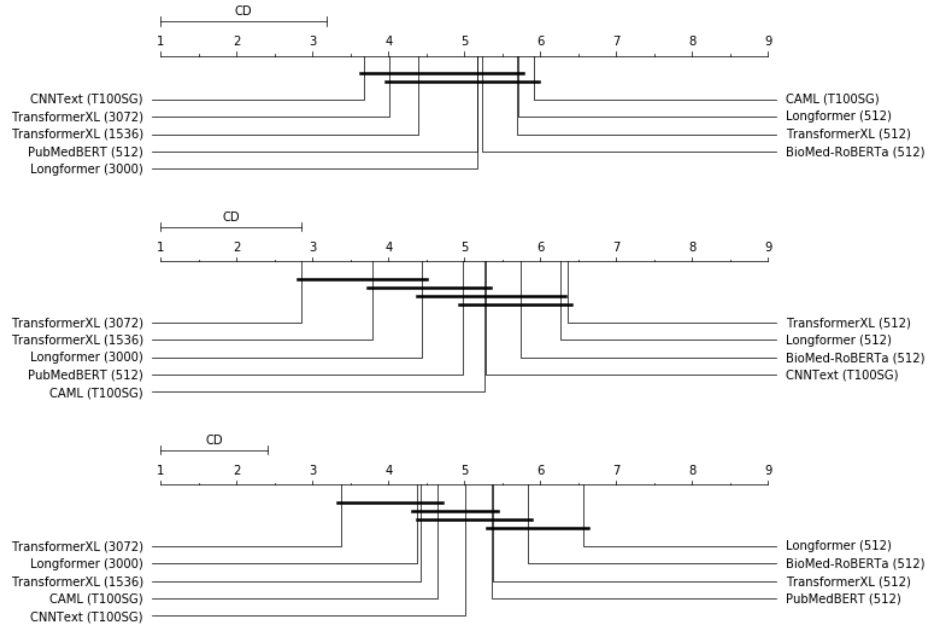


Figure D.6: Critical difference plots for case studies - cardiovascular disease (top); COVID-19 (middle); and fungal or bacterial infection (bottom), for MIMIC-III data using transformers and embeddings-based neural networks. Nemenyi post-hoc test (95% confidence level), identifying statistical differences between label F1 scores of multi-label neural networks presented in Table 6.7 for MIMIC-III data.

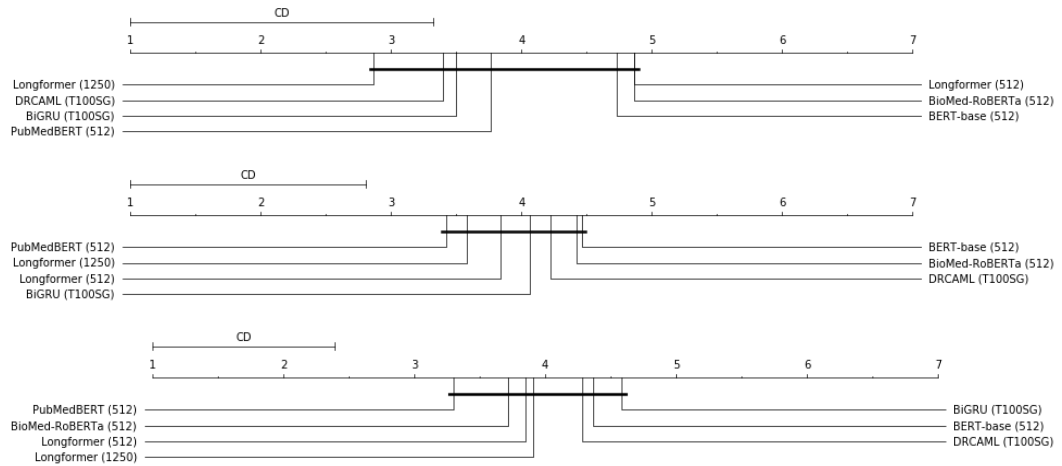


Figure D.7: Critical difference plots for case studies - cardiovascular disease (top); COVID-19 (middle); and fungal or bacterial infection (bottom), for eICU data using transformers and embeddings-based neural networks. Nemenyi post-hoc test (95% confidence level), identifying statistical differences between label F1 scores of multi-label neural networks presented in Table 6.8 for eICU data.

