UNIVERSITÄT KOBLENZ · LANDAU
Faculty 4: Computer Science

WeST People and Knowledge Networks
Institute for Web Science and Technologies

Deep Learning for Differential Diagnosis and Prediction in EHR Data

Master's Thesis

in partial fulfillment of the requirements for
the degree of Master of Science (M.Sc.)
in M.Sc. Web Science

submitted by
Prantik Goswami

First supervisor:	PD Dr. Matthias Thimm Institute for Web Science and Technologies
Second supervisor:	Dr. Zeyd Boukhers Institute for Web Science and Technologies

Koblenz, December 2021

Statement

I hereby certify that this thesis has been composed by me and is based on my own work, that I did not use any further resources than specified – in particular no references unmentioned in the reference section – and that I did not submit this thesis to another examination before. The paper submission is identical to the submitted electronic version.

	Yes	No
I agree to have this thesis published in the library.	☒	☐
I agree to have this thesis published on the Web.	☒	☐
The thesis text is available under a Creative Commons License (CC BY-SA 4.0).	☒	☐
The source code is available under a GNU General Public License (GPLv3).	☒	☐
The collected data is available under a Creative Commons License (CC BY-SA 4.0).	☒	☐

Koblenz, 17.12.2021

.....

(Place, Date) (Signature)

Note

- If you would like us to contact you for the graduation ceremony,
  please provide your personal E-mail address: prantik.goswami1990@gmail.com
- If you would like us to send you an invite to join the WeST Alumni
  and Members group on LinkedIn,
  please provide your LinkedIn ID : https://www.linkedin.com/in/mePrantik

Abstract

Over the last decade, the generation of massive Electronic Health Records (EHR) has allowed researchers to explore the secondary use of these data in the field of biomedical informatics researches. Recent researches showed that Deep Learning (DL) models are efficient in collecting important features from textual EHR data and predicting a disease diagnosis. Many DL research experimenters have approached automatic prediction of the International Statistical Classification of Diseases and Related Health Problems (ICD) codes from clinical notes as a disease diagnosis task. However, these approaches tend to predict wrongly because of the lack of knowledge behind the medical entities present in the clinical notes. This thesis introduces *KG-MultiResCNN - **K**nowledge **G**uided **Multi**-filter **Res**idual **C**onvolutional **N**eural **N**etwork* , a Deep Neural Network (DNN) model that utilizes external knowledge of the medical entities in clinical text for better prediction. The model uses the Wikidata Knowledge Graph (KG) to extract embeddings of the medical entities. The KG embeddings and the word embeddings combined pass through multiple convolution filters and residual blocks to finally predict the ICD codes. As a differential diagnosis approach, the model predicts the relevant ICD codes while rejecting the non-related codes. Extensive experiments with MIMIC-III data showed that KG-MultiResCNN significantly outperformed the current state-of-the-art model and other baseline approaches.

Contents

1 Introduction

Over the past decade, the ease of curating, maintaining, and sharing electronic data has motivated healthcare providers and hospitals to slowly adapt from a paper-based approach to Electronic Health Records (EHR). EHR data from millions of patients are now routinely collected across diverse healthcare institutions [14]. According to the Office of the National Coordinator for Health Information Technology (ONC [1]) report in the USA, nearly 84% of healthcare providers globally have already adopted digital administration, which made a rapid growth of EHR generation between the year 2008 to 2016 itself[3]. The data generation has skyrocketed with about 50 Petabytes of data being generated worldwide every year [135]. These EHRs contain large amounts of longitudinal patient data generated as a byproduct of daily clinical activities. From all the current generated EHR data, a wide variety of data types can be identified. The types include structured data such as - basic demographics, drug details, diagnoses results, and laboratory tests, medical images, and unstructured free-text clinical notes. Based on the different literature [106] [53] [35] available, we can categorize the EHR data into mainly 3 types -

- Descriptive: Descriptive data contains a patient's health-related details like age, gender, blood pressure, and other symptomatic data.
- Diagnostic: Diagnostic data contains the clinical diagnosis results in text and images format for a patient.
- Decision: Decision data contains the detailed decision made by physicians or doctors based on descriptive and diagnostic data of a patient. While descriptive and diagnostic data are mostly structured, decision data are often free text types, making it unstructured.

Even though a massive amount of EHR data are being generated and stored every day, at least 97% of these digitally archived data remain underutilized [1]. While primarily designed for improving healthcare efficiency from an operational standpoint, many studies [17] [55] have found a secondary use of this vast trove of underutilized EHR data. In the field of biomedical informatics applications EHR data provides an excellent opportunity for improving data-driven decision making in hospitals and the health care sectors[1]. However, analyzing and acquiring valuable information from these vast volumes of EHR data is not only expensive for human computation but also time-consuming and often beyond human ability [63]. This manual data analysis process is often considered "black art," requiring creativity, trial-and-error, and sometimes luck[33].
Artificial intelligence (AI) and Machine Learning (ML) address this issue of manual processing by facilitating both speed and accuracy in meaningful information extraction. In particular, various data mining and ML research works like medical concept extraction [81][56], patient trajectory modeling [34], disease inference

[1] https://www.healthit.gov/

[139][9], and clinical decision support systems are benefited by the use of patient data contained in EHR systems. The meaningful use of these EHR data also facilitates opportunities to grow from individual-level to population-level research in biomedical informatics [52].
Based on various types of EHR data, different approaches [74] [71] [98] [104] of ML techniques have been explored. For example, for image-related EHR data, more advanced Deep Neural Networks (DNN) have shown promising results in object detection and segmentation [101]. Whereas for textual EHR data, Natural Language Processing (NLP) has been used, and it focuses on analyzing text to infer meaning from words. In recent years, Deep Learning (DL) [44] has seen exponential growth, largely based on the new technology-driven computational power and the availability of massive digital datasets. To this end, DL methods have observed remarkable improvements in the ability of machines to understand and manipulate various types of data such as images and language. The advancement of DL has allowed NLP approaches to adapt deep learning techniques (called DL-NLP) and has proven useful with increased accuracy and efficiency [65] [40] [46] [127] [28]. Two of the most frequently used DL methods are Recurrent Neural Networks (RNNs) [82], and Convolutional Neural Networks (CNNs) [7]. RNNs and CNNs have been successfully applied in different domains for various free-text analysis tasks such as text classification [27], sentiment analysis [137], summarization [134], and machine translation [48].

1.1 Problem Statement and Thesis Motivation

The rise of DL-NLP has motivated ML researchers to tackle some crucial challenges in diagnostic decision making, such as temporality in working with EHR data. EHR data is often recorded in a timely event fashion, which means a patient's visit can be dependent on the history of illness. Temporality is particularly important for chronic diseases where symptoms start appearing gradually and rapidly[75]. In EHR data, all these sequential temporal events are captured promptly. Free-text clinical notes such as discharge summary, physician notes, and hospital nursing observation notes contain all the temporal incidence of a patient. This sequential temporal pattern of EHR data makes it harder for traditional machine learning algorithms to understand and capture the temporal dependency and clinical relevance of the data.
EHR data also contains information about multiple other unobserved chronic conditions such as hypertension, diabetes, asthma, COPD, epilepsy, and osteoarthritis. In clinical terms, it is called multi-morbidity. Despite the increasing concerns about multi-morbidity, professional caregivers under-diagnosed up to 71% of multimorbid patients [49]. A possible reason could be that physicians frequently miss diagnosing diseases outside their field of specialization. Traditional models also have not shown auspicious results when it comes to predicting multiple diagnosis outcomes [23].

The time factor is another issue that troubled many patients as diagnosis of some disease takes a very long time (for some cases several years) [4]. In addition, some rare diseases are hard to determine by the physicians as the symptoms are often atypical and can point in many different directions making the exact disease diagnosis too time-consuming. Data-driven methods could help in analyzing the data and lower the disease diagnosis time to a considerable amount [38].
Recent works[23][110][19] on DL-NLP include using CNN and RNN with patients' textual EHR data with the temporal sequence of structured events to face the challenges of disease prediction, as mentioned earlier. Choi et al. [23] implemented a deep learning model called "Doctor AI" based on RNN and used temporal sequence EHR data to predict multiple disease diagnoses. "Doctor AI" achieved a score of 79.58% recall on benchmark datasets and outperformed baseline models. However, the data used in Doctor AI are from a single data source, and it is not entirely random. Therefore, the approach had to undergo considerable data processing to create structured data for their model. In real life, EHR data is generated from various sources, and most of these data are highly heterogeneous and free-text in nature. Traditionally, input features to an ML algorithm use hand-crafted raw data depending on the practitioner's credibility, expertise, and domain knowledge. However, this process often leads to losing crucial information essential for improving ML models. Therefore, for better learnability, a model should allow EHR data that can be used without much data processing. For example, as an earlier approach, different research works[126][88][87][124][39][78][76] attempted to predict disease from free-text clinical notes. Derived by the massive computational power and faster GPUs [20], DL approaches can process the textual EHR data easily and quickly. However, DL models can not directly process textual data. As an additional step, DL methods on textual data often come with an extra embedding layer that provides a high dimensional embedding vector for a word token. DL-NLP models with this extra embedding layer use the embed vectors of the text to predict the results. Based on the different embedding options such as Word2Vec[83], Glove[93], ELMO[95], and Bert[120], different DL-NLP models for disease diagnosis applications have been explored. From the different available applications, "Patient Phenotyping"[32] [26][131][132] and "Risk Prediction" [24][64][21][136] are mainly explored as a singular disease prediction or a singular outcome prediction. As a more complex and multiple disease prediction task the application of "Automatic ICD Coding" [84][11][69] has been researched. In the ICD code prediction task, DL approaches try to predict the ICD code most suited for a clinical text. International Statistical Classification of Diseases and Related Health Problems (ICD) is the global representation of a disease or a clinical procedure. Hospitals and healthcare providers generally assign the ICD codes to a patient's admission for better understandability of the actual disease, easier maintenance of diagnostic information, and billing [85][18][10]. DL approaches like CAML[84] attempted the ICD code prediction task as a multi-label prediction task. They used a simple CNN network with an additional label attention mechanism to predict ICD codes from patients'

free-text discharge notes. Even though the model performed fairly, it failed to predict ICD codes from a varied set of discharge summary notes. As an alternative, Fei Li and Hong Yu [69] argued that a single and fixed-length CNN layer might not be sufficient to capture the detailed features of a clinical text. As a suggestive and current state-of-the-art approach, they used a multi-filter residual CNN to capture the detailed feature representation of a clinical document. The result showed a significant improvement over the CAML model. However, the model failed to capture the similarities between varied medical terms present in the clinical document. Clinical documents are generally filled with many clinically important words. The meaning of those words is hard to capture from just a clinical text unless some external knowledge is provided. For example, to identify the similarity between the two sentences, "The patient showed signs of high fever" and "current symptoms indicate acute febrile response," the model must need external knowledge that "fever" and "febrile response" are similar terms. Moreover, the model failed to identify the important rare terms present in the text, thus making faulty predictions. These existing problems motivate this thesis to create a model that uses clinical text data guided by external medical knowledge. And then use the data to train a DL-NLP model that can predict multiple ICD codes associated with the text and thus predict the disease outcomes from textual EHR data.
I will sequentially discuss the research questions and then the proposed method to solve the problems in the below sections.

1.2 Research Questions and Thesis Contribution

The existing challenges in current DL-NLP models set the aim of this thesis to investigate the following research questions -

- **RQ1:** How can we use external medical knowledge with raw heterogeneous textual EHR data for better disease prediction?
- **RQ2:** How can we use DL to predict multiple diseases outcomes while removing non-related diseases using free-text EHR data?

Thesis Contribution:

This thesis addresses the problems as mentioned earlier in existing models and the research questions in the following way -

- As an enhancement of the model MultiResCNN[69] by Fei Li and Hong Yu, this thesis implements an extra embedding layer along with the word embedding layer to the model. The additional embedding layer uses the important medically significant tokens/entities of types such as treatment, test, and problem; then, it provides a knowledge graph embedding vector for those tokens. The model is then trained with the concatenated embedding vectors of the text words and the medical entities present in the text.

- As a novel approach, this thesis is the first that uses two embedding layers for a model in the domain of biomedical research.
- Along with the knowledge graph embedding, this thesis calculates the Term Frequency-Inverse Document Frequency (Tf-idf) value of each word present in the text and then uses that as a weighting factor to the word embedding vector of the words.
- To deal with the massive size of embedding vectors provided by the two embedding layers, this thesis uses two residual (ResNet) blocks to extract better feature representation, unlike the state-of-the-art MultiResCNN[69] that used only one residual block.

This thesis introduces the model ***K**nowledge **G**uided **Multi**-Filter **Res**idual **C**onvolutional **N**eural **N**etwork (KG-MultiResCNN)*. The model is trained on the Medical Information Mart for Intensive Care (MIMIC-III) discharge summary notes, and it predicts diagnosis ICD codes using the free-text notes. Discharge summary notes are essential for this research as they contain vital patient information such as patient demographics, medical history, family history, admission observations, and lab test results. The discharge summary notes also contain the disease diagnosis of the patient. However, the disease diagnosis mentioned in the discharge summary notes is often superficial and falls short of providing a specific diagnosis. For example, a disease diagnosis in the discharge summary note can be "Restrictive lung defect," which does not mention the actual disease. There can be multiple conditions for a "Restrictive lung defect," such as "Acute bronchospasm," "Atelectasis," or "Mediastinitis." A diagnosis ICD prediction system can identify the proper disease as each disease has its unique ICD code. The KG-MultiResCNN does the job by predicting the actual diagnosis ICD codes from discharge summary notes. The KG-MultiResCNN model is developed on top of the MultiResCNN[69] designed by Fei Li and Hong Yu. This thesis utilizes the Python code implementation from the publicly available repository of MultiResCNN[2]. As a differential diagnosis approach, KG-MultiResCNN is trained not only to learn the correct ICD codes but also the wrong ICD codes that should not be associated with the text. The model is evaluated against the baseline models of CAML[84], DR-CAML[84], and MultiResCNN[69].
The code for this thesis is publicly available on GitHub[3]. The thesis is done using the Python programming language. Specifically, this thesis used Jupyter Notebook to implement the model as a single file application. The KG-MultiResCNN model is trained on the high-end GPU of "NVIDIA TITAN V" provided by the department of Web Science and Technology[4] at the University of Koblenz-Landau.

[2]https://github.com/foxlf823/Multi-Filter-Residual-Convolutional-Neural-Network
[3]https://github.com/PrantikGoswami/KG-MultiResCNN
[4]https://west.uni-koblenz.de/

Thesis Outline:

This paper is organized as follows.

- Section 2 discusses the related works done in the field of disease diagnosis on EHR data. More specifically, it highlights the ICD code prediction task as it is more relevant for the scope of this thesis.
- Section 3 introduces the proposed KG-MultiResCNN model and describes the methodology. Then, in further subsections, the architecture of the model is described. Finally, the implementation detail of the model is explained.
- Section 4 describes a general structure of EHR data first. Then introduces the data repository used for this thesis. The section further provides the data collection method and the description of the collected data.
- Section 5 presents the experimental setup starting with the data processing step to training and finally parameter tuning of the model.
- Section 6 provides the evaluation details of the model. Further data processing and experimental setup steps are discussed to evaluate the model against existing works.
- Section 7 presents the results and findings of the thesis. It also provides a result comparison against baseline approaches.
- Section 8 and section 9 concludes the paper with a summary of the thesis, key findings, limitations, and future improvements.

2 Related Work

The use of AI and ML to learn from medical data has been researched for a long. EHR data have provided the data needs for all those research works.
Even though EHR data provides rich digital data [6], a recent study of the medical literature found that predictive models built with EHR data use a median of only 27 variables. Furthermore, it relies on traditional data generalization methods and is built using data from a single-center [43]. This section provides an overview of different works with EHR data and their data collection methods.
In different ML fields, research works are done with different types of EHR data. In the field of Computer Vision (CV), imaged-related EHR data have been experimented with extensively because CNNs have achieved human-level performance in object classification tasks [101]. Other works involved Reinforcement Learning (RL) [111] [51]. A pre-learned supervised learning model keeps learning and correcting from expert demonstration and is accomplished either by learning to predict the expert's actions directly via supervised learning or by inferring the expert's objective [5] [100].
For the scope of our research, this thesis mainly focuses on supervised learning with textual EHR data. So, this section highlights some well-known works in the field of NLP and DL-NLP.