Appendix E

Additional Results for Language Model Variations

Neural Network Details	Input	Micro-F1	Macro-F1
DualCNNTText	dis 512 + dis 512	0.67	0.19
TripleCNNTText	dis 512 + dis 512 + ecg 512	0.67	0.20
TripleCNNTText	dis 512 + dis 512 + rad 512	0.68	0.19
QuadrupleCNNTText	dis 512 + dis 512 + ecg 512 + rad 512	0.68	0.21

Table E.1: Additional results for Cardiovascular disease, MIMIC-III data.

Nueral Network Details	Tokens	Micro-F1	Macro-F1
CAML (T100SG)	1,250	0.68	0.41
BiGRU (T100SG)	1,250	0.67	0.43
PubMedBERT	256	0.67	0.41
PubMedBERT	512	0.67	0.41
Longformer	1,250	0.69	0.42
DualPubMedBERT	2*128	0.60	0.35
DoublePubMedBERT	2*256	0.62	0.37
TriplePubMedBERT	3*128	0.62	0.34
TriplePubMedBEERT	3*256	0.62	0.35
QuadruplePubMedBERT	4*128	0.66	0.36

Table E.2: Cardiovascular disease, eICU data. Number of instances are 53,477 and number of labels are 15.

Neural Network	Embeddings	Tokens	Micro-F1	Macro-F1
CNNText	T100SG	512	0.41	0.28
CNNText	T300SG	3000	0.57	0.37
CNNText	T600SG	3000	0.57	0.36
DualCNNText	T100SG	1024	0.45	0.21
TripleCNNText	T100SG	1536	0.48	0.22
QuadrupleCNNText	T100SG	2048	0.43	0.22

Table E.3: Additional results for systemic fungal or bacterial infection, MIMIC-III data.

Nueral Network Details	Max tokens	Micro-F1	Macro-F1
BiGRU (T100SG)	1250	0.62	0.27
DR-CAML (T100SG)	1250	0.63	0.25
Longformer	1250	0.62	0.29
PubMedBERT	128	0.58	0.25
PubMedBERT	256	0.61	0.30
PubMedBERT	512	0.63	0.30
DualPubMedBERT	2*128	0.58	0.24
TriplePubMedBERT	3*128	0.60	0.24
QuadruplePubMedBERT	4*128	0.58	0.24
DualPubMedBERT	2*256	0.58	0.26
TriplePubMedBEERT	3*256	0.58	0.24

Table E.4: Systemic fungal or bacterial infection, eICU data. Number of instances are 54,193 and number of labels are 42.

References

- [1] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751. Association for Computational Linguistics, 2014.
- [2] James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. Explainable prediction of medical codes from clinical text. *arXiv preprint arXiv:1802.05695*, 2018.
- [3] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489, 2016.
- [4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30:5998–6008, 2017.
- [5] Hercules Dalianis. *Clinical Text Mining: Secondary Use of Electronic Patient Records*. Springer, 2018.
- [6] Wencheng Sun, Zhiping Cai, Yangyang Li, Fang Liu, Shengqun Fang, and Guoyan Wang. Data processing and text mining technologies on electronic medical records: a review. *Journal of healthcare engineering*, 2018, 2018.
- [7] Jake Luo, Min Wu, Deepika Gopukumar, and Yiqing Zhao. Big data application in biomedical research and health care: a literature review. *Biomedical informatics insights*, 8:BII–S31559, 2016.
- [8] Wei Ruan, Naveenkumar Appasani, Katherine Kim, Joseph Vincelli, Hyun Kim, and Won-Sook Lee. Pictorial Visualization of EMR Summary Interface and Medical Information Extraction of Clinical Notes.