2.1 NLP

NLP focuses on analyzing textual data. According to Sheikhalishahi et al. [104], NLP models have benefitted hugely and have seen tremendous growth as more textual EHR data generates rapidly. Classical NLP depends on various manually defined rules (e.g., regular expression patterns, terminology lookup, dictionary) for extracting specific information from free-text data. Defining a uniform set of rules can be challenging, as one set of rules that applies to one particular database might not be used to another. Liang et al. [71] suggested a basic information extraction model that extracted the key concepts and associated categories in EHR raw data and transformed them into reformatted clinical data in query-answer pairs. This NLP approach involved lexicon creation where a lexicon was generated by manually reading sentences in the training data and selecting clinically relevant words for query-answer model construction [71] [36] [80]. As an improvement to the system, the method emphasizes schema design consisting of a list of physician curated question-answer pairs that the physician would use to extract symptom information towards the diagnosis. Tokenization and word embedding also helped in embedding tokens with features to represent the semantics and similarities of any query word in the higher dimensional space [71]. The research collected a total, 101.6 million data points from 1,362,559 pediatric patient visits. Furthermore, the data were analyzed to train and validate the framework. Even though keyword search and word tokenization received a good amount of success, the unstructured, noisy nature of the narrative text (e.g., grammatical ambiguity, synonyms, misspelling,

or negation of concepts) is still a bottleneck for this process. Additional rules or other more complex criteria have been added to the keyword search to improve the performance. In a typical rule-based system, standards need to be pre-defined by domain experts. For example, Wiley et al. [125] incurred a rule-based system for statin-induced myotoxicity detection. First, they developed a set of keywords for their work by manually annotating about 300 patients' allergy listings. Then they developed a set of rules on top of the keywords to detect contextual mentions around the keywords. This study achieved a positive predictive value (PPV) score of 86 percent and a negative (NPV) score of 91 percent. Some of the other important literature [86][129][70][47] applied rule-based NLP tools and achieved modest success.
Even though these research works got some attention, much manual work was needed to prepare the data. This problem brings to the need for a deep learning approach on NLP.

2.2 DL-NLP

Over the last decade, extensive research works [65] [40] [46] [127] [28] have shown the potential of deep learning approaches on NLP tasks, including text classification, language translation, POS tagging, entity recognition, sentiment analysis, and paraphrase detection. Different domains[133] such as finance, automobile, e-Commerce have already benefited hugely from the application of DL-NLP. Researchers[126] [88] [87] [124] [39] [78] [76] have also exploited the possibility of using DL-NLP approaches in the healthcare domain because the healthcare domain produces a massive amount of free-text electronic data. One of the main applications in the healthcare domain is medical diagnostic decision-making[66][112], which has been explored for decades. Motivated by the fact that medical diagnosis using DL approaches has already reached human-level accuracy on image data, DL approaches on text data gained much attention in the same application.

Patient Phenotyping:

Derived by the enormous computation power, DL approaches can utilize long free text EHR data for disease detection. One particular application area for EHR where DL approaches are used is called patient phenotyping[32]. As a diagnostic approach, patient phenotyping aims to predict patients' medical condition or any risk factor based on a patient's symptoms and other medical conditions. According to Collobert et al.[26], DL approaches help to identify symptomatic details and other intrinsic structures from high dimensional textual data like EHR data. Word vector [97][13] representation from unstructured text has provided a foundation to the DL approaches in clinical phenotyping and disease diagnosis. One of the initial DL approaches with Deep Neural Network (DNN) on clinical phenotyping was done by Beaulieu-Jones et al.[13]. Their research used a neural network structure to learn the

free text structure of a patient's discharge report. From the data, they constructed different phenotypes to identify a patient's disease. Their approach outperformed traditional ML models such as SVM, Decision Tree, and Random Forest. In more advanced research, Wu et al.[126] used CNN with word vectors from pre-trained embedding models to recognize named entities from free-text data. The extracted entities are then classified for different phenotypes. Their model outperformed the conditional random field (CRF) baseline, model. In most recent works, Gehrmann et al.[41] used CNN to do clinical phenotyping by detecting medical phrases from the text. They used MIMIC-III discharge summary notes for their model evaluation, and it turned out that their model could predict some difficult phases that core domain experts can only identify. As an extension of this work, Yang et. al. [131] utilized a word and sentence level CNN architecture to do clinical phenotyping in determining ten medical disorders. They proposed that other than using just word level embedding, addition of sentence level embedding gives more contextual meaning to the embedding and CNN network performs better in this setting. In their proposed model "ws-CNN" they first created a sequence of word embedding vector for a sentence and then used the summing and average pooling mechanism on the word vectors sequence to create the corresponding sentence vector. Then they concatenated the word embedding and their corresponding sentence embedding together to form the final feature embedding. Their result showed that the model achieved the largest performance gain in classifying medical conditions. They also concluded that large number of samples and good quality of data per label is the most important criteria for a better performing model. To include additional information for the betterment of quality and quantity of data, Liang et. al.[132] utilized a knowledge guided CNN model in combination with rule-based features from clinical text. In their study they first utilized a rule based system to identify medical trigger terms in a text. For each trigger terms they created the embedding vectors by using a pre-trained (trained on MIMIC-III data) word2vec model. For the additional knowledge they used medical knowledge base. They used a process called MetaMap[8] that can help in linking clinical text to create Concept Unique Identifiers (CUIs) of Unified Medical Language System (UMLS)[15]. These CUIs can provide entity embeddings as additional knowledge to the model. Their CNN model combined with trigger term embedding and knowledge entity embedding is used on the i2b2 obesity challenge dataset[117]. The results showed that their model achieved an overall f1-score of 67% in finding common medical conditions for obesity. Other than medical phenotyping, DL approaches are used hugely in risk and mortality prediction tasks.

Risk Prediction:

Guided by the fact that DL approaches can identify patterns in a text for a particular outcome disease, many researchers[24][116][72][21][124][88] have explored DL techniques in predicting future clinical outcomes such as mortality, hospital readmission, and any other medical risks or diseases. Based on the outcome prediction of these approaches, the research works can be categorized into mainly two parts- 1) One-time outcome (e.g., Heart failure, Hypertension, Suicide risk, Mortality) 2) Temporal outcome (e.g., Hospital readmission, Heart failure within six months, future disease prediction from historical medical data)
The most straightforward approach between this two is one-time or static outcome prediction, as it does not consider the temporal dependency of the data. As an initial work, Liang et al.[72] used a Deep Belief Network (DBN) to create a patient vector representation by training each layer of the DBN separately. Then they used the patient vectors to support vector machines (SVN) to predict the disease finally. They used the model on the data of patients with Hypertension. Their model showed promising results in predicting Hypertension from free-text clinical data. In a similar approach, Choi et al.[24] used a linear model of several ANNs to predict the outcome of heart failure. Their method used a new way to represent the free-text clinical document as a combination of critical medical concepts co-occurring in the text. So the model learns the distribution of the co-occurring words to predict disease. Their result showed that other than using simple word representation of the documents, the medical concept representation helps predict better outcomes. In another disease detection approach, Lauritsen et al. [64] have presented a scalable deep learning method to detect sepsis early with heterogeneous data that includes hospitalizations within and outside of the ICUs from multiple health centers. They used a CNN-LSTM based model approach to detect feature and temporal patterns present in the data. The data included health data on all citizens 18 years or older with residency in four Danish municipalities (Odder, Hedensted, Skanderborg, and Horsens). The result showed that the model achieved an AUROC of 0.856 and an mAP of 0.79 when evaluated 3h before sepsis on the vital sign test data. Cheng et al. [21] experimented with capturing temporal features from sparse EHR data by using CNN as a chronic disease detection approach. In their research, they represent the EHR data of each patient in a temporal matrix fashion with time as one dimension and events as the other. Then the data is passed through a series of convolutional layers to extract the most significant features. Different fusion mechanism such as "early fusion," "late fusion," and "temporal fusion" was applied to the EHR data. The fusion mechanism helped to incorporate the temporal smoothness into the data. The model showed promising results in the early prediction of onset risk when evaluated on a real-world EHR data warehouse with an EHR record of 319,650 patients over four years.
The other approach on DL involves temporal outcome prediction. The primary purpose of these approaches is to predict an outcome of a disease within a specific time

interval by analyzing time series EHR data. Lipton et al.,[73] in their work, used an Long Short-Term Memory (LSTM) network (a variant of RNN) to predict disease outcomes based on temporal data. Their work used a target replication strategy to predict disease from a list of 128 diseases in each time step. To reduce the overfitting of their model, they used additional information from the patients' data as a technique called auxiliary outputs. They achieved the best performance after assembling the LSTM model with a standard MLP of 3 layers. Doctor AI[23] is another successful approach with Recurrent Neural Network (RNN) based deep learning model developed to predict future disease diagnosis and medical prescriptions. Because of the RNN architecture, Doctor AI can assess the entire medical history as time sequence data to extract essential features over a while. Doctor AI fed its neural network with data from 700,000 EHRs and randomly combined them to make and test new variables for disease risk. As an extension of the DoctorAI project, Choi et al.[25] used a GRU network for predicting heart disease during several time prediction windows. They used a time sequence clinical event vector representation from the patients' historical EHR data. They achieved better performance over their previous baseline approach. In another time series prediction approach, Nguyen et al. described a deep learning model named Deepr[87]. The model is a multi-layered architecture based on Convolutional Neural Networks (CNNs). The network learns how to extract features from medical records to predict the risk for the patients. The medical records are collected as a sequence of visits, and for each visit, a subset of coded diagnosis, lab test, and text data are combined. The model showed promising results in predicting unplanned readmission within six months. Following similar work, Pham et al. created a model called DeepCare[96]. They argued that LSTM networks better capture the temporal irregularities present in a patient's EHR data. For the model, they used two embedding vectors created via the skip-gram embedding approach. Utilizing the current clinical concepts in the data, one diagnosis code embedding vector and another intervention code embedding vector were created. The result showed that their model performed well for predicting readmission for both diabetic and mental health patients. In a different approach to predicting mortality in ICU, Kim et al. developed a CNN-based model called PROMPT[58]. They used the model to predict the mortality of critically ill children who are admitted to the ICU. They created two groups of data for vital signs development in the last 24 hours window from the data. The first group is the positive instance where they extracted data between 6 to 60 hours before the patient's death. The second group is the negative instance where the patient survived during the stay in the ICU. The CNN model learns the vital signs representations from the data and predicts a binary outcome. For mortality prediction within 6 hours before death, PROMPT achieved ad AUROC of 96%. They showed that their approach outperformed the LSTM based approaches. In a recent practice, Zhang et. al.[136] adopted a DNN model to predict three risk prediction tasks - in-hospital mortality, hospital readmission, and extended stay prediction. Their work argued that combining the structured and unstructured EHR data can provide a better present represen-

tation. They used two fusion models, namely "Fusion-CNN" and "Fusion-LSTM," to create a document vector representation of the free-text clinical notes. Whereas, for the static patient demographic details and other admission-related details, they used one-hot encoding. The document vector and the one-hot encoded vector together serve as the patient representation vector. The patient representation vector is finally classified using a binary classifier to predict the result on the data. They applied the model on MIMIC-III data and showed that the approach performed better and produced more accurate predictions for the three risk prediction tasks. Even though these works showed promising results, a general multi-label prediction task seemed more important when the symptoms were unknown and hard to predict. As a solution, the disease ICD code prediction task became a topic in biomedical research.

Automatic ICD Coding:

The International Classification of Diseases (ICD) is a global term representing a disease diagnosis or procedure performed on a patient. The World Health Organization (WHO) maintains and provides this list of pre-defined ICD codes. Most physicians and healthcare providers have already adopted the ICD code for different reasons such as - better usability and maintainability, reimbursement, accessible storage, and retrieval of diagnostic and procedural information[18][85]. All the clinical EHR documents are linked with the corresponding ICD codes for every patient's hospital admission as part of hospital services. However, assigning an ICD code to a free-text EHR document is not a simple task as it is laborious, expensive, error-prone, and requires a good amount of domain (health care) knowledge[91]. As a solution, the research on automatic ICD coding from free-text clinical notes has been ongoing for more than two decades[62][30]. The old methods of automatic ICD coding were mainly dependent on handcrafted approaches[103]. With the rise of better technology and more data processing power, different research scopes started opening. The early works on ICD code prediction tasks were based on traditional supervised ML approaches. Perotte et al.[94] used Support Vector Machine (SVM) to classify "flat" and "hierarchical" ICD codes. Koopman et al.[61] used SVM to classify hierarchical ICD codes related to cancer from free-text death certificates in a similar job. In DL approaches in the last five years, there has been much improvement in ICD coding. To start with, Shi et al.[105] used character-level LSTM to learn the similarities between the discharge summary notes and the description of ICD codes. In a different approach, Prakash et al.[99] created a neural memory network model called "C-MemNNs" that learns representations from free-text data and predicts top-50 and top-100 codes. In their approach, they used external knowledge from Wikipedia to enhance the model performance. In the Recurrent Neural Network (RNN) approach, Vani et al.[119] created a Grounded Recurrent Neural Network (GRU) that utilizes label-specific dimensions for the hidden units to predict specific diseases. Baumel et al.[12] used Hierarchical Attention-bidirectional Gated Recurrent Unit

(HA-GRU) to assign multiple ICD codes to patients' discharge summary notes. As an embedding approach, Wang et al.[122] argued that projecting word and label vectors in the same embedding vector space yields better results. For their approach, they proposed a mixed embedding model. The model calculates the cosine similarity between the word embedding vector and the label vector to predict the labels.
The following list contains the most recent works in automatic ICD prediction in clinical disease diagnosis as a baseline reference to this thesis.

- As one of the recent essential and works, Xu et al.[130] approached the ICD prediction task as a combination of 3 model outputs. Their work implemented an ensemble-based approach to deal with unstructured, semi-structured, and tabular data. They included DL methods to classify uninstructed and semi-structured clinical notes, and for the tabular data, they utilized a decision tree to classify the data. For the unstructured texts, they used a CNN-based model called "Text-TF-IDF-CNN." The model used word2vec[83] word embedding for each word in the text. The word embedding is filtered through multi-filter convolution layers and then collected together using max pooling. Finally, the max pooled features from the convolution layers and the Tf-IDF features of the whole text are concatenated to form the final feature vector. The final feature vector is then passed to a fully connected layer for the classification. For the semi-structured text data, they used character-level CNN and bidirectional LSTM for a task called "Diagnosis-based Ranking (DR)." In this particular task, they created a low-dimensional diagnosis vector from the text and mapped it into the same vector space of the ICD code description vector. According to their approach, the model minimizes the distance between the two vectors to create a similarity ranking. For the tabular data, they created a binary feature vector for each type of data. Furthermore, using the data to a decision tree for a binary multi-class classifier. They used MIMIC-III clinical notes and other tabular data such as lab events, prescriptions, microbiology events, and chart events for evaluation. In their approach, the MIMIC-III ICD codes were used to predict the top 32 clinically significant ICD codes. The result concluded that adding multiple modalities of data yields better results.

- As a label weight attention approach, Mullenbach et al.[84] argued that DL approaches on ICD coding tasks perform better with an attention mechanism. Their approach proposed a single-filter CNN network model. Additionally, they proposed label attention following the convolution operation. They called their model Convolutional Attention for Multi-Label classification (CAML). As a solution to the multi-label classification problem, their model utilized a pre-trained word vector for each word in the discharge summary note to predict multiple ICD codes finally. For their work, they used MIMIC-III and MIMIC-II discharge summary notes to predict the complete list of ICD codes present in the MIMIC data. They also used their model to predict the top 50 frequently occurring ICD codes as an additional evaluation step. The results

showed that their approach significantly outperformed all the past methods. They finally concluded that adding the label attention layer helped the model find the most significant text features, thus improving the performance.

- As an extension of CAML[84], Tian Bai, and Slobodan Vucetic[11] created a knowledge source integration system on top of the baseline model. In their work, they proposed that added external knowledge improves the model performance significantly. Their Knowledge Source Integration (KSI) framework uses the superficial knowledge from Wikipedia to provide extra weight on the input text for the prediction of a particular ICD code. Motivated by the fact that the DL models were not very good in predicting rare diseases, they showed that adding KSI with CAML helped predict better results along with rare diseases. To evaluate their model, they used the MIMIC-III dataset, and for each ICD code in the MIMIC-III dataset, they extracted the disease document from Wikipedia.

- As the most recent state-of-the-art model, Fei Li, and Hong Yu[69] proposed another improvement over CAML[84]. They approached the ICD code prediction problem as a multi-label prediction task. They used free text discharge summary notes to predict multiple possible ICD codes related to that discharge summary text. For the multi-label prediction, they used a one-hot encoded label vector. The label vector has the dimension same as the total number of ICD codes. For example, if the task predicts ICD codes from 100 ICD codes, the label vector is 100 dimensional. The ICD codes that are true for the discharge note are set to 1, and the other codes are set to 0. For the modeling, they argued that the multiple filter CNN network performs much better than using a single filter CNN network. Along with the CNN filters, they used one residual network[50] following each of the CNN filters. They called their model Multi-Filter Residual Convolutional Neural Network (MultiResCNN). The input layer used the word2vec[83] model that transformed each word in the discharge summary notes into a word vector. The word vectors then passed through the CNN filters and residual block to create a feature distribution of the input text. Following the work of CAML[84], they employed a label attention mechanism for better prediction accuracy. Their approach created a weight matrix for the label distribution and multiplied it with the text feature distribution. Intuitively this process learns the important word features that are important for a particular label and applies extra weight to those features. The weight attention features are finally classified using a fully connected layer. They experimented with their approach for predicting ICD codes on the MIMIC-III discharge summary notes dataset. For evaluation of their model, they used MIMIC-Full codes and MIMIC-50 codes. The result showed that using multiple filters CNN network, the prediction accuracy of the model increases significantly. Furthermore, their result showed that adding a layer of a residual network increases the performance even more.

3 Approach

3.1 Overview

This thesis introduces *KG-MultiResCNN - **K**nowledge **G**uided **Multi**-filter **Res**idual **C**onvolutional **N**eural **N**etwork* , a model that can predict disease ICD codes given an unstructured clinical text. The model aims to predict disease diagnosis and thereby help doctors and medical staff in diagnostic decision making. It does so by extracting important features from the clinical text to replicate human-level decision support. The intuition is that certain words/tokens in a text hold the maximum value. For example, given a clinical text *"The patient is admitted with severe chest pain."*, the model should be classified the text as *"Heart Attack"*. Internally the model should be able to learn the importance of the key terms and their context in the text to classify the text to a precise disease class. For the above example, the goal of the model is to learn the importance of the term *"chest pain"* to predict the class as *"Heart Attack (Acute myocardial infarction)" - 410.00 (ICD 9)*. The best way to achieve the goal is to provide the model with a tokenized text for each document of clinical text. For a given text of $|M|$ words, the model is passed with each word from the text, and the model learns the representation of each text to understand the context and is finally able to find a correlation between the texts and the resulting disease. As an added attention step, the model is provided with the key terms separately after they are extracted using an *Entity Extraction* method. The combined representation of the tokens and the extracted entities is then fed to the model for the training. Internally the model consists of multiple CNN layers that capture the feature representations. Finally, the model output is evaluated against the ground truth value using a loss function to adjust the model parameters properly. In the next section, the architectural details of the methodology are discussed.

3.2 KG-MultiResCNN

The proposed model for this thesis is called KG-MultiResCNN, whose architecture is shown in figure 1.
The whole architecture is divided into six parts, and each part is described below.

3.2.1 Input Word Embedding Layer

The input layer creates an embedding matrix (E) out of the sequence of the words of a text document. The word sequence is denoted as w, and w is defined as $w = \{w_1, w_2, ..., w_n\}$, n is the sequence number or the number of words present in the text. For each word, there is a pre-trained embedding vector created by training the word2vec model [83]. As a weighting mechanism, the Tf-idf score of each word is calculated. The score is then multiplied with the embedding vector of that word. If the Tf-idf score of a word becomes 0, then the pure embedding vector of that word is used. The embedding vector can be formulated as $e = u \times g$ where u is the word

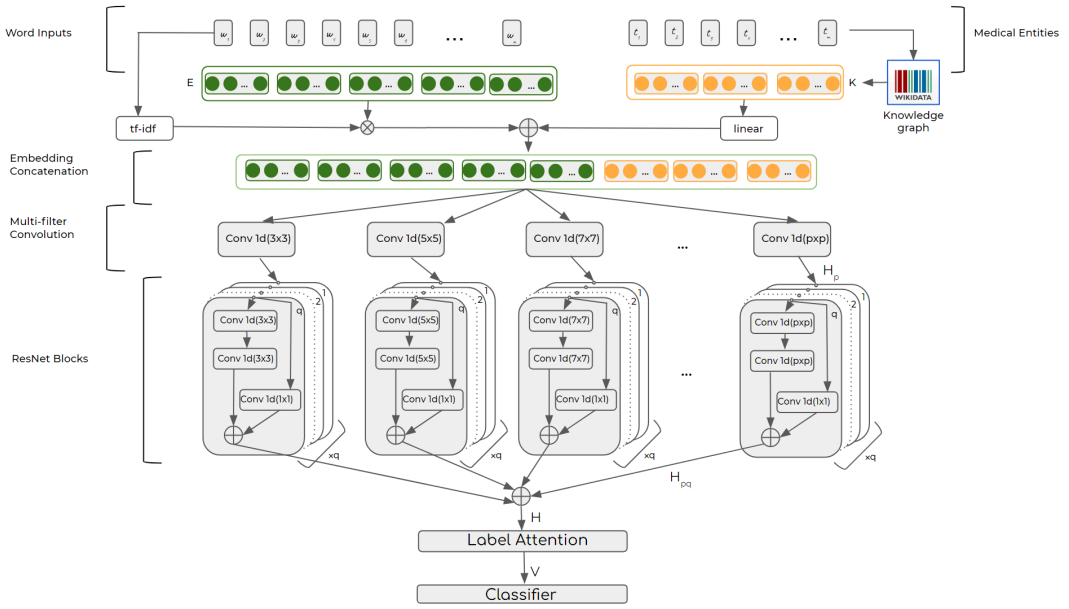

Figure 1: The general architectural diagram of "Kg-MultiResCNN" follwing the work of Li and Yu[69].

embedding and $g > 0$ is the Tf-idf score of that word. $\times$ indicates the scalar vactor multiplication. Therefore, the input embedding matrix will be $E = \{e_1, e_2, ..., e_n\}$ where e_i is the final word embedding vector of i^{th} word and $e_i \in \mathbb{R}^{d^e}$. d^e is the dimension of each word vector.

3.2.2 Input KG-Embedding Layer

The model is fed with an extra embedding input layer to strengthen the feature extraction process. From the text of word sequence, the most significant medical entities are extracted using a pre-trained Bert model[5]. The entity sequence is denoted as t, which is defined as $t = \{t_1, t_2, ..., t_m\}$, m is the number of medically significant entities extracted by the entity extraction model. The extracted clinically relevant tokens, or entities are used to query on a knowledge graph to get the knowledge graph embeddings of the tokens. These embeddings are used to create the additional KG embedding matrix K. For each entity t_j the corresponding embedding k_j of dimension d^t is extracted from the knowledge graph. Hence the knowledge graph embedding matrix becomes, $K = \{k_1, k_2, ..., k_m\} \in \mathbb{R}^{m \times d^t}$. The word embedding matrix and the KG embedding matrix jointly serve as the input layer to the model.

[5]https://huggingface.co/samrawal/bert-base-uncased_clinical-ner

3.2.3 Multi-Filter Convolution Layer

To extract the features from varying lengths of texts, I followed the work of Fei Li and Hong Yu [69] to create a multi-filter 1 dimensional Convolutional Neural Network architecture. The strategy is to pass the varied length of texts through a parallel set of CNN networks. However, the kernel size is of different lengths for each CNN filter. If there are p no of filters, then let us assume the kernel size of p^{th} filter would be k_p. And the convolution filter would be $W_p \in \mathbb{R}^{k_p \times d^e \times d^c}$ where d^e is the input dimension and d^c is the output dimension. In general, the filter/convolution operation on a vector reduces the size of the output vector. However, in this approach, I wish to keep the size of the output vector the same as the input. We can achieve that by adding the right kernel size, padding, and stride. We can use the equation provided by Pytorch[6] to calculate the right parameter numbers.

$$L_{out} = \left[\frac{L_{in} + 2 \times padding - dilation \times (kernel\_size - 1) - 1}{stride} + 1\right]$$

By setting the stride = 1, dilation = 1, kernel_size = k, and padding = $floor(\frac{k}{2})$ we can achieve our goal of same output size. With all these adjustments, the 1-Dimensional convolution operation can be formalized as :

$$\mathbb{C}_{p,j}(E) = W_p^T \otimes E^{j:j+k_p-1}$$

$$H_p = \sum_{j=1}^{n} tanh(\mathbb{C}_{p,j}(E))$$

Here, $\otimes$ represents a convolution operation and $\mathbb{C}_{p,j}$ indicates the output of p^{th} convolution where the input matrix position starts from jth row and ends at the row $j + k_p - 1$. H_n indicates the final layer output after the convolution output is passed though $tanh$ activation for total n sequence of input and then concatenated (indicated by $\sum$) together. A typical 1-D convolution architecture is shown in figure 2 where the convolution filter W_p slides through the embedding matrix E with a stride of 1.

3.2.4 Residual Convolution Layer

The output of each convolutional filter again goes through a series of convolution filters. Following the work of Fei Li and Hong Yu [69], this series of convolution filters together is called a residual block. Each residual block consists of 3 convolution layers. To formulate it mathematically, if we consider that we have p number of multi-filter convolution layer then each of these p number of convolution filter has a series of q number of residual blocks on top of it. Each of the residual blocks have

[6] https://pytorch.org/docs/stable/generated/torch.nn.Conv1d.html

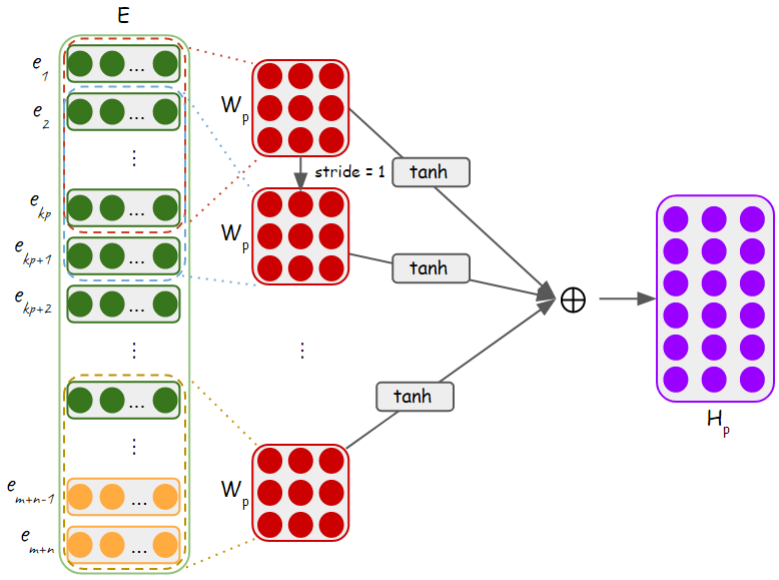

Figure 2: A general architectural diagram of 1-D convolution with stride 1.

3 convolution filters namely $r_{pq_1}, r_{pq_2}, r_{pq_3}$ and their corresponding filter weights are $W_{pq_1}, W_{pq_2}, W_{pq_3}$ where r_{pq} is the q^{th} residual block on top of p^{th} multi-filter convolution layer. The output of each convolution filter inside a residual block can be formulated as

$$\mathbb{C}_{pq_1,j}(X) = W_{pq_1}^T \otimes X^{j:j+k_{pq_1}-1}$$

$$H_{pq_1} = \sum_{j=1}^{n} tanh(\mathbb{C}_{pq_1,j}(X))$$

$$H_{pq_2} = \sum_{j=1}^{n} \mathbb{C}_{pq_2,j}(H_{pq_1})$$

$$H_{pq_3} = \sum_{j=1}^{n} \mathbb{C}_{pq_3,j}(X)$$

$$H_{pq} = tanh(H_{pq_2} + H_{pq_3})$$

Here, $+$ represents element-wise addition. H_{pq} represents the final output from the q^{th} residual block that used the initial input matrix from the output of p^{th} multi-filter convolutional block. X is the input matrix to each of the residual blocks. The first residual block receives the input as the output of the multi-filter convolution layer, and the last residual block receives the input from its previous residual block. Finally, the output of each of the final residual blocks is concatenated together to use

in the next step. The final output can be formulated as

$$H = \sum_{1}^{p} H_{pq}$$

where p is the total no of filters used in the multi-filter convolution layer. The residual architecture is shown in figure 3 where the input H_p (from the CNN channel) goes through a series of convolution filters W_{pq1}, W_{pq2} and a shortcut convolution filter W_{pq3} to finally added together to form H_{pq}.

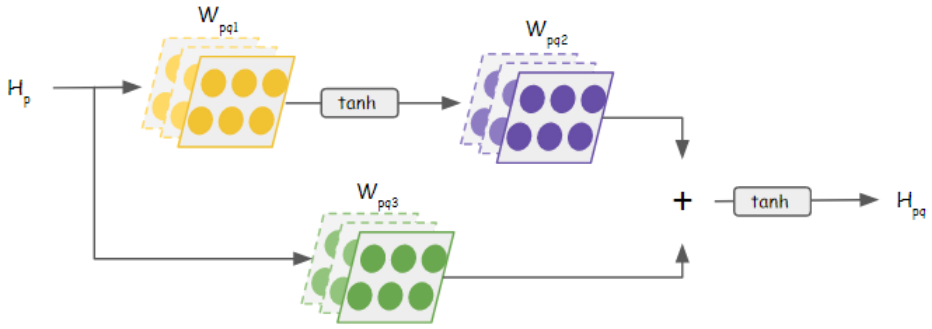

Figure 3: Architectural diagram of Residual Convolution Layer.

3.2.5 Attention Layer

The final output matrix H is typically reduced to a vector using the max-pooling operation before passing it to a classifier. However, in this model, I used an additional label attention step following the work of Fei Li and Hong Yu [69], and Mullenbach et al. [84]. The idea is that some words have higher weights for a label for multi-class classification. Therefore, the label attention can select the most relevant k-grams from the text that can benefit in predicting the correct label. Formally the procedure is to create a vector parameter U for the labels and then compute the matrix-vector product HU. Then we use a softmax layer to obtain the word distribution in the text.

$$\alpha = softmax(HU)$$

α is the attention vector. To get the final vector representation from the attention layer we again perform a matrix multiplication between the attention vector α and the input matrix H. The final output is formulated as

$$V = \alpha^T H$$

3.2.6 Output Layer

The output layer is a superficial linear layer that takes the input V from the attention layer. The linear layer does the linear transformation on the incoming data and outputs a vector of the size of the labels. The score vector of all the labels is obtained using the sum-pooling operation on the output vector. The final probability vector is calculated using sigmoid activation on the score vector for multi-class classification. The whole process is formalized below.

$$Y = VW$$

where, W is the weight matrix and the shape of W is $((p \times d^{pq}), l)$. Here, p is the total number of convolution filters used in the multi-filter convolution step, and d^{pq} is the output dimension from the residual convolution layer. l is the output dimension, the total number of labels that we are classifying. The score vector $\hat{Y}$ can be formulated as :

$$\hat{Y} = pooling(\sum_{j=1}^{l} Y_{ij})$$

and the final predicted vector is :

$$\tilde{Y} = \sigma(\hat{Y})$$

3.3 Implementation

To begin with, the implementation of the model follows the work of Fei Li and Hong Yu, [69]. For simplicity of the design and architecture, I used Jupyter Notebook [7] provided by university Jupyter Hub [8] server. The implementation is a single-page implementation yet structured like Object-Oriented Programming. In the below passage, I will describe the implementation of the model in detail. The model is implemented in such a way that it can take any text input. However, the text inputs need to go through a mandatory data processing step that I have discussed in the section 5.1. The data processing step is essential because the model can not work on raw text data. The data processing step creates a vector representation or embedding of each text input. To this end, I have used word2vec to create a 100-dimensional embedding vector for words/tokens in a text. These vector representations are the inputs to the model.