- In *2018 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, pages 1–6. IEEE, 2018.
- [9] S. Garfinkel. De-identification of personally identifiable information technical report. *National institute of Standards and Technology (NIST), U.S. Department of Commerce*, 2015.
 - [10] Franck Dernoncourt, Ji Young Lee, Özlem Uzuner, and Peter Szolovits. De-identification of patient notes with recurrent neural networks. *Journal of the American Medical Informatics Association*, 24(3):596–606, 2017.
 - [11] John Mark Michael Rumbold and Barbara Pierscioneck. The effect of the general data protection regulation on medical research. *Journal of medical Internet research*, 19(2), 2017.
 - [12] Gauthier Chassang. The impact of the EU general data protection regulation on scientific research. *Ecancer medical science*, 11, 2017.
 - [13] Office of the Privacy Commissioner. Health Information Privacy Code 1994. <https://www.privacy.org.nz/the-privacy-act-and-codes/codes-of-practice/health-information-privacy-code-1994/>, 2013. Original date: 2013; Accessed: December 09, 2017.
 - [14] Health & Disability Commissioner. The Code of Rights, Health & Disability Commissioner. <http://www.hdc.org.nz/the-act-code/the-code-of-rights>, 2009. Original date: 2009; Accessed: December 09, 2017.
 - [15] Henning Schäfer and Christoph Friedrich. Multilingual ICD-10 Code Assignment with Transformer Architectures using MIMIC-III Discharge Summaries. In *Proceedings of Lab Working Notes of CLEF 2020, Thessaloniki, Greece*, 2020.
 - [16] Ali Amin-Nejad, Julia Ive, and Sumithra Velupillai. Exploring Transformer Text Generation for Medical Dataset Augmentation. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4699–4708, 2020.
 - [17] Biplob Biswas, Thai-Hoang Pham, and Ping Zhang. TransICD: Transformer Based Code-wise Attention Model for Explainable ICD Coding. *arXiv preprint arXiv:2104.10652*, 2021.
 - [18] Guergana K Savova, James J Masanz, Philip V Ogren, Jiaping Zheng, Sunghwan Sohn, Karin C Kipper-Schuler, and Christopher G

- Chute. Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications. *Journal of the American Medical Informatics Association*, 17(5):507–513, 2010.
- [19] Vijay Garla, Vincent Lo Re III, Zachariah Dorey-Stein, Farah Kidwai, Matthew Scotch, Julie Womack, Amy Justice, and Cynthia Brandt. The Yale cTAKES extensions for document classification: architecture and application. *Journal of the American Medical Informatics Association*, 18(5):614–620, 2011.
- [20] Hongfang Liu, Kavishwar B Waghlikar, Siddhartha Jonnalagadda, and Sunghwan Sohn. Integrated cTAKES for Concept Mention Detection and Normalization. In *CLEF (Working Notes)*, 2013.
- [21] Ruth Reátegui and Sylvie Ratté. Comparison of MetaMap and cTAKES for entity extraction in clinical notes. *BMC Medical Informatics and Decision Making*, 18(3):74, 2018.
- [22] Hua Yang and Teresa Gonçalves. UEvora at CLEF eHealth 2017 Task 3. In *CLEF (Working Notes)*, 2017.
- [23] A. N. Jagannatha and H. Yu. Bidirectional RNN for medical event detection in electronic health records. *Association for Computational Linguistics. North American Chapter. Meeting*, page 473–482, 2016.
- [24] Chang Wang, Liangliang Cao, and Bowen Zhou. Medical synonym extraction with concept space models. *CoRR*, abs/1506.00528, 2015.
- [25] José Antonio Miñarro-Giménez, Oscar Marín-Alonso, and Matthias Samwald. Exploring the application of deep learning techniques on medical text corpora. 205:584–588, 08 2014.
- [26] E. Choi, A. Schuetz, W. F. Stewart, and J Sun. Using recurrent neural network models for early detection of heart failure onset. *Journal of the American Medical Informatics Association : JAMIA*, 24(2):361–370, 2017.
- [27] Alistair EW Johnson, Tom J Pollard, and Roger G Mark. Reproducibility in critical care: a mortality prediction case study. In *Machine Learning for Healthcare Conference*, pages 361–376, 2017.
- [28] Vithya Yogarajan, Henry Gouk, Tony Smith, Mike Mayo, and Bernhard Pfahringer. Comparing High Dimensional Word Embeddings Trained on