In the first step of the model, an embedding layer is created, and this embedding layer acts as a lookup table of the embedding. Figure 4 shows the general representation of the input embedding layer. To this end, the embedding layer is created with a total of 49342 unique words that are present in the training data (data splitting strategy is mentioned in section 5.1). The embedding layers receive a sequence of ids representing the words or tokens of a text. Along with the ids, the embedding

[7] https://jupyter.org/
[8] https://github.com/jupyterhub/jupyterhub

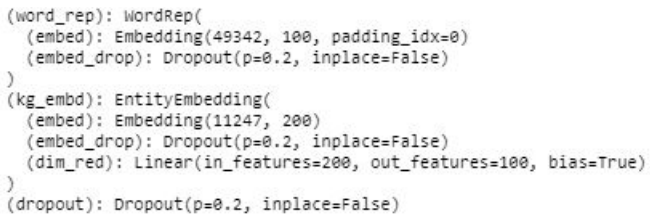

Figure 4: Word embedding and kg embedding layer structure.

layer receives the Tf-idf score of each word. This thesis utilizes "TfidfVectorizer" from "scikit-learn"[9] python package to generate the Tf-idf score of the words. Based on the word sequence ids, the embedding layer fetches the embedding of each token. Each word embedding is then multiplied with the corresponding Tf-idf score of the word that creates an input matrix. Next, as an additional step, another embedding layer is implemented that provides extra knowledge graph embedding for medically important tokens present in the text. The medical entity/token extraction and generating their knowledge graph embedding process is discussed in section 5.4. To this end, the entity extraction method found 11247 unique medical named entities from the training corpus. The KG-embedding produces a 200-dimensional embedding vector for each unique medical named entity. The KG-embedding vectors are reduced to 100-dimensional vectors using a linear filter to maintain uniformity. Embeddings from both the embedding layers are concatenated to form the final input embedding for a text. The input embedding is then passed to the multi-filter residual network. The network structure is implemented dynamically based on the chosen number of filters of CNN channels. To this end, I have chosen nine filters of sizes 3, 5, 7, 9, 13, 15, 17, 23 and 29. The numbers indicate the kernel size of each of the 1-dimensional convolution layers in the channel. As an example, Figure 5 shows the general architecture of the CNN channel with a filter size of 3. Each

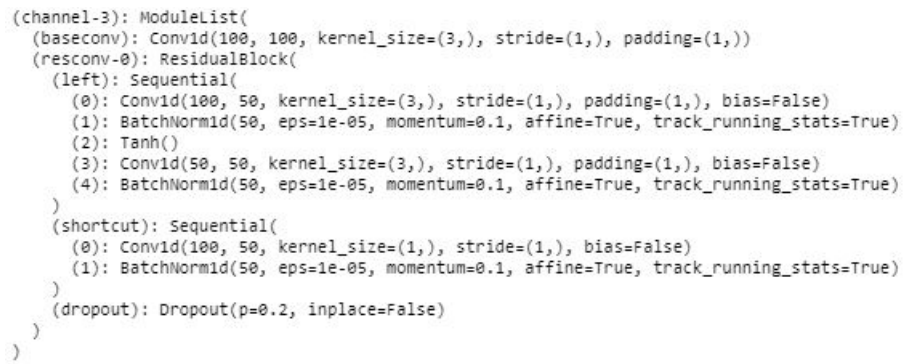

Figure 5: A Multi-filter residual convolution layer structure of filter size 3.

[9]https://scikit-learn.org/

CNN channel consists of a base convolution filter and multiple residual blocks. The example in the figure shows a convolution channel with only one residual layer. The base convolution layer has input and output channels of 100 and 100, respectively. The 100-dimensional inputs from the embedding layer pass through the base convolution layer with a kernel size of 3, stride 1, and padding 1. The output from the convolution layer produces another 100-dimensional vector that is the input to the next residual block. Each residual block consists of two convolution layers placed in series and one shortcut convolution layer. The first two sequential convolution layers take an input of 100 dimensions and reduce it to a 50-dimensional output. The two convolution layers use kernel size of 3, stride 1, and padding 1. However, the in-channel and out-channel sizes of the two convolution layers are (100, 50) and (50, 50), respectively. The shortcut convolution layer receives the same 100-dimensional input from the base convolutional layer. It converts it into a 50-dimensional output using a kernel size of 1 rather than 3, stride 1, and without any padding. The output of the series of convolution layers and the shortcut convolution layer are concatenated together to generate the final output vector of size 50 for each of the multi-filter residual convolution networks. To this end, the nine convolution channels output is concatenated together to create a final vector of size $50 \times 9 = 450$. The 450-dimensional output from the multi-filter convolution layer then goes to the output layer for the final classification. Figure 6 depicts the architecture of the output layer. The output layer starts with a label attention layer. As a starting point in the

```
  (output_layer): OutputLayer(
    (U): Linear(in_features=450, out_features=6918, bias=True)
    (final): Linear(in_features=450, out_features=6918, bias=True)
    (loss_function): BCEWithLogitsLoss()
  )
)
```

Figure 6: The output layer structure.

output layer, a label attention weight matrix $U \in \mathbb{R}^{(p \times d) \times l}$ is created using a linear layer. For this thesis $p = 9$ for 9 convolutional channels and $d = 50$ for each of the channel output dimensions. $l = 6918$ is the total number of unique labels present in our training corpus. This makes the label weight matrix of shape (450, 6918). The attention weight matrix (α) is created by matrix multiplying H and U after passing through a softmax activation to get the input distribution. $H \in \mathbb{R}^{m \times (p \times d)}$ is the input matrix to the output layer from the multi-filter convolution layer. m represents the sequence number of input text. $\alpha \in \mathbb{R}^{m \times l}$ is the attention weight of each pair of labels and a word. The attention output ($V \in \mathbb{R}^{l \times (p \times d)}$) is obtained after multiplying the two matrix α^T and H. The output from the attention layer is finally fed to a linear layer with an in-channel of size $(p \times d)$ ad out-channel of size l. The linear layer creates a weight matrix $W \in \mathbb{R}^{(p \times d) \times l}$. The final output vector ($Y \in \mathbb{R}^{l \times l}$) for all labels is generated after the multiplication of V and W followed by the sum-pooling operation.

Implemented Model Structure:

The detailed structure of the full implemented model is shown in the below figures.

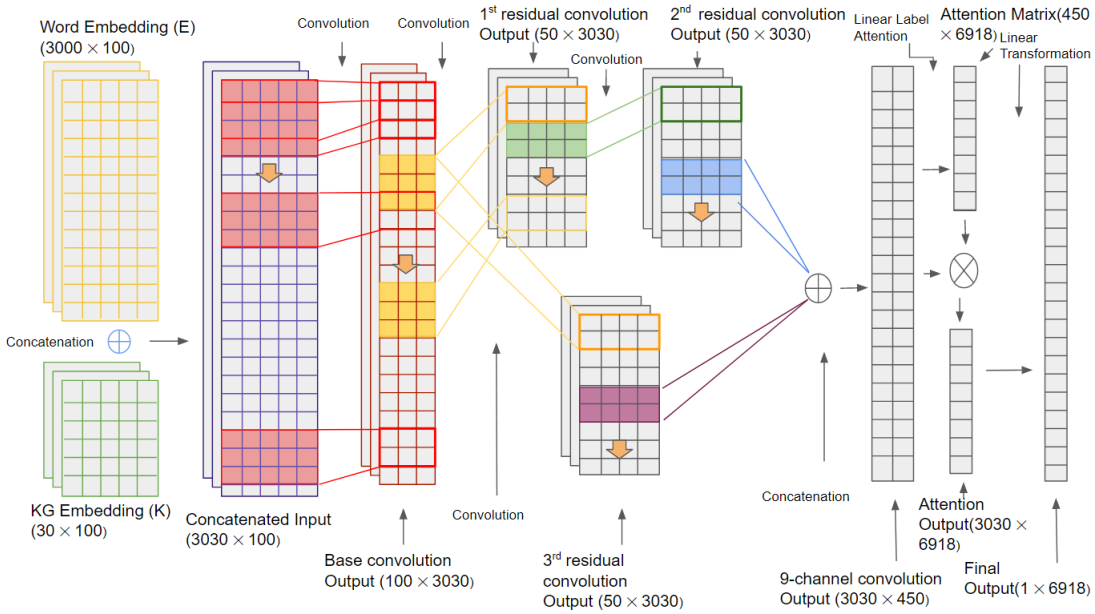

Figure 7: Full implemented architecture of "KG-MultiResCNN."

4 Data

4.1 EHR Data

This thesis targets to learn a model from EHR data that is a systematic collection of longitudinal patient's health information such as symptoms, medication, lab tests, procedures generated by each encounter of patients to the hospital or health care provider [55]. However, the structure of EHR data varies hugely with various clinical departments that are generating the data [140]. For example, a patient's diagnostic data generated by a doctor is very different from the diagnostic data generated by the lab test department. From all the generated EHR data, we can categorize them into three structural parts - structured, semi-structured, and unstructured [123]. We can further categorize EHR data into various data types like images and text. For the scope of this thesis, we are only considering unstructured textual EHR data. The figure below shows an example of EHR data from a structural point of view.

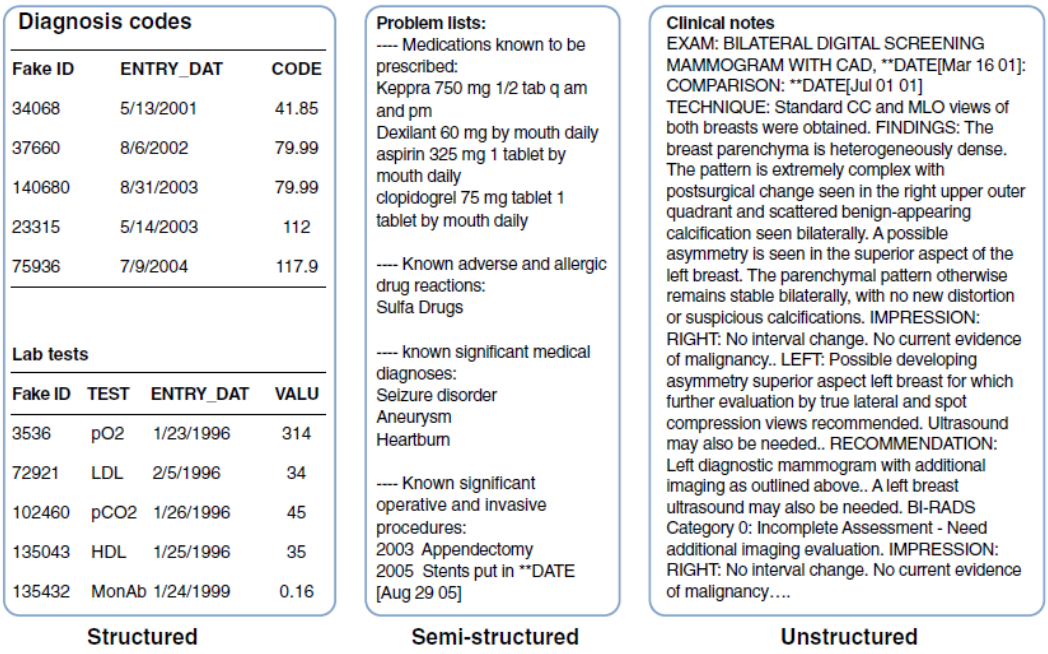

Figure 8: Different textual EHR data structure [123].

This thesis focuses on unstructured raw text data because there is plenty of raw EHR text data already available [14]. If a model learns to use raw text data, then the model can be used for any other raw text data.

4.2 Data Collection

Finding a suitable and available data source for this research is one of the major challenges of this thesis. Because of privacy reasons, most of the research works is done closely with the hospitals and the health service providers under specific terms and conditions and maintaining acceptable privacy policies. However, there are freely available data sources like CDC[10], n2c2(formally i2b2)[11][118] and Medical Information Mart for Intensive Care (MIMIC)[12] [2] [57] that can provide free health data. For our research, I choose to use the MIMIC database for several reasons. First of all, the MIMIC data is freely available. Secondly, there are a lot of research works [69][84][45][130][136] already done on this database so using this data will help in evaluating my approach against other approaches. Finally, MIMIC provides a huge number of textual data records for patients. MIMIC is an extensive, freely available database that contains thousands of patient details from their Intensive Care Unit (ICU) stay. The database is a part of PhysioNet[13] repository that is maintained by the MIT Laboratory for Computational Physiology[14] and supported by the National Institute of Biomedical Imaging and Bio-engineering (NIBIB)[15]. The data is collected from the USA's Beth Israel Deaconess Medical Center. Based on the collected data over the years from the same medical center, there are two versions of MIMIC data publicly available - MIMIC-II [2] and MIMIC-III[57]. The MIMIC-II is the old version of the data that contains over 10 thousand patients record from the ICU from the years between 2001 to 2008. Whereas the MIMIC-III is the newer database that contains the records for over 40 thousand patients who were admitted to the critical care unit between the years 2001 to 2012. For this thesis,s I have used the MIMIC-III database as it has more records than MIMIC-II, and it is comparatively new. However, the publicly available version of MIMIC-III[16] is a subset of the actual MIMIC-III data, and it contains about 100 patients information; moreover, it does not contain the free-text clinical notes data. Since this thesis is primarily based on free-text data, getting the whole MIMIC-III data along with the free text data was an essential step for this thesis. The full MIMIC-III data, specifically with the latest version(v1.4), is restricted for authorized use only. To get the restricted data, I had to complete a course on "Data or Specimens Only Research" under the "Human Research" curriculum group. The course is provided by The Collaborative Institutional Training Initiative (CITI Program)[17] which is a program to provide training on the trusted standard in human research, ethics, and data compliance. After finishing the training, CITI provided a certificate for the course under the Massachusetts Institute

[10]https://wonder.cdc.gov/DataSets.html
[11]https://n2c2.dbmi.hms.harvard.edu/
[12]https://physionet.org/content/mimic2-iaccd/1.0/
[13]https://physionet.org/
[14]https://lcp.mit.edu/
[15]https://www.nibib.nih.gov/
[16]https://physionet.org/content/mimiciii-demo/1.4/
[17]https://about.citiprogram.org/

of Technology affiliation. This particular certificate, along with a data compliance agreement called "PhysioNet Credentialed Health Data Use Agreement 1.5.0," was required to get the full version of MIMIC-III data. The CITI certificate is valid for three years for an applicant, so according to the agreement, it is ethical to use the data for three years by the applicant. After that period, a new certificate and a new agreement are required to use the data.

4.3 Data Description

The MIMIC-III[57] database is an extensive data of size 6.2GB (zipped). This immense database consists of 26 relation tables that contain de-identified patient information such as demographics, vital observations, measurements data made at the ICU, medications, procedures, laboratory test results, imaging reports, mortality, and clinical notes. In the entire database zip file, all these 26 tables are included as CSV files. All these files are linked with a foreign key identifier column. A suffix of "ID" can identify the foreign key identifiers at the end of the column. For example, a column with the name "SUBJECT_ID" is a unique value representation of a patient. All the files having the column "SUBJECT_ID" are related to each other with this foreign key identifier. All the 26 files can be categorized into mainly four categories. The first category is patients tracking. This category contains the files "ADMISSIONS", "CALLOUT.CSV", "ICUSTAYS.CSV", "PATIENTS.CSV", "SERVICES.CSV", "TRANSFERS.CSV". These files contain data related to a patient. The second category is critical care unit data. The files in this category are "CAREGIVERS.CSV", "CHARTEVENTS.CSV", "DATETIMEEVENTS", "INPUTEVENTS_CV.CSV", "INPUTEVENTS_MV.CSV", "NOTEEVENTS.CSV", "OUTPUTEVENTS.CSV", "PROCEDUREEVENTS_MV.CSV". These files contain the data collected while the patient was in the critical care unit. The third category is the hospital record system. This category contains the files "CPTEVENTS.CSV", "DIAGNOSES_ICD.CSV", "DRGCODES.CSV", "LABEVENTS.CSV", "MICROBIOLOGYEVENTS.CSV", "PRESCRIPTIONS.CSV", "PROCEDURES_ICD.CSV". These files contain the data that are collected for hospital record-keeping and billing to the patient [57]. The fourth category is the dictionary. The files in this category are stated with "D" to identify the dictionary files. The files are "D_CPT.CSV", "D_ICD_DIAGNOSES.CSV", "D_ICD_PROCEDURES.CSV", "D_ITEMS.CSV", "D_LABITEMS.CSV." These files are used as lookups for the shortcode used in other files. Out of these files, this thesis is particularly concerned about mainly the files named "NOTEEVENTS.CSV" and "DIAGNOSES_ICD.CSV." The "NOTEEVENTS.CSV" file contains the free-text notes from the doctors, medical staff, nurses, physicians. It also contains free-text notes from medical events such as radiology events, electrocardiography, echocardiography, and respiratory check events. With all these text notes, the "NOTEEVENTS.CSV" also contains the "Discharge summary" notes that contain a general description of the patient, starting from their medical history to final discharge notes. To be specific the "Discharge Summary" report contains the following record types : "HIS-

TORY OF PRESENT ILLNESS", "PAST MEDICAL HISTORY", "MEDICATIONS ON ADMISSION", "ALLERGIES", "FAMILY HISTORY", "SOCIAL HISTORY", "PHYSICAL EXAM AT TIME OF ADMISSION", "LABORATORY STUDIES", "BRIEF SUMMARY OF HOSPITAL COURSE", "DISCHARGE CONDITION", "DISCHARGE STATUS", "DISCHARGE MEDICATIONS", "FOLLOW-UP PLANS", "FINAL DIAGNOSES". However, these record types are not always uniform and in similar order throughout the data records simply because of the very nature of the free text data unstructured data. A typical example of "Discharge summary" note is shown in the figure 9. Past research works [54][130][136] have shown favor in using the "Discharge summary" notes as it contains an overall description of the patient's stay in the hospital. The text data in the "NOTEEVENTS.CSV" present in the combination of a patient (identified by "SUBJECT_ID") and a particular hospital admission (identified by "HADM_ID") for that patient. This relation between a patient and the admission makes it easier to combine the "NOTEEVENTS.CSV" with other files using the foreign key "SUBJECT_ID" and "HADM_ID." The other concerned file, "DIAGNOSES_ICD.CSV," contains the hospital-assigned diagnosis codes for patients during their stay at the hospital. The diagnosis code uses the International Statistical Classification of Diseases and Related Health Problems (ICD) system to represent a disease. In the clinical field, International Statistical Classification of Diseases and Related Health Problems (ICD) codes[92][37] are the identifiers that can describe a diagnosis and procedure done by a clinical institute for a patient. The idea of ICD is to provide an internationalized system for health management. The ICD system and its versions are managed by World Health Organization (WHO)[18] that coordinates and maintains the health system under the United Nations system [19]. To incorporate the changes in the medical field, the WHO updates the version of the codes periodically. The current running ICD version is 10, which was accepted by the World Health Assembly in 1990 [90]. In MIMIC-III data, the ICD code of the ninth revision is provided. To be specific, in MIMIC-III, the ICD code is called The International Classification of Disease and Clinical Modification version 9 (ICD-9-CM). ICD-9-CM is the American adaptation of the ICD-9 code and is the official diagnosis and procedures assign system in the USA[37]. The National Center of Health Statistics (NCHS) and the Centers of Medicare and Medicaid Services (CMS)[20] are responsible for managing and maintaining the ICD-9-CM codes. The ICD-9-CM contains more than 15,000 codes and follows a hierarchical code structure between the lengths 3 to 5. The codes with three digits are considered as the heading category, and an extra 1 or 2 digits are added after a decimal separator for more specific detail about the code[90]. For example[21], code 787 is the heading category for symptoms involving the digestive system. 787 code is further subdivided between 787.0 to 787.9 for more specific detail. For example, 787.2 is the code

[18]https://www.who.int/about
[19]https://www.un.org/en/
[20]https://www.cms.gov/
[21]http://www.icd9data.com/

for "Dysphagia," a sub-category under digestive system-related symptoms. 787.2 is then finally divided between 787.20 and 787.29 for even more specificity. For example, code 787.21 indicates "Dysphagia in the oral phase." The heading categories are mainly classified into surgical, diagnostic, and therapeutic procedures. The MIMIC-III "DIAGNOSES_ICD.CSV" file contains the ICD-9-CM diagnosis code for a patient's(indicated by "SUBJECT_ID") admission(indicated by "HADM_ID") with a sequence number (indicated by "SEQ_NUM"). Each time a patient is assigned to a diagnosis ICD-9-CM code, the sequence number is provided.

Admission Date: [**2118-6-2**] Discharge Date: [**2118-6-14**]

Date of Birth: Sex: F

Service: MICU and then to [**Doctor Last Name **] Medicine

HISTORY OF PRESENT ILLNESS: This is an 81-year-old female with a history of emphysema (not on home O2), who presents with three days of shortness of breath thought by her primary care doctor to be a COPD flare. Two days prior to admission, she was started on a prednisone taper and one day prior to admission she required oxygen at home in order to maintain oxygen saturation greater than 90%. She has also been on levofloxacin and nebulizers, and was not getting better, and presented to the [**Hospital1 18**] Emergency Room.

In the [**Hospital3 **] Emergency Room, her oxygen saturation was 100% on CPAP. She was not able to be weaned off of this despite nebulizer treatment and Solu-Medrol 125 mg IV x2.

Review of systems is negative for the following: Fevers, chills, nausea, vomiting, night sweats, change in weight, gastrointestinal complaints, neurologic changes, rashes, palpitations, orthopnea. Is positive for the following: Chest pressure occasionally with shortness of breath with exertion, some shortness of breath that is positionally related, but is improved with nebulizer treatment.

PAST MEDICAL HISTORY:
1. COPD. Last pulmonary function tests in [**2117-11-3**] demonstrated a FVC of 52% of predicted, a FEV1 of 54% of predicted, a MMF of 23% of predicted, and a FEV1:FVC ratio of 67% of predicted, that does not improve with bronchodilator treatment. The FVC, however, does significantly improve with bronchodilator treatment consistent with her known reversible air flow obstruction in addition to an underlying restrictive ventilatory defect. The patient has never been on home oxygen prior to this recent episode. She has never been on steroid taper or been intubated in the past.
2. Lacunar CVA. MRI of the head in [**2114-11-4**] demonstrates "mild degree of multiple small foci of high T2 signal within the white matter of both cerebral hemispheres as well as the pons, in the latter region predominantly to the right of midline. The abnormalities, while nonspecific in etiology, are most likely secondary to chronic microvascular infarction. There is no mass, lesion, shift of the normal midline strictures or hydrocephalus. The major vascular flow patterns are preserved. There is moderate right maxillary, moderate bilateral ethmoid, mild left maxillary, minimal right sphenoid, and frontal sinus mucosal thickening. These abnormalities could represent an allergic or some other type of inflammatory process. Additionally noted is a moderately enlarged subtotally empty sella turcica".
3. Angina: Most recent stress test was in [**2118-1-3**] going for four minutes with a rate pressure product of 10,000, 64% of maximum predicted heart rate without evidence of ischemic EKG changes or symptoms. The imaging portion of the study demonstrated no evidence of myocardial ischemia and a calculated ejection fraction of 84%. The patient denies angina at rest and gets angina with walking a few blocks. Are alleviated by sublingual nitroglycerin.
4. Hypothyroidism on Synthroid.
5. Depression on Lexapro.
6. Motor vehicle accident with head injury approximately 10 years ago.

Figure 9: An example of "Discharge summary" note from MIMIC-III data.

5 Experiment

5.1 Data Pre-Processing

As discussed in section 4, for this thesis, I am using MIMIC-III data. As a universal approach, this thesis inputs raw heterogeneous text. The MIMIC-III data also provides patients' clinical notes in a raw text format. However, ML models cannot work with raw text data. Hence, I used data pre-processing to create machine readable representation out of the unstructured clinical text as a necessary step. The data processing follows a sequential step. In the first step, I Followed the work of Fei Li and Hong Yu [69]. I used "DIAGNOSES_ICD.csv" and "NOTEEVENTS.csv" to get the ICD codes and clinical notes respectively for a patient using their SUBJECT_ID. In their original work, Fei Li and Hong Yu [69] used "PROCEDURES_ICD.csv" along with "DIAGNOSES_ICD.csv" to predict the procedural ICD codes and disease ICD codes. However, unlike a general ICD code prediction task of Fei Li and Hong Yu [69], this thesis focuses on predicting disease ICD codes. Because of this reason, I only used the "DIAGNOSES_ICD.csv" to train the model so that it can predict only disease ICD codes. Following their process, two data frames are created from "DIAGNOSES_ICD.csv" and "NOTEEVENTS.csv" using the pandas[22] library of Python. Then the two data frames are merged for the same patient (identified by "SUBJECT_ID") and same hospital admission (identified by "HADM_ID"). The "NOTEEVENTS.csv" file contains following category of notes namely "Discharge summary", "Echo", "ECG", "Nursing", "Physician ", "Rehab Services", "Case Management ", "Respiratory", "Nutrition", "General", "Social Work", "Pharmacy", "Consult", "Radiology", and "Nursing/other". Fei Li and Hong Yu [69] in their work only considered the "Discharge Summary" reports. Table 1 shows the overall code coverage of MIMIC-III notes.

Coverage	Patients	Hospital Admission	Diag. ICD-9 Codes
MIMIC-III (All)	46520	58976	6984
NOTEEVENTS.CSV (all categories)	46146	58361	6967
"Discharge Summary" notes	41127	52726	6918

Table 1: MIMIC-III Code coverage statistics [54].

However, for this thesis, I am considering not only "Discharge summary" but also "Nursing" and "Physician " reports. Because in the MIMIC dataset, there are 59652 records for discharge summaries, whereas for "Nursing" and "Physician" notes, there are 223556 and 141624 records, respectively. Also, the "Nursing" and "Physician"

[22]https://pandas.pydata.org/

notes contain detailed and vital observations and symptoms of a patient, which are lacking in "Discharge Summaries." Therefore, all the notes from "Discharge Summaries," "Nursing," and "Physician" are combined to create one combined note for a unique combination of the patient ("SUBJECT_ID") and admission ("HADM_ID"). To this end, I created three different data frames while concatenating the different note categories for this experiment. The first one contains the notes only from "Discharge summary." The second one is from "Nursing" and "Physician" and the last one from all three categories. The reason is to compare and check how removing "Discharge summary" and adding "Physician" and "Nursing" notes affect the ICD code prediction. The notes then go through the process of tokenization, where I am tokenizing the whole text into words. The process also involves removing unwanted texts (de-identified names, dates) and lower casing the words. The concatenated and processed notes are then joined with the disease ICD codes for the unique combination of the patient ("SUBJECT_ID") and admission ("HADM_ID") to create a complete record. To this end, the three concatenated data frame contains 52726, 9070, and 52993 records, respectively.

5.2 Generating Word Embedding

Since the primary task of this thesis is text classification, the text is the main input to our model. With that goal in mind, I have discussed how the text is tokenized into a list of words in the previous step. However, these tokenized words can not be fitted into a model simply because of the fact that machine learning models only understand numbers. So it was necessary to transform the words into numbers so that the model could be trained. Following the work of Fei Li and Hong Yu [69], I used pretrained word2vec[83] model to create an embedding vector for the words. For word embedding creation, I used the texts from the training set after the whole dataset was split into train, validation, and test. The data splitting technique is discussed in section 5.5. The word embedding generation process involves the prior task of creating a vocabulary. I used the training dataset for vocabulary creation and tokenized all the texts into word tokens. The vocabulary can be defined as $V = \{v_1, v_2, ..., v_n\}$ where, n is the number of unique words in the whole dataset. A document representation is created using the vocabulary. A document record can be represented as $D = \{d_1, d_2, d_3, ..., d_n\}$ where $d_i \in \{0, \mathbb{R}^+\}, i \in \{0, n\}$ is the i^{th} dictionary word and the value can be 0 or any positive non-zero real number $\mathbb{R}$ depending on the number of occurrence of the word in the document record. Then a document matrix is created for each record with their corresponding document vocab representation. The document matrix can be represented as $C = \{D_1, D_2, D_3, ...D_k\} \in \mathbb{R}^{k \times n}, C^T \in \mathbb{R}^{n \times k}$ where k is the total number of record present in the dataset. Following the work of Fei Li and Hong Yu [69], I have considered the records with words that have appeared at least three different records. All the terms that have occurred in less than three records are considered rare terms. The rare terms are deleted from the document matrix (C^T) to create the final document matrix. Using the indices of the

document matrix, a new vocabulary is created. Next, the indices numbers of the rare terms are taken from the document matrix, and the corresponding positional words from the vocabulary(V) are deleted to create the final vocabulary. After the vocabulary is created, the word2vec model is trained to create the word embedding for all the words in the vocabulary. To do that, I used all the sentences from all the records in the dataset so that the word embedding has contextual representations. Then, following the same approach of Fei Li and Hong Yu [69], I used the word2vec model to train on the entire training dataset, with five epochs to create 100-dimensional embedding vectors for each word. After the training, I saved each word's embedding into a file to create the word embedding layer for the model. The word embedding layer creation is discussed in section 3.2.1. In the next step, we extracted the medical entities from the text to use separate embedding for the words as a weighting factor to the word embedding.

5.3 Entity Extraction

As an additional knowledge to the model, this thesis provides some extra information about the model. The extra information is generated by collecting named entities. The intuition is that certain entities carry maximum information in a text for a text classification task. The thesis employs an entity extraction method to collect meaningful entities from the text. Since the work is on clinical text data, I used a pre-trained entity extraction model to extract medical entity types, namely "symptoms," "treatment," "test." The model is provided by Huggingface [23] which is a huge library of Transformer[120] based models. "samrawal/bert-base-uncased_clinical-ner"[24] is the entity extraction model from the Huggingface library that I am using for this thesis. The model is trained on the n2c2(formerly i2b2)[25]/VA challenge dataset[118]. The dataset was released as part of the "Fourth i2b2/VA Shared-Task and Workshop Challenges in Natural Language Processing for Clinical Data"[26] challenge. One of the tasks of this challenge was to extract medical problems, tests, and treatments. The data for this challenge was provided by MIMIC-II [102] [42], a clinical database from Berth Israel Deaconess Medical Center, and the University of Pittsburgh Medical Center. Discharge summary and progress notes are used from these two databases to create the final dataset for the challenge. All the records in the dataset are fully de-identified for privacy reasons and manually annotated for the concept, assertion, and relation information extraction. The used model for our entity extraction task has an overall accuracy of 94%, macro f1 score of 87%, and weighted f1 score of 94%. For extracting "problem", "test", and "treatment" the model has f1 score of 87%, 85%, and 86% respectively. After using the model on our dataset, it identifies medically important entities related to "problem," "test," and

[23]https://huggingface.co/
[24]https://huggingface.co/samrawal/bert-base-uncased_clinical-ner
[25]https://n2c2.dbmi.hms.harvard.edu/
[26]https://www.i2b2.org/NLP/Relations/

"treatment." These entities are then passed to the next step to generate knowledge graph embeddings.