- Medical Text to Bag-of-Words For Predicting Medical Codes. *Proceedings of the Asian Conference on Intelligent Information and Database Systems (ACIIDS 2020)*. In N. T. Nguyen et al. (Eds.), *Lecture Notes on Artificial Intelligence (LNAI)*, Springer Nature., 12033:1–12, 2020.
- [29] Vithya Yogarajan, Jacob Montiel, Tony Smith, and Bernhard Pfahringer. Seeing the whole patient: Using multi-label medical text classification techniques to enhance predictions of medical codes. *arXiv preprint arXiv:2004.00430*, 2020.
- [30] Vithya Yogarajan, Jacob Montiel, Tony Smith, and Bernhard Pfahringer. Transformers for multi-label classification of medical text: An empirical comparison. In Allan Tucker, Pedro Henriques Abreu, Jaime Cardoso, Pedro Pereira Rodrigues, and David Riaño, editors, *Artificial Intelligence in Medicine*, pages 114–123, Cham, 2021. Springer International Publishing.
- [31] Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Liwei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. MIMIC-III, a freely accessible critical care database. *Scientific data*, 3:160035, 2016.
- [32] Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000.
- [33] MIT Critical Data. *Secondary Analysis of Electronic Health Records*. Springer, 2016.
- [34] Tom J. Pollard, Alistair E. W. Johnson, Jesse D. Raffa, Leo A. Celi, Roger G. Mark, and Omar Badawi. The eICU Collaborative Research Database, a freely available multi-center database for critical care research. *Scientific Data*, 5:180178, September 2018.
- [35] Kirk Roberts, Dina Demner-Fushman, Ellen M Voorhees, William R Hersh, Steven Bedrick, Alexander J Lazar, and Shubham Pant. Overview of the trec 2017 precision medicine track. *NIST Special Publication*, pages 500–324, 2017.
- [36] Peter B. Jensen, Lars J. Jensen, and Søren Brunak. Mining electronic health records: towards better research applications and clinical care. *Nature Reviews Genetics*, 13(6):395, 2012.

- [37] Haoran Shi, Pengtao Xie, Zhiting Hu, Ming Zhang, and Eric P Xing. Towards automated ICD coding using deep learning. *arXiv preprint arXiv:1711.04075*, 2017.
- [38] World Health Organization. Cardiovascular Diseases, [https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-\(cvds\)](https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)), Last accessed 11 January 2021.
- [39] John PA Ioannidis. Precision shielding for covid-19: metrics of assessment and feasibility of deployment. *BMJ Global Health*, 6(1):e004614, 2021.
- [40] Elizabeth Reilly, Sarah Skeoch, Sarah Hardcastle, John D Pauling, Megan Rowe, Tehseen Ahmed, Andrew Allard, Bashaar Boyce, Ellie Korendowych, Chloe Lapraik, et al. Evaluation of a patient self-stratification methodology to identify those in need of shielding during covid-19. *Clinical Medicine*, 20(6):e212, 2020.
- [41] Getachew Alemkere, Admasu Tenna, and Ephrem Engidawork. Antibiotic use practice and predictors of hospital outcome among patients with systemic bacterial infection: Identifying targets for antibiotic and health care resource stewardship. *PloS one*, 14(2):e0212661, 2019.
- [42] Clinton K Murray, Florence L Loo, Duane R Hospenthal, Leopoldo C Cancio, John A Jones, Seung H Kim, John B Holcomb, Charles E Wade, and Steven E Wolf. Incidence of systemic fungal infection and related mortality following severe burns. *Burns*, 34(8):1108–1112, 2008.
- [43] SK Bharwani and R Dhanireddy. Systemic fungal infection is associated with the development of retinopathy of prematurity in very low birth weight infants: a meta-review. *Journal of Perinatology*, 28(1):61–66, 2008.
- [44] Sema Demirçin, Fatih Dirol, Yaşar Mustafa Karagöz, Aylin Aksoy, Derya Bulgur Kırbaş, and Pelin Göral. Velvet heart: Systemic fungal infection accompanying with cardiac involvement and review of the literature. *Adli Tıp Bülteni*, 22(2):134–139, 2017.
- [45] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. Mining multi-label data. In *Data mining and knowledge discovery handbook*, pages 667–685. Springer, 2009.
- [46] Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)*, 3(3):1–13, 2007.

- [47] Yoav Goldberg. Neural network methods for natural language processing. *Synthesis Lectures on Human Language Technologies*, 10(1):1–309, 2017.
- [48] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.
- [49] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.
- [50] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.
- [51] Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, H erve J egou, and Tomas Mikolov. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*, 2016.
- [52] Zellig S. Harris. Distributional structure. *WORD*, 10(2-3):146–162, 1954.
- [53] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
- [54] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.
- [55] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.
- [56] D Cox. The regression analysis of binary sequences. *Journal of the Royal Statistical Society. Series B (Methodological)*, 20(2):215–242, 1958.
- [57] S. le Cessie and J.C. van Houwelingen. Ridge estimators in logistic regression. *Applied Statistics*, 41(1):191–201, 1992.
- [58] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
- [59] George H. John and Pat Langley. Estimating continuous distributions in bayesian classifiers. In *Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 338–345, San Mateo, 1995. Morgan Kaufmann.

- [60] Shantanu Godbole and Sunita Sarawagi. Discriminative methods for multi-labeled classification. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 22–30. Springer, 2004.
- [61] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 254–269. Springer, 2009.
- [62] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification. *Machine learning*, 85(3):333, 2011.
- [63] Min-Ling Zhang and Zhi-Hua Zhou. A k-nearest neighbor based algorithm for multi-label classification. In *2005 IEEE international conference on granular computing*, volume 2, pages 718–721. IEEE, 2005.
- [64] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.
- [65] W. Finnoff, F. Hergert, and H. G. Zimmermann. Extended regularization methods for nonconvergent model selection. In S. Hanson, J. Cowan, and C. Giles, editors, *Advances in Neural Information Processing Systems*, volume 5. Morgan-Kaufmann, 1993.
- [66] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.
- [67] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.
- [68] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [69] Kyunghyun Cho, B van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. In *Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-8), 2014*, 2014.
- [70] Elias Moons, Aditya Khanna, Abbas Akkasi, and Marie-Francine Moens. A comparison of deep learning methods for ICD coding of clinical records. *Applied Sciences*, 10(15):5262, 2020.

- [71] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.
- [72] Dzmitry Bahdanau, KyungHyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *International Conference on Learning Representations (ICLR)*, 2015.
- [73] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In *ACL*, 2019.
- [74] Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. Domain-specific language model pretraining for biomedical natural language processing. *arXiv preprint arXiv:2007.15779*, 2020.
- [75] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*, 2019.
- [76] Mohammad Taher Pilehvar and Jose Camacho-Collados. Embeddings in natural language processing: Theory and advances in vector representations of meaning. *Synthesis Lectures on Human Language Technologies*, 13(4):1–175, 2020.
- [77] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*, 2020.
- [78] Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, 2019.
- [79] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.
- [80] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don’t Stop Pretraining: Adapt Language Models to Domains and Tasks. In *Proceedings of ACL*, 2020.

- [81] Zhi Wen, Xing Han Lu, and Siva Reddy. MeDAL: Medical abbreviation disambiguation dataset for natural language understanding pretraining. In *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, pages 130–135, Online, November 2020. Association for Computational Linguistics.
- [82] I Beltagy, ME Peters, and A Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.
- [83] Mike Schuster and Kaisuke Nakajima. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152, 2012.
- [84] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *In International Conference on Learning Representations (ICLR).*, 2019.
- [85] David R Cox. The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, 20(2):215–232, 1958.
- [86] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I.H. Witten. The WEKA Data Mining Software: An Update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.
- [87] I.H. Witten, E. Frank, M. Hall, and C.J. Pal. *Data Mining: Practical Machine Learning Tools and Techniques, 4th Edition*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2016.
- [88] Jesse Read, Peter Reutemann, Bernhard Pfahringer, and Geoff Holmes. MEKA: A Multi-label/Multi-target Extension to WEKA. *Journal of Machine Learning Research*, 17(21):1–5, 2016.
- [89] Min-Ling Zhang and Zhi-Hua Zhou. A review on multi-label learning algorithms. *IEEE transactions on knowledge and data engineering*, 26(8):1819–1837, 2013.
- [90] Robert E Schapire and Yoram Singer. Boostexter: A boosting-based system for text categorization. *Machine learning*, 39(2):135–168, 2000.
- [91] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, December 2006.