5.4 Generating KG Embedding

In this thesis, I am using knowledge graph embedding of medically sound entities as a weighting factor or extra attention on the word embedding. For this task, I used PyTorch BigGraph (PGB)[67] which is an embedding system provided by Meta Research[27] (formerly Facebook Research) community. The system learns the node and edges representations of massive knowledge graphs and embeds the nodes and relations in the graph. The system aims to keep the similar entities/nodes of a knowledge graph closer to each other in vector space while pushing apart unrelated nodes/entities. In other words, the embedding of similar nodes will be similar so that they can be closer to each other in the vector space. PGB was evaluated on the large knowledge graphs of Freebase[28], LiveJournal [68], YouTube[114] and Twitter[16] social network graph. The result shows that PBG outperformed on large knowledge graphs. As part of their research, the PGB also trained on the large Wikidata[29] knowledge graph with 78 million entities and 4,131 relations. Moreover, the Meta Research community has made the embedding for the entities publicly available[30]. The embedding dimension for each entity is 200, and the whole file with the embedding of 78 million entities takes about 36Gib of memory space. Ideally, the whole file can be used in an embedding layer to create an embedding matrix, and for a sequence of entities, it can provide the embedding for each entity. However, loading this massive file in an embedding layer would require a huge space of GPU memory if we wish to train the model in GPU. Therefore, as an effective and efficient approach, I used the extracted medical entities (discussed in section 5.3) from the whole dataset to query through the whole 78 million entity embedding and extracted the embedding for those medical entities. Then I saved those embedding and created a separate file of comparatively more diminutive size. This approach allowed me to use the new file easily to create an embedding layer (see figure 4) for the model. To this end, I have collected 11247 medical entities from the whole dataset, and for each of those entities, I have a 200-dimensional embedding vector obtained from the knowledge graph embedding. In the next step, the splitting strategy is discussed.

5.5 Splitting Strategy

In the earlier works by Fei et al. [69], and Mullenbach et al. [84], they used a static technique to split the data into train set, validation set, and test set. For their work, they used three static files for each set. The files are namely "train_full_hadm_ids.CSV",

[27]https://github.com/facebookresearch
[28]Google, Freebase Data Dumps, https://developers.google.com/freebase, Sept. 10, 2018.
[29]https://www.wikidata.org/wiki/Wikidata:Main$_{Page}$
[30]https://github.com/facebookresearch/PyTorch-BigGraphpre-trained-embeddings

"dev_full_hadm_ids.CSV", and "test_full_hadm_ids.CSV". These files contain a static list of "HADM_ID." While splitting the whole data into train, validation, and test set, they referred to their corresponding static hadm_ids list file. In that way, the datasets contain the records with the "HADM_ID" mentioned in their corresponding static hadm_ids file list. However, I did not follow this process; rather, we used a dynamic splitting approach to the data. I used the standard approach for the training dataset and randomly collected 70% of the complete data. I did a 50-50 split from the reaming data to randomly select 50% of the remaining data as the validation set, and the rest 50% is taken as test data. So in terms of the full data, the data split ratio is 70:15:15 for train, validation, and test set. The reason for choosing this particular ratio is that the model's performance was the same when I used the static hadm_ids files. Another reason to use this splitting technique was to incorporate the "HADM_IDs" that are not present in the static hadm_ids files. For the experiment where I used the "Physician" and "Nursing" notes along with "Discharge summary" notes, there were some new "HADM_IDs" that came to the dataset because for those "HADM_IDs" there were no "Discharge Summary" reports. To incorporate those new "HADM_IDs," the standard splitting approach fitted better.

5.6 Loss Function

Since the task is multi-label binary classification, in this thesis used I used the binary cross-entropy as the loss function to calculate the loss. The loss function can be formulated as below.

$$Loss(D, Y, \theta) = -\sum_{j=1}^{|L|} Y_j log(\tilde{Y}_j) + (1 - Y_j) log(1 - \tilde{Y}_j)$$

Here, D is the document of word sequence, L is the total number of labels, θ is the trainable parameters of the model, and $\tilde{Y} \in \mathbb{R}_0^1$ is the predicted output with the value of any real number between 0 and 1. The output is then converted to binary output $\in 0, 1$ using a predefined threshold such as 0.5. Commonly this is done by simply rounding off the output value. The training process typically aims to minimize the cross-entropy loss between the ground truth(Y) value and the predicted output($\tilde{Y}$) using an optimizer. This thesis experimented with Adam[60] and AdamW[77] optimizer for the model training.

5.7 Hyper-parameter Tuning

The "KG-MultiResCNN" model is a complex model with multiple hyper-parameters. Hence, a trade-off amongst the hyper-parameters was required to create the optimal performing model. To this end, this thesis utilizes several hyper-parameter tuning options, such as the number of CNN channels, dimension of the word embedding, a learning rate scheduler, number of medical KG entities used, number of tokens used, gradient clipping, batch size, and dropout rate selection.

I started the hyper-parameter tuning empirically following the earlier work of "MultiResCNN" [69] by Li and Yu. Following their work, "KG-MultiResCNN" used a learning rate of 0.0001, batch size of 16, a dropout rate of 0.2, and six CNN channels with the filter size of 3,5,9,15,19,25. However, multiple experiments revealed that the "KG-MultiResCNN" reaches the optimal performance when using the batch size of 6 and CNN channels with nine filters of size 3,5,7,9,13,15,17,23,29. For the learning rate, I used a learning rate scheduler[107] such as "StepLR" that used an initial learning rate of 0.001; then, after every 8th epoch step, it decreases the learning rate by a factor of 10. This thesis also explored the gradient clipping technique that can handle exploding or vanishing gradients while training. However, the exploding and vanishing gradient problems are often observed in RNN based models. Since my model is not RNN based, the experiment did not show significant performance improvement with gradient clipping. The thesis also experimented with the word embedding dimensions between 100 and 200, and it is found that the model worked better with the 100-dimensional word embedding vector generated by the Word2Vec[83] model. While experimenting for the optimal number of medical KG entities, it is found that the MIMIC-III discharge summary contains an average of 28 entities per clinical note. Hence, the model performed best when applying 30 medical KG entities. Finally, for the number of word tokens, I followed "MultiResCNN"[69] to use 2500 medical entities. However, while experimenting, it is discovered that the discharge summary notes contain an average of 1878 tokens per note and the model performance showed optimal when using 3000 tokens for the experiments among the lengths of 2500, 3000, 3500. In addition, the model also implemented an early stopping mechanism that ensures stopping the training automatically if there is no improvement for the patience of 10 epochs.

5.8 Baselines

To complete the experiment process, the following conventional and deep learning models are compared against the "KG-MultiResCNN" model.

- **Logistic Regression**: As an experiment, Mullenbach et al. [84] used Logistic Regression (LR) to predict ICD codes using unigram bag-of-words vector for all words in the MIMIC-III text data.
- **SVM**: Perotte et al. [94] experimented with hierarchical and flat ICD code prediction tasks using Support Vector Machine (SVM). Text notes from MIMIC-II data found that the hierarchical prediction worked better than flat ICD prediction. Later Xie et al. [128] used the SVM for hierarchical ICD code prediction on MIMIC-III data. Their model performed moderately with 10,000 unigram word vectors and with Tf-idf weighting.
- **CNN**: Kim et al.[59] was the pioneer for developing a 1-D convolutional neural network for sentence classification. Mullenbach et al. [84] experimented with

the performance of 1D-CNN on classifying ICD codes from MIMIC-III clinical notes.

- **Bi-GRU**: Cho et al.[22] first introduced the Bidirectional Gated Recurrent Neural Network (Bi-GRU) for text classification. Later Mullenbach et al. [84] achieved modest success while applying the Bi-GRU for ICD classification with MIMIC-III clinical notes.

- **C-MemNN**: Prakash et al.[99] introduced the The Condensed Memory Neural Network (C-MemNN) that used the memory network[99] and iterative condensed memory network[109] together. The model claimed to achieve a good result on the MIMIC-III 50 code dataset. However, the model was not evaluated with the F1-score.

- **C-LSTM-Att**: Shi et al.[105] used an LSTM based language model called the Character-aware LSTM-based Attention (C-LSTM-Att). The model used an attention mechanism to handle the mismatch between notes and ICD codes. The model was used to predict the top 50 ICD codes from the MIMIC-III dataset.

- **LEAM**: Wang et al.[122] proposed a text classification model called the Label Embedding Attentive Model (LEAM) that predicts the top 50 ICD codes from the MIMIC-III dataset. The model used projects the embedding of words and labels in the same latent vector space and calculates the similarities between the embeddings.

- **CAML**: Mullenbach et al. [84] introduced the Convolutional Attention Network for Multi-Label classification (CAML) for ICD code classification using MIMIC-III notes. The model used one convolution layer and a label attention mechanism. The model achieved high performance for multi-label ICD code classification.

- **DR-CAML**: As an extension of CAML, Mullenbach et al. [84] introduced the Description Regularized CAML (DR-CAML). The model used the text description of the codes for better prediction accuracy.

- **MultiResCNN**: As the current state-of-the-art, the Multi-Filter Residual Convolutional Neural Network (MultiResCNN) was introduced by Li and Yu[69]. The model used multiple convolution filters followed by one residual neural network to predict ICD codes from free-text clinical notes. In addition, the model used a label attention mechanism for better prediction accuracy. As a result, the model achieved high performance in predicting full and top 50 ICD codes from MIMIC-III clinical notes.

6 Evaluation

The proposed model is a binary multi-class classifier that means the predicted results for different ICD codes will be in the range of zero to one. The prediction value for an ICD code should be near 1 for the patients at higher risk to the disease corresponding to that ICD code. It is customary to evaluate this kind of binary decision model at a range of thresholds $p_\tau \in [0; 1]$ for the decision $p > p_\tau$ and then report the results in the form of receiver operating characteristic (ROC) curves, area under ROC (AUROC). However, for our case, even though discrimination is an important statistical property, it may not properly address clinical usefulness [89] [79] [113], [115] [121] [108]. For example, a false negative decision can cause greater harm than a false positive decision. In that case, a model with high sensitivity may be preferable to a model with high specificity and low sensitivity, even though the model might have, say, a higher AUROC. In general terms, a model is clinically useful if its decisions for patients lead to a better ratio between benefits and harms than not using the model. In this kind of situation, to evaluate a model, we needed to make sure the model should have good precision and recall both. Given all these conditions, f1-score is considered to be the best metric for our model as it is the harmonic mean between precision and recall. Another reason to choose the f1-score as the primary evaluation metric is because of the comparison flexibility amongst other models. Past research works and baseline models used f1-score as their primary evaluation metric. So, to compare my result with other models' results, it was necessary to use f1-score. As the task is a multi-label prediction, micro and macro averaging strategies are adopted for better computation of the average score among different labels. Since there can be more samples for a particular label, the micro averaging strategy seemed important as it calculates the average aggregating each label's contribution. Whereas, in the macro averaging strategy, each label is treated independently and then takes the average overall labels.

The model evaluation criteria are set as below.

- **Based on Codes count:** As mentioned by Huang et al.[54] in their paper, it is observed from the MIMIC-III data that ICD codes follow the Zip's law pattern. That means most of the notes contribute to only a few ICD Codes. Two sets of data were prepared for the evaluation by adopting the baseline approach to evaluate the model based on the code count. The first dataset for evaluation contains the notes that cover all the ICD codes present in the whole MIMIC-III dataset. Whereas, the second one is a subset of the whole notes set that covers the top 50 frequently occurring ICD codes. The table 2 shows that out of all the 15 notes category present in the MIMIC-III data, only "Discharge summary" notes itself cover about 90% of the total codes present in the data. Table 1 shows that total of 52726 "Discharge summary" notes for each unique hospital admission covers total 6918 unique diagnosis ICD codes. Whereas, out of all "Discharge summary" notes almost 85% of the notes cover only the top 50 ICD codes, making most of the ICD codes very rare. The two created

datasets are split into train, validation, and test set. Table 3 shows the data split values for both full and 50 codes. Separate experiments are carried on the model for both datasets. The datasets are applied on the baseline approach setting to compare the results between the baseline approach and the proposed approach of this thesis. For the baseline approach "MultiResCNN"[69], the word embeddings are prepared by training the word2vec model. The embeddings are then passed through six parallel CNN networks with filter sizes of 3,5,9,15,19,25. Each CNN filter used one residual network to pass the data into a label attention layer and the final classifier layer. Whereas, for this thesis modeling approach, the process is followed as mentioned in the experiment section (section 5). This evaluation approach allowed the models to compare based on code prediction tasks.

Dataset	Hospital Admission	Discharge Summary Coverage (%)
full codes (6918 codes)	52726	90.34
top-50 codes	49354	84.56

Table 2: MIMIC-III Code coverage statistics only for "Discharge summary" notes [54].

Data Split		Samples	Average codes
full-label set	train	36906	5.40
	validation	7910	5.40
	test	7910	5.41
50-label set	train	34547	5.40
	validation	7404	5.40
	test	7403	5.40

Table 3: Data split configuration for full-codes and top-50 codes with "Discharge summary" notes only.

- **Based on Notes type:** The proposed model of the thesis is also evaluated on the basis of note types. In the MIMIC-III dataset, there are 14 categories of notes available. Out of those 14 categories, past approaches only used the

notes with the category "Discharge summary." However, as a different evaluation approach, this thesis utilizes the "Physician" and "Nursing" category notes along with the "Discharge summary" notes. The intuition is that the "Nursing" and "Physician" notes are detailed observation reports of a patient's admission. These reports can provide valuable information on patients' health conditions and help in better ICD code prediction. To use the "Nursing" and "Physician" notes for model evaluation, this thesis approached to create three more datasets:
1) The first dataset contains only "Discharge summary" notes. (Table 2)
2) The second one contains data from "Discharge summary," "Nursing," and "Physician" notes together. (Table 4)
3) The final dataset contains text data only from "Nursing" and "Physician" notes. (Table 5)

Dataset	Hospital Admission	DS+Nursing+ Physician Coverage (%)
full codes (6919 codes)	52985	90.78
top-50 codes	49555	84.91

Table 4: Code coverage statistics for "Discharge summary"+"Nursing"+"Physician" notes.

Dataset	Hospital Admission	Nursing+ Physician Coverage (%)
full codes (4216 codes)	9070	15.54
top-50 codes	8306	14.74

Table 5: Code coverage statistics for "Nursing"+"Physician" notes.

All the datasets followed the same experimental setting mentioned in the experiment section (section 5). The data splitting strategy for the combined "Discharge summary", "Nursing," and "Physician" notes are shown in table 6. And the splitting strategy for only "Nursing" and "Physician" notes are shown in table 7.

Data Split		Samples	Average codes
full-label set	train	37088	5.41
	validation	7948	5.41
	test	7949	5.41
50-label set	train	34688	5.41
	validation	7434	5.41
	test	7433	5.41

Table 6: Data split configuration for full-codes and top-50 codes with "Discharge summary"+"Nursing"+"Physician" notes.

Data Split		Samples	Average codes
full-label set	train	6349	5.44
	validation	1360	5.44
	test	1361	5.43
50-label set	train	6022	5.43
	validation	1290	5.43
	test	1291	5.44

Table 7: Data split configuration for full-codes and top-50 codes with "Nursing"+"Physician" notes.

- **Performance:** As an evaluation criterion, this thesis also compares the model's computational cost based on different categories such as the total number of trainable parameters, number of epochs, and the average time taken for a training epoch. These different categories indicate how complex a model is. For example, a very complex model is expected to have a longer training time than a comparatively less complex one. On the other hand, a complex model tends to converge faster than a less complex model for the same learning rate. That means a complex model takes a fewer number of epochs to converge. Furthermore, a higher number of epochs would eventually over-fit the model. In the result section (section), the performance efficiency of the

model for different settings is compared to find the best model setting. The results section also discusses the performance comparison between the baseline and proposed models.

- **Diagnosis ICD and Procedural ICD prediction:** Since this thesis mainly focused on disease diagnosis, it approached the task by predicting only diagnosis ICD codes. However, the baseline approaches used a generic approach to predict ICD codes that include both diagnosis ICD codes as well as procedural ICD codes. This thesis experimented with an additional data processing step to include procedural ICD codes for a fair comparison between the baseline and proposed models. Following the work of Fei Li and Hong Yu [69], this thesis includes the "PROCEDURAL_ICD.CSV" file to get the procedural ICD codes for each hospital admission of a patient. The procedural ICD codes and the diagnosis ICD codes for a patient admission are combined to form the final list of ICD codes for the unique hospital admission of a patient. The new ICD codes list is then used as the prediction labels for a discharge summary input text. Table 8 shows the ICD codes values when using only diagnosis ICD and both diagnosis and procedural ICD codes. Finally, the data is split into train, validation, and test set following a similar data splitting approach (section 5.5). The new dataset is used in different modeling approaches to get comparative results.