- [92] Salvador Garcia and Francisco Herrera. An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. *Journal of Machine Learning Research*, 9:2677–2694, 2009.
- [93] André CPLF de Carvalho and Alex A Freitas. A tutorial on multi-label classification techniques. *Foundations of computational intelligence volume 5*, pages 177–195, 2009.
- [94] Gjorgji Madjarov, Dragi Kocev, Dejan Gjorgjevikj, and Sašo Džeroski. An extensive experimental comparison of methods for multi-label learning. *Pattern recognition*, 45(9):3084–3104, 2012.
- [95] Haibo He and Eduardo A Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009.
- [96] Andrew Mayne and Russell Perry. Hierarchically classifying documents with multiple labels. In *2009 IEEE Symposium on Computational Intelligence and Data Mining*, pages 133–139. IEEE, 2009.
- [97] Mongkud Klungpornkun and Peerapon Vateekul. Hierarchical text categorization using level based neural networks of word embedding sequences with sharing layer information. *Walailak Journal of Science and Technology (WJST)*, 16(2):121–131, 2019.
- [98] Carlos N Silla and Alex A Freitas. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22(1-2):31–72, 2011.
- [99] Amber Stubbs, Christopher Kotfila, Hua Xu, and Özlem Uzuner. Identifying risk factors for heart disease over time: Overview of 2014 i2b2/UTHealth shared task track 2. *Journal of Biomedical Informatics*, 58(Supplement):S67 – S77, 2015.
- [100] Seyedmostafa Sheikhalishahi, Riccardo Miotto, Joel T Dudley, Alberto Lavelli, Fabio Rinaldi, and Venet Osmani. Natural language processing of clinical notes on chronic diseases: Systematic review. *JMIR medical informatics*, 7(2):e12239, 2019.
- [101] Duy Duc An Bui, Mathew Wyatt, and James J. Cimino. The UAB informatics institute and 2016 CEGS N-GRID de-identification shared task challenge. *Journal of Biomedical Informatics*, 2017.
- [102] Hee-Jin Lee, Yonghui Wu, Yaoyun Zhang, Jun Xu, Hua Xu, and Kirk Roberts. A hybrid approach to automatic de-identification of psychiatric notes. *Journal of Biomedical Informatics*, 2017.

- [103] Hui Yang and Jonathan M. Garibaldi. Automatic detection of protected health information from clinic narratives. *Journal of Biomedical Informatics*, 58(Supplement):S30 – S38, 2015.
- [104] Zengjian Liu, Buzhou Tang, Xiaolong Wang, and Qingcai Chen. De-identification of clinical notes via recurrent neural network and conditional random field. *Journal of Biomedical Informatics*, 2017.
- [105] Carol Friedman, George Hripcsak, et al. Natural language processing and its future in medicine. *Academic Medicine*, 74(8):890–5, 1999.
- [106] Peter J Haug, David L Ranum, and Philip R Frederick. Computerized extraction of coded findings from free-text radiologic reports. work in progress. *Radiology*, 174(2):543–548, 1990.
- [107] Peter J Haug, Spence Koehler, Lee Min Lau, Ping Wang, Roberto Rocha, and Stanley M Huff. Experience with a mixed semantic/syntactic parser. In *Proceedings of the Annual Symposium on Computer Application in Medical Care*, page 284. American Medical Informatics Association, 1995.
- [108] Anthony N Nguyen, Julie Moore, John O’Dwyer, and Shoni Philpot. Assessing the utility of automatic cancer registry notifications data extraction from free-text pathology reports. In *AMIA Annual Symposium Proceedings*, volume 2015, page 953. American Medical Informatics Association, 2015.
- [109] Stéphane M Meystre, Guergana K Savova, Karin C Kipper-Schuler, and John F Hurdle. Extracting information from textual documents in the electronic health record: a review of recent research. *Yearbook of Medical Informatics*, 17(01):128–144, 2008.
- [110] Yonghui Wu, Joshua C Denny, S Trent Rosenbloom, Randolph A Miller, Dario A Giuse, and Hua Xu. A comparative study of current clinical natural language processing systems on handling abbreviations in discharge summaries. In *AMIA annual symposium proceedings*, volume 2012, page 997. American Medical Informatics Association, 2012.
- [111] Alan R Aronson and François-Michel Lang. An overview of MetaMap: historical perspective and recent advances. *Journal of the American Medical Informatics Association*, 17(3):229–236, 2010.
- [112] Zhengping Che, David Kale, Wenzhe Li, Mohammad Taha Bahadori, and Yan Liu. Deep computational phenotyping. In *Proceedings of the*

21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '15, pages 507–516, New York, NY, USA, 2015. ACM.