Coverage	Patients	Hospital Admission	ICD-9 Codes
"Discharge Summary" notes (Diagnosis ICD)	41127	52726	6918
"Discharge Summary" notes (Diagnosis + Procedural ICD)	41127	52726	8917

Table 8: MIMIC-III Diagnosis and Procedural code coverage statistics.

7 Results

This section describes the results of each category mentioned in the Evaluation section (section 6). Following the baseline model approach of "MultiResCNN"[69], each experiment is conducted for different random parameter initialization seeds. The average value of multiple experiment results with a standard deviation is shown in the tables.

- **Based on Codes count:** As mentioned in the evaluation section, the proposed "KG-MultiResCNN" model is evaluated against the baseline approach of "MultiResCNN" to predict the diagnosis ICD codes using the "Discharge summary" notes. Table 9 and table 10 show the result comparison between the two approaches. It can be seen from the result shown in table 9 that "KG-MultiResCNN" achieved better macro and micro f1-score compared to the state-of-the-art baseline model "MultiResCNN." When applied to the "full code" dataset, "KG-MultiResCNN" achieved the micro F1-score of 53.8%, increasing a margin of 1.1 over the state-of-the-art. Whereas, for the macro F1-score comparison, "KG-MultiResCNN" achieved 10.2%, outperforming the state-of-the-art by a margin of 1.95. Similarly, table 10 shows that when applied to the "50-code" dataset, "KG-MultiResCNN" increased with a margin of 1.46 over the state-of-the-art for micro F1-score comparison. Whereas, for macro F1-score comparison, "KG-MultiResCNN" achieved a score of 64.21%, a significant increase with a margin of 3.11 over the state-of-the-art. The results also show a stable standard deviation for both the "full-codes" and "50 codes" experiment

Model	Full Codes	
	F1-Score (Micro) (%)	F1-Score (Macro) (%)
MultiResCNN	52.7	8.25
KG-MultiResCNN	**53.8** $\pm$0.1	**10.2** $\pm$ 0.1

Table 9: Models result comparison for full diagnosis ICD code with "Discharge summary" notes. $\pm$ indicates standard deviations.

- **Based on Notes type:** As discussed in the evaluation section, the "KG-MultiResCNN" model is experimented with and evaluated with multiple note types. Since all the past research and the state-of-the-art model only used "Discharge summary" notes, it was essential to see how the "KG-MultiResCNN" model works with other notes types. Table 11 shows a comparative results for the experiment of "KG-MultiResCNN" model with different notes combination for the

Model	Top-50 Codes	
	F1-Score (Micro) (%)	F1-Score (Macro) (%)
MultiResCNN	67.6	61.1
KG-MultiResCNN	**69.06** $\pm$0.1	**64.21** $\pm$ 0.1

Table 10: Models result comparison for top-50 diagnosis ICD code with "Discharge summary" notes. $\pm$ indicates standard deviations.

full code prediction setting. The result shows that the model performed the best when using only "Discharge summary" notes. The model worked rather poorly for the combination of "Physician" and "Nursing" notes. This is because the combination of "Physician" and "Nursing" notes does not cover a good amount of codes in MIMIC-III data. Table 5 indicates that the "Physician" and "Nursing" note coverage is very less. This means, even if there are massive numbers of "Physician" and "Nursing" notes present in MIMIC-III data, they actually cover a very less number of patient's hospital admission. For the "Discharge summary" notes combined with "Nursing" and "Physician" notes, the model was assumed to have a good result. However, the model showed a limitation when applying the three notes combination as it performed poorly against the experiment with only "Discharge summary" notes. When applying the three notes combination, the model's micro F1-score decreased with a margin of 0.4, and the macro accuracy decreased with a margin of 1.4. A possible reason would be that the model was optimally set to handle a text of length 3000 tokens. Furthermore, when all three types of notes were combined, the notes' length became huge, and the model needed more tokens to get the proper text inference. However, using more tokens usually depends on the GPU computational capacity and model complexity. A more sophisticated model with more layers would work better with a large number of word tokens. To maintain the scope and time limitation of the thesis, I did not investigate more on using a large number of tokens. Table 12 shows the similar result when applying the model with the three notes combination for top 50 code prediction setting. In 50 code prediction also, the model performed the best with only "Discharge summary" notes. The reason for this is the same as discussed for the full code setting. As a future improvement, a more sophisticated model that can use more word tokens can be investigated.

Notes Type	Full Codes	
	F1-Score (Micro) (%)	F1-Score (Macro) (%)
"Discharge summary" notes	**53.8**	**10.2**
"Discharge summary"+ "Nursing"+"Physician" notes	53.4	8.8
"Nursing"+"Physician" notes	30.5	2.46

Table 11: "KG-MultiResCNN" result comparison for full diagnosis ICD code with multiple note combinations.

Notes Type	Top-50 Codes	
	F1-Score (Micro) (%)	F1-Score (Macro) (%)
"Discharge summary" notes	**69.06**	**64.21**
"Discharge summary"+ "Nursing"+"Physician" notes	68.19	61.83
"Nursing"+"Physician" notes	48.32	38.13

Table 12: "KG-MultiResCNN" result comparison for top-50 diagnosis ICD code with multiple note combinations.

- **Performance:** Table 13 shows the comparison between "KG-MultiResCNN" and the state-of-the-art baseline for different categories as mentioned in the evaluation section (section 6). The result showed that "KG-MultiResCNN" took only 15 epochs to converge, whereas the state-of-the-art "MultiResCNN" took 26 epochs to converge. The result also showed that both the models have the same number of training parameters. However, "KG-MultiResCNN" took about 2185 seconds for each epoch to finish whereas, "MultiResCNN" took about a half of the time for "KG-MultiResCNN." A possible explanation would be that the complexity of the "MultiResCNN" model is much less than the "KG-MultiResCNN" model. The "KG-MultiResCNN" model is more complex and sophisticated as it has two embedding layers, unlike the baseline model's only one embedding layer. Then the baseline model used six convolution channels, whereas "KG-MultiResCNN" used nine. Finally, instead of one residual

layer, the "KG-MultiResCNN" used two residual layers for each convolution channel.

Categories	MultiResCNN	KG-MultiResCNN
Trainable Parameters (million)	11.9	11.9
Training Time (seconds/epoch)	1026	2185
No of epochs	26	15

Table 13: Model performance comparison based on categories.

- **Diagnosis ICD and Procedural ICD prediction:** As discussed in the evaluation section (section 6), this thesis mainly focused on predicting only diagnosis ICD codes using free-text data. However, for the sake of fairness, additional data processing and experiments were carried out to compare with the baseline approaches that are modeled to predict both diagnosis and procedural ICD codes.
  Table 14 shows the comparative results between the baseline approaches (section 5.8) and "KG-MultiResCNN". It is evident from the results that "KG-MultiResCNN" significantly outperformed over all the baseline approaches, including current state-of-the-art "MultiResCNN." Even with the full diagnosis and procedural ICD coding setting, "KG-MultiResCNN" acquired a micro F1-score of 56.1%, surpassing the state-of-the-art "MultiResCNN" by a margin of 0.9. Whereas, for the macro F1-score, "KG-MultiResCNN" acquired a score of 10.2, significantly surpassing the "MultiResCNN" by a margin of 1.7.
  Similarly, the table 15 shows the comparative result between "KG-MultiResCNN" and the baseline approaches in top 50 diagnosis and procedural ICD code prediction setting. The result depicts that even in 50 code prediction settings, the "KG-MultiResCNN" significantly outperformed the baseline models, including the state-of-the-art "MultiResCNN." "KG-MultiResCNN" achieved the micro F1-score of 69.5%, an increase of margin of 2.5 over "MultiResCNN." Whereas, for macro F1-score, "KG-MultiResCNN" obtained a score of 64.5%, greatly surpassing the state-of-the-art "MultiResCNN" by a margin of 3.9.
  The results also show a stable standard deviation for both the "full-codes" and "50 codes" experiments for diagnosis and procedural ICD code prediction.

Model	Full Codes	
	F1-Score (Micro) (%)	F1-Score (Macro) (%)
LR[84]	27.2	1.1
Flat SVM[94]	39.7	-
Hierarchy SVM[94]	44.1	-
CNN[59]	41.9	4.2
Bi-GRU[22]	41.7	3.8
CAML[84]	53.9	8.8
DR-CAML[84]	52.9	8.6
MultiResCNN[69]	55.2	8.5
KG-MultiResCNN	**56.1** $\pm$0.1	**10.2** $\pm$ 0.1

Table 14: Models result comparison for full diagnosis and procedural ICD code with "Discharge summary" notes. $\pm$ indicates standard deviations.

Model	Top-50 Codes	
	F1-Score (Micro) (%)	F1-Score (Macro) (%)
LR[84]	53.3	47.7
C-LSTM-Att[105]	53.2	-
CNN[59]	62.5	57.6
Bi-GRU[22]	54.9	48.4
LEAM[122]	61.9	54.0
CAML[84]	61.4	53.2
DR-CAML[84]	63.3	57.6
MultiResCNN[69]	67.0	60.6
KG-MultiResCNN	**69.5** $\pm$0.1	**64.5** $\pm$ 0.1

Table 15: Models result comparison for top-50 diagnosis and procedural ICD code with "Discharge summary" notes. $\pm$ indicates standard deviations.

8 Discussion and Limitation

This thesis showed quite an improvement over past research in diagnosis ICD code prediction from clinical free-text data. Particularly with the discharge summary notes, the model performed the best in predicting the diagnosis ICD codes. This thesis chose the discharge summary notes because the discharge summary notes cover almost 85% of patients' hospital admission. Furthermore, the discharge summary notes contain vital information about the patient, such as demographics, history of illness, health condition during admission, and other lab test information. The proposed model of this thesis can be viewed as the extension of "MultiResCNN" [69] that used six CNN channels with a residual block for a multi-label text classification task. In particular, in this thesis, the "MultiResCNN" [69] model is improved by adding an additional embedding layer. The extra embedding layer provides the knowledge graph embedding of medically significant entities from the text. Additionally, the proposed model used nine convolution channels and two residual blocks for better feature representation. The model does the job of differential diagnosis in a multi-label binary classification setting by predicting significant diagnosis ICD codes as 1 and rejecting the non-significant codes as 0. Several experiments were done on the model to find the optimal operation setting of the model. For the input word embedding, the model was tested with 100-dimensional and 200-dimensional word embedding vectors, and it was revealed that the model worked better with 100-dimensional embedding vectors. The number of word tokens played a significant role as well. The model experimented with 2500, 3000, and 3500 word tokens. It turned out the model performed better with a maximum of 3000 word tokens from the clinical notes. Since the word embedding vector and the KG embedding vector of the medical terms combined serve as the input to the model, the thesis experimented for the optimal number of medical entities to be used from the text. The experiment showed that using a maximum of 30 medical entities provides the best performing result. This research also discovered that the number of CNN channels and residual blocks hugely impact the model's performance. The experiments showed that for a higher number of word tokens, a higher number of CNN channels perform better. To this end, in this thesis, for 3000 word tokens, nine CNN channels performed best. The experiments also showed that the model performed optimally with two residual layers for the combined input of word embeddings and KG embeddings. The model showed comparable computational cost against the state-of-the-art "MultiResCNN" [69] with training time about twice than that of "MultiResCNN" [69]. This is reasonable given the fact that the thesis model is more complex with more CNN channels and double the number of residual blocks. Irrespective of the success, the experiments revealed some important limitations in this thesis.

Limitations

The experiment indicated that for a higher number of word tokens, a higher number of CNN channels and a higher number of residual layers increase the model performance. This finding encouraged me to include "Nursing" and "Physician" notes along with "Discharge summary" notes for this research. However, more CNN channels and more residual layers mean higher model complexity. Moreover, loading a complex model requires higher GPU memory. Unfortunately, the GPU memory was limited for this thesis, which restricted the model from performing well with the colossal size of notes. Moreover, this limitation restricted me from creating the model with a maximum of nine CNN channels and two residual blocks. The thesis also spends a considerable amount of time researching Bert[31] model. This thesis implemented three versions of the Bert model that uses sentence embedding, word embedding from the last hidden layer, and combined word embedding for the last four hidden layers for the clinical notes classification task. However, the Bert models poorly predicted ICD codes from clinical discharge summary notes.

9 Conclusion and Future Improvements

This thesis introduces "KG-MultiResCNN," a multi-channel convolutional network model for multi-label disease diagnosis prediction. The model performs the disease diagnosis by automatically predicting the ICD codes related to free-text EHR notes. As a universal approach, this thesis utilized the free-text EHR notes because the notes are unstructured in nature and do not maintain any standard guideline. The model finds essential features from the clinical notes to predict the ICD codes significant for that note. Furthermore, to strengthen the vital feature extraction, the model utilizes the Tf-idf weighting of each word in the text. The weighting is done by multiplying (scalar vector multiplication) the Tf-idf score of each word to their corresponding word embedding vector. The model also utilizes the knowledge graph embeddings of medically significant word tokens from the Wikidata knowledge graph as a novel approach. A pre-trained Bert model is used for the medical NER task. The NER model extracts the medical entities from the text. The medical entities are then used to fetch their corresponding knowledge graph representation from a knowledge graph embedding system called "PyTorch-BigGraph." The "PyTorch-BigGraph" is trained on the Wikidata knowledge graph, and that can produce a knowledge graph representation for about 78 million entities. The combined representation of word embeddings and knowledge graph embeddings of medical entities is used in the proposed model of this thesis for the prediction of diagnosis ICD codes. This thesis used a differential diagnosis approach for disease prediction where the model predicts the relevant ICD codes while rejecting the non-related codes for a clinical note. The model also used a label attention mechanism where a label-specific weighting strategy was adopted. The label attention allowed the model to find the essential words for an ICD code and focus on those words for better prediction.

Several experiments are done on the model with MIMIC-III clinical notes. The experiments revealed that the model with knowledge graph embedding surpasses the result of the state-of-the-art model. Furthermore, further experiments disclosed that additional convolution channels and additional residual layers significantly improved the model's performance. As a comparative study, the proposed model is evaluated against the current state-of-the-art "MultiResCNN" to predict the diagnosis ICD codes. The result indicated that the proposed model significantly outperformed the current state-of-the-art. It is also found that in MIMIC-III data, the top 50 ICD codes are covered by about 85% of all discharge summary notes. Empirically a top-50 codes dataset was also prepared for the model evaluation, and the results showed that even for the top-50 codes, the proposed model outperformed the "MultiResCNN" model. For the fair comparison between the state-of-the-art "MultiResCNN," this thesis utilized the procedural ICD codes combined with the diagnosis ICD codes from MIMIC-III data. Two new datasets are prepared further for the full-code prediction and top-50 code prediction. The result showed that even for the combined diagnosis and procedural ICD code prediction task, the proposed

model significantly improved over the state-of-the-art for both full-code and top-50 code prediction. Irrespective of the success of the thesis, there lies a good ground of improvement over the current thesis model.

Future improvements

Following the lines of Li and yu[69], the thesis could improve significantly with the proper inclusion of Bert embedding. To be specific, exploring recurrent Transformer[29] and hierarchical BERT[138] for the ICD code prediction seems to be exciting research. Furthermore, while experimenting with this thesis model, it is found that the ICD codes follow a structured hierarchy. For example, the ICD code 084.8 indicates the disease "Blackwater fever," a sub-disease under the parent disease "Malaria" of ICD code 084. Since MIMIC-III provides the exact or the most specific ICD codes, the thesis model is trained to predict the specific diseases such as "Blackwater fever" mentioned in the last example. However, an exciting approach could be to predict the parent ICD code when the system cannot precisely predict the actual ICD code. Therefore, a hierarchical ICD prediction task sounds like a compelling approach for the future. Another future improvement of this thesis could be to include a clinical knowledge graph. In this thesis, I used the Wikidata knowledge graph, which is considered generic as it contains all kinds of entities. Therefore, a clinical knowledge graph with only clinical entities will better serve the prediction task.