- [113] Zhengping Che, David Kale, Wenzhe Li, Mohammad Taha Bahadori, and Yan Liu. Deep computational phenotyping. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 507–516. ACM, 2015.
- [114] Zhengping Che, Sanjay Purushotham, Robinder Khemani, and Yan Liu. Distilling knowledge from deep networks with applications to healthcare domain. *arXiv preprint arXiv:1512.03542*, 2015.
- [115] Sanjay Purushotham, Chuizheng Meng, Zhengping Che, and Yan Liu. Benchmark of deep learning models on large healthcare mimic datasets. *arXiv preprint arXiv:1710.08531*, 2017.
- [116] Lung-Hao Lee, Yi Lu, Po-Han Chen, Po-Lei Lee, and Kuo-Kai Shyu. NCUEE at MEDIQA 2019: Medical Text Inference Using Ensemble BERT-BiLSTM-Attention Model. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 528–532, 2019.
- [117] Asma Ben Abacha, Chaitanya Shivade, and Dina Demner-Fushman. Overview of the MEDIQA 2019 shared task on textual inference, question entailment and question answering. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 370–379, 2019.
- [118] Y. Choi, C. Y.-I. Chiu, and D Sontag. Learning low-dimensional representations of medical concepts. *AMIA Summits on Translational Science Proceedings*, page 41–50, 2016.
- [119] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.
- [120] Qingyu Chen, Yifan Peng, and Zhiyong Lu. Biosentvec: creating sentence embeddings for biomedical texts. *The 7th IEEE International Conference on Healthcare Informatics*, 2019.
- [121] Yijia Zhang, Qingyu Chen, Zhihao Yang, Hongfei Lin, and Zhiyong Lu. BioWordVec, improving biomedical word embeddings with subword information and MeSH. *Scientific data*, 6(1):52, 2019.

- [122] Andrew L Beam, Benjamin Kompa, Inbar Fried, Nathan P Palmer, Xu Shi, Tianxi Cai, and Isaac S Kohane. Clinical concept embeddings learned from massive sources of multimodal medical data. *arXiv preprint arXiv:1804.01486*, 2018.
- [123] Eneldo Loza Mencía, Gerard De Melo, and Jinseok Nam. Medical concept embeddings via labeled background corpora. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 4629–4636, 2016.
- [124] Yixin Cao, Lifu Huang, Heng Ji, Xu Chen, and Juanzi Li. Bridge text and knowledge by learning multi-prototype entity mention embedding. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1623–1633, 2017.
- [125] Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. Joint learning of the embedding of words and entities for named entity disambiguation. *arXiv preprint arXiv:1601.01343*, 2016.
- [126] Saadullah Amin, Günter Neumann, Katherine Dunfield, Anna Vechkaeva, Kathryn Annette Chapman, and Morgan Kelly Wixted. MLT-DFKI at CLEF eHealth 2019: Multi-label Classification of ICD-10 Codes with BERT. In *CLEF (Working Notes)*, 2019.
- [127] Xi Yang, Jiang Bian, William R Hogan, and Yonghui Wu. Clinical concept extraction using transformers. *Journal of the American Medical Informatics Association*, 27(12):1935–1942, 2020.
- [128] Shang Gao, Mohammed Alawad, Michael Todd Young, John Gounley, Noah Schaefferkoetter, Hong-Jun Yoon, Xiao-Cheng Wu, Eric B. Durbin, Jennifer Doherty, Antoinette Stroup, Linda Coyle, and Georgia D Tourassi. Limitations of transformers on clinical text classification. *IEEE Journal of Biomedical and Health Informatics*, pages 1–12, 2021.
- [129] Yuqi Si and Kirk Roberts. Hierarchical transformer networks for longitudinal clinical document classification. *arXiv preprint arXiv:2104.08444*, 2021.
- [130] Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics.

- [131] Michael L Gundersen, Peter J Haug, T Allan Pryor, Rudy van Bree, Spence Koehler, Kay Bauer, and Brenda Clemons. Development and evaluation of a computerized admission diagnoses encoding system. *Computers and Biomedical Research*, 29(5):351–372, 1996.
- [132] Min Li, Zhihui Fei, Min Zeng, Fangxiang Wu, Yaohang Li, Yi Pan, and Jianxin Wang. Automated ICD-9 coding via a deep learning approach. *IEEE/ACM transactions on computational biology and bioinformatics*, 2018.
- [133] Adler Perotte, Rimma Pivovarov, Karthik Natarajan, Nicole Weiskopf, Frank Wood, and Noemie Elhadad. Diagnosis code assignment: models and evaluation metrics. *Journal of the American Medical Informatics Association*, 21:231–237, 2013.
- [134] Min Zeng, Min Li, Zhihui Fei, Ying Yu, Yi Pan, and Jianxin Wang. Automatic ICD-9 coding via deep transfer learning. *Neurocomputing*, 324:43–50, 2019.
- [135] Jingcheng Du, Qingyu Chen, Yifan Peng, Yang Xiang, Cui Tao, and Zhiyong Lu. Ml-net: multi-label classification of biomedical texts with deep neural networks. *Journal of the American Medical Informatics Association*, 26(11):1279–1285, 2019.
- [136] Tal Baumel, Jumana Nassour-Kassis, Raphael Cohen, Michael Elhadad, and Noemie Elhadad. Multi-label classification of patient notes: case study on ICD code assignment. In *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [137] Anthony Rios and Ramakanth Kavuluru. EMR Coding with Semi-Parametric Multi-Head Matching Networks. In *Proceedings of the conference. Association for Computational Linguistics. North American Chapter. Meeting*, volume 2018, page 2081. NIH Public Access, 2018.
- [138] Hamish Huggard, Aaron Zhang, Edmond Zhang, and Yun Sing Koh. Feature importance for biomedical named entity recognition. In *Australasian Joint Conference on Artificial Intelligence*, pages 406–417. Springer, 2019.
- [139] Mario Sanger, Leon Weber, Madeleine Kittner, and Ulf Leser. Classifying german animal experiment summaries with multi-lingual bert at clef ehealth 2019 task 1. In *CLEF (Working Notes)*, 2019.

- [140] Serguei VS Pakhomov, Greg Finley, Reed McEwan, Yan Wang, and Genevieve B Melton. Corpus domain effects on distributional semantic modeling of medical terms. *Bioinformatics*, 32(23):3635–3644, 2016.
- [141] Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.
- [142] Ken Flegel. What we need to learn about multimorbidity. *CMAJ*, 190(34), 2018.
- [143] Bridget L Ryan, Krista Bray Jenkyn, Salimah Z Shariff, Britney Allen, Richard H Glazier, Merrick Zwarenstein, Martin Fortin, and Moira Stewart. Beyond the grey tsunami: a cross-sectional population-based study of multimorbidity in ontario. *Canadian Journal of Public Health*, 109(5-6):845–854, 2018.
- [144] Daniel Hausmann, Vera Kiesel, Lukas Zimmerli, Narcisa Schlatter, Amandine von Gunten, Nadine Wattinger, and Thomas Rosemann. Sensitivity for multimorbidity: The role of diagnostic uncertainty of physicians when evaluating multimorbid video case-based vignettes. *PloS one*, 14(4), 2019.
- [145] Carole E Aubert, Jeffrey L Schnipper, Niklaus Fankhauser, Pedro Marques-Vidal, Jérôme Stirnemann, Andrew D Auerbach, Eyal Zimlichman, Sunil Kripalani, Eduard E Vasilevskis, Edmondo Robinson, et al. Patterns of multimorbidity associated with 30-day readmission: a multi-national study. *BMC public health*, 19(1):738, 2019.
- [146] Takahiro Mori, Shota Hamada, Satoru Yoshie, Boyoung Jeon, Xueying Jin, Hideto Takahashi, Katsuya Iijima, Tatsuro Ishizaki, and Nanako Tamiya. The associations of multimorbidity with the sum of annual medical and long-term care expenditures in japan. *BMC geriatrics*, 19(1):69, 2019.
- [147] Wei-Cheng Chang, Hsiang-Fu Yu, Kai Zhong, Yiming Yang, and Inderjit Dhillon. Taming pretrained transformers for extreme multi-label text classification. *arXiv preprint arXiv:1905.02331*, 2019.
- [148] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. O’Reilly Media, 2009.
- [149] Ilias Chalkidis, Manos Fergadiotis, Sotiris Kotitsas, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. An empirical study

on large-scale multi-label text classification including few and zero-shot labels. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7503–7515, 2020.

- [150] Xin Liang, Dawei Cheng, Fangzhou Yang, Yifeng Luo, Weining Qian, and Aoying Zhou. F-HMTC: Detecting Financial Events for Investment Decisions Based on Neural Hierarchical Multi-Label Text Classification. In *IJCAI*, pages 4490–4496, 2020.