Acknowledgment

My sincere acknowledgment goes to my first supervisor, Prof. Dr. Matthias Thimm, who trusted me with the topic and provided all the necessary bits of help. Special thanks and gratitude goes to my second supervisor, Dr. Zeyd Boukhers, who thoroughly guided me throughout this thesis by providing crucial inputs, suggestions, and ideas during our weekly meetings. He was always available and kept his mind open in understanding any challenges I faced during my thesis. I would also like to thank Mr. Korok Sengupta, a researcher from the Institute for Parallel and Distributed Systems at the University of Stuttgart, for helping me write my thesis proposal. Furthermore, a special thank you to Ms. Shreya Chatterjee for helping me structure my thesis, correcting errors, and being the pillar of strength through my highs and lows throughout the tenure of my thesis. Finally, I thank my parents for their constant support and encouragement.

Abbreviations

Bi-GRU Bidirectional Gated Recurrent Unit. 36, 46, 47

CAML Convolutional Attention Network for Multi-Label Classification. 3–5, 13, 14, 36, 46, 47

CNN Convolutional Neural Network. 2–4, 7, 9–15, 17, 19, 21, 22, 34–36, 38, 46–49

DL Deep Learning. iv, 2–4, 8–10, 12–14

DNN Deep Neural Network. iv, 2, 8, 11

DR-CAML Description Regularized CAML. 5, 36, 46, 47

EHR Electronic Health Records. iv, 1–4, 6–8, 10–12, 24, 50, 54

ICD International Statistical Classification of Diseases and Related Health Problems. iv, 3–6, 12–15, 27, 30, 31, 35–37, 39, 41–51, 55

ICU Intensive Care Unit. 10, 11, 25, 26

KG-MultiResCNN Knowledge Guided Multi-Filter Residual Convolutional Neural Network. iv, 5, 6, 15, 23, 34, 35, 42–47, 50, 54, 55

LR Logistic Regression. 35, 46, 47

LSTM Long Short-Term Memory. 10–13, 36

MIMIC Medical Information Mart for Intensive Care. iv, 5, 9, 12–14, 25–30, 32, 35–38, 41, 43, 50, 51, 54, 55

MultiResCNN Multi-Filter Residual Convolutional Neural Network. 4, 5, 14, 35, 36, 38, 42–48, 50

NER Named Entity Recognition. 50

NLP Natural Language Processing. 2, 7, 8

RNN Recurrent Neural Network. 2, 3, 11, 12, 35

SVM Support Vector Machine. 9, 12, 35, 46

Tf-idf Term Frequency-Inverse Document Frequency. 5, 15, 16, 21, 35, 50

WHO World Health Organization. 12, 27

List of Figures

List of Tables

References

[1] Global innovation index. 2019. four ways data is improving healthcare. https://www.weforum.org/agenda/2019/12/four-ways-data-isimproving-healthcare/.

[2] Laboratory for computational physiology, m. i. t. the mimic-iii clinical database. physionet.,. *https://doi.org/10.13026/C2XW26 (2015).*

[3] T. o. of the national coordinator for health information technology. adoption of electronic health record systems among u.s. non-federal acute care hospitals: 2008-2015, year = 2016.

[4] The uk strategy for rare diseases. *https://assets.publishing.service.gov.uk/government/uploads/system/*

[5] Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1, 2004.

[6] Julia Adler-Milstein, Catherine M DesRoches, Peter Kralovec, Gregory Foster, Chantal Worzala, Dustin Charles, Talisha Searcy, and Ashish K Jha. Electronic health record adoption in us hospitals: progress continues, but challenges persist. *Health affairs*, 34(12):2174–2180, 2015.

[7] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)*, pages 1–6. Ieee, 2017.

[8] 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.

[9] Peter C Austin, Jack V Tu, Jennifer E Ho, Daniel Levy, and Douglas S Lee. Using methods from the data-mining and machine-learning literature for disease classification and prediction: a case study examining classification of heart failure subtypes. *Journal of clinical epidemiology*, 66(4):398–407, 2013.

[10] Anand Avati, Kenneth Jung, Stephanie Harman, Lance Downing, Andrew Ng, and Nigam H Shah. Improving palliative care with deep learning. *BMC medical informatics and decision making*, 18(4):55–64, 2018.

[11] Tian Bai and Slobodan Vucetic. Improving medical code prediction from clinical text via incorporating online knowledge sources. In *The World Wide Web Conference*, pages 72–82, 2019.

[12] Tal Baumel, Jumana Nassour-Kassis, Raphael Cohen, Michael Elhadad, and Noémie 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.

[13] Brett K Beaulieu-Jones, Casey S Greene, et al. Semi-supervised learning of the electronic health record for phenotype stratification. *Journal of biomedical informatics*, 64:168–178, 2016.

[14] Guthrie S Birkhead, Michael Klompas, and Nirav R Shah. Uses of electronic health records for public health surveillance to advance public health. *Annual review of public health*, 36:345–359, 2015.

[15] Olivier Bodenreider. The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, 32(suppl_1):D267–D270, 2004.

[16] Paolo Boldi, Marco Rosa, Massimo Santini, and Sebastiano Vigna. Layered label propagation: A multiresolution coordinate-free ordering for compressing social networks. In *Proceedings of the 20th international conference on World Wide Web*, pages 587–596, 2011.

[17] Taxiarchis Botsis, Gunnar Hartvigsen, Fei Chen, and Chunhua Weng. Secondary use of ehr: data quality issues and informatics opportunities. *Summit on Translational Bioinformatics*, 2010:1, 2010.

[18] Alex Bottle and Paul Aylin. Intelligent information: a national system for monitoring clinical performance. *Health services research*, 43(1p1):10–31, 2008.

[19] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8(1):1–12, 2018.

[20] Zhilu Chen, Jing Wang, Haibo He, and Xinming Huang. A fast deep learning system using gpu. In *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1552–1555. IEEE, 2014.

[21] Yu Cheng, Fei Wang, Ping Zhang, and Jianying Hu. Risk prediction with electronic health records: A deep learning approach. In *Proceedings of the 2016 SIAM International Conference on Data Mining*, pages 432–440. SIAM, 2016.

[22] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[23] Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F Stewart, and Jimeng Sun. Doctor ai: Predicting clinical events via recurrent neural networks. In *Machine learning for healthcare conference*, pages 301–318. PMLR, 2016.

[24] Edward Choi, Andy Schuetz, Walter F Stewart, and Jimeng Sun. Medical concept representation learning from electronic health records and its application on heart failure prediction. *arXiv preprint arXiv:1602.03686*, 2016.

[25] Edward Choi, Andy Schuetz, Walter F Stewart, and Jimeng Sun. Using recurrent neural network models for early detection of heart failure onset. *Journal of the American Medical Informatics Association*, 24(2):361–370, 2017.

[26] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(ARTICLE):2493–2537, 2011.

[27] Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. Very deep convolutional networks for text classification. *arXiv preprint arXiv:1606.01781*, 2016.

[28] Andrew M Dai and Quoc V Le. Semi-supervised sequence learning. *arXiv preprint arXiv:1511.01432*, 2015.

[29] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019.

[30] Luciano RS de Lima, Alberto HF Laender, and Berthier A Ribeiro-Neto. A hierarchical approach to the automatic categorization of medical documents. In *Proceedings of the seventh international conference on Information and knowledge management*, pages 132–139, 1998.

[31] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[32] Parr DG. Patient phenotyping and early disease detection in chronic obstructive pulmonary disease. *Proc Am Thorac Soc.*, 8(4):(3):338–49, 2011 Aug;.

[33] Pedro Domingos. A few useful things to know about machine learning. *Communications of the ACM*, 55(10):78–87, 2012.

[34] Shahram Ebadollahi, Jimeng Sun, David Gotz, Jianying Hu, Daby Sow, and Chalapathy Neti. Predicting patient's trajectory of physiological data using temporal trends in similar patients: a system for near-term prognostics. In *AMIA annual symposium proceedings*, volume 2010, page 192. American Medical Informatics Association, 2010.

[35] R Scott Evans. Electronic health records: then, now, and in the future. *Yearbook of medical informatics*, (Suppl 1):S48, 2016.

[36] Marcelo Fiszman, Wendy W Chapman, Dominik Aronsky, R Scott Evans, and Peter J Haug. Automatic detection of acute bacterial pneumonia from chest x-ray reports. *Journal of the American Medical Informatics Association*, 7(6):593–604, 2000.

[37] National Center for Health Statistics (US) and Council on Clinical Classifications. *The International classification of diseases, 9th revision, clinical modification: ICD-9-CM*, volume 2. US Department of Health and Human Services, Public Health Service, Health . . . , 1980.

[38] Leopold Franz, Yash Raj Shrestha, and Bibek Paudel. A deep learning pipeline for patient diagnosis prediction using electronic health records. *arXiv preprint arXiv:2006.16926*, 2020.

[39] Jason Alan Fries. Brundlefly at semeval-2016 task 12: Recurrent neural networks vs. joint inference for clinical temporal information extraction. *arXiv preprint arXiv:1606.01433*, 2016.

[40] Andrea Frome, Greg Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. 2013.

[41] Sebastian Gehrmann, Franck Dernoncourt, Yeran Li, Eric T Carlson, Joy T Wu, Jonathan Welt, John Foote Jr, Edward T Moseley, David W Grant, Patrick D Tyler, et al. Comparing deep learning and concept extraction based methods for patient phenotyping from clinical narratives. *PloS one*, 13(2):e0192360, 2018.

[42] 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.

[43] Benjamin A Goldstein, Ann Marie Navar, Michael J Pencina, and John Ioannidis. Opportunities and challenges in developing risk prediction models with electronic health records data: a systematic review. *Journal of the American Medical Informatics Association*, 24(1):198–208, 2017.

[44] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[45] 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. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23, 2021.

[46] Varun Gulshan, Lily Peng, Marc Coram, Martin C Stumpe, Derek Wu, Arunachalam Narayanaswamy, Subhashini Venugopalan, Kasumi Widner, Tom Madams, Jorge Cuadros, et al. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *Jama*, 316(22):2402–2410, 2016.

[47] Krystl Haerian, Hojjat Salmasian, and Carol Friedman. Methods for identifying suicide or suicidal ideation in ehrs. In *AMIA annual symposium proceedings*, volume 2012, page 1244. American Medical Informatics Association, 2012.

[48] Marja Härkänen, Katri Vehviläinen-Julkunen, Trevor Murrells, Jussi Paananen, Bryony D Franklin, and Anne M Rafferty. The contribution of staffing to medication administration errors: A text mining analysis of incident report data. *Journal of Nursing Scholarship*, 52(1):113–123, 2020.

[49] 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):e0215049, 2019.

[50] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[51] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[52] William R Hersh. Adding value to the electronic health record through secondary use of data for quality assurance, research, and surveillance. *Clin Pharmacol Ther*, 81:126–128, 2007.

[53] A Hoerbst and E Ammenwerth. Electronic health records. *Methods Inf Med*, 49(4):320–336, 2010.

[54] Jinmiao Huang, Cesar Osorio, and Luke Wicent Sy. An empirical evaluation of deep learning for icd-9 code assignment using mimic-iii clinical notes. *Computer methods and programs in biomedicine*, 177:141–153, 2019.

[55] 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):395–405, 2012.

[56] Min Jiang, Yukun Chen, Mei Liu, S Trent Rosenbloom, Subramani Mani, Joshua C Denny, and Hua Xu. A study of machine-learning-based approaches to extract clinical entities and their assertions from discharge summaries. *Journal of the American Medical Informatics Association*, 18(5):601–606, 2011.

[57] Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-Wei, 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(1):1–9, 2016.

[58] Soo Yeon Kim, Saehoon Kim, Joongbum Cho, Young Suh Kim, In Suk Sol, Youngchul Sung, Inhyeok Cho, Minseop Park, Haerin Jang, Yoon Hee Kim, et al. A deep learning model for real-time mortality prediction in critically ill children. *Critical care*, 23(1):1–10, 2019.

[59] Yoon Kim. Convolutional neural networks for sentence classification, 2014.

[60] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[61] Bevan Koopman, Guido Zuccon, Anthony Nguyen, Anton Bergheim, and Narelle Grayson. Automatic icd-10 classification of cancers from free-text death certificates. *International journal of medical informatics*, 84(11):956–965, 2015.

[62] Leah S Larkey and W Bruce Croft. Combining classifiers in text categorization. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 289–297, 1996.

[63] Pedro Larranaga, Borja Calvo, Roberto Santana, Concha Bielza, Josu Galdiano, Inaki Inza, José A Lozano, Rubén Armananzas, Guzmán Santafé, Aritz Pérez, et al. Machine learning in bioinformatics. *Briefings in bioinformatics*, 7(1):86–112, 2006.

[64] Simon Meyer Lauritsen, Mads Ellersgaard Kalør, Emil Lund Kongsgaard, Katrine Meyer Lauritsen, Marianne Johansson Jørgensen, Jeppe Lange, and Bo Thiesson. Early detection of sepsis utilizing deep learning on electronic health record event sequences. *Artificial Intelligence in Medicine*, 104:101820, 2020.

[65] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[66] June-Goo Lee, Sanghoon Jun, Young-Won Cho, Hyunna Lee, Guk Bae Kim, Joon Beom Seo, and Namkug Kim. Deep learning in medical imaging: general overview. *Korean journal of radiology*, 18(4):570–584, 2017.

[67] Adam Lerer, Ledell Wu, Jiajun Shen, Timothee Lacroix, Luca Wehrstedt, Abhijit Bose, and Alex Peysakhovich. Pytorch-biggraph: A large-scale graph embedding system. *arXiv preprint arXiv:1903.12287*, 2019.

[68] Jure Leskovec, Kevin J Lang, Anirban Dasgupta, and Michael W Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, 6(1):29–123, 2009.

[69] Fei Li and Hong Yu. Icd coding from clinical text using multi-filter residual convolutional neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8180–8187, 2020.

[70] Qi Li, Kristin Melton, Todd Lingren, Eric S Kirkendall, Eric Hall, Haijun Zhai, Yizhao Ni, Megan Kaiser, Laura Stoutenborough, and Imre Solti. Phenotyping for patient safety: algorithm development for electronic health record based automated adverse event and medical error detection in neonatal intensive care. *Journal of the American Medical Informatics Association*, 21(5):776–784, 2014.

[71] Huiying Liang, Brian Y Tsui, Hao Ni, Carolina CS Valentim, Sally L Baxter, Guangjian Liu, Wenjia Cai, Daniel S Kermany, Xin Sun, Jiancong Chen, et al. Evaluation and accurate diagnoses of pediatric diseases using artificial intelligence. *Nature medicine*, 25(3):433–438, 2019.

[72] Znaonui Liang, Gang Zhang, Jimmy Xiangji Huang, and Qmming Vivian Hu. Deep learning for healthcare decision making with emrs. In *2014 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 556–559. IEEE, 2014.

[73] Zachary C Lipton, David C Kale, Charles Elkan, and Randall Wetzel. Learning to diagnose with lstm recurrent neural networks. *arXiv preprint arXiv:1511.03677*, 2015.

[74] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

[75] Jingshu Liu, Zachariah Zhang, and Narges Razavian. Deep ehr: Chronic disease prediction using medical notes. In *Machine Learning for Healthcare Conference*, pages 440–464. PMLR, 2018.

[76] Yue Liu, Tao Ge, Kusum S Mathews, Heng Ji, and Deborah L McGuinness. Exploiting task-oriented resources to learn word embeddings for clinical abbreviation expansion. *arXiv preprint arXiv:1804.04225*, 2018.

[77] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[78] Xinbo Lv, Yi Guan, Jinfeng Yang, and Jiawei Wu. Clinical relation extraction with deep learning. *International Journal of Hybrid Information Technology*, 9(7):237–248, 2016.

[79] Kevin McGeechan, Petra Macaskill, Les Irwig, and Patrick MM Bossuyt. An assessment of the relationship between clinical utility and predictive ability measures and the impact of mean risk in the population. *BMC medical research methodology*, 14(1):1–12, 2014.

[80] Stéphane Meystre and Peter J Haug. Natural language processing to extract medical problems from electronic clinical documents: performance evaluation. *Journal of biomedical informatics*, 39(6):589–599, 2006.

[81] 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.

[82] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*, 2010.

[83] 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.

[84] 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.

[85] Shyamala G Nadathur. Maximising the value of hospital administrative datasets. *Australian Health Review*, 34(2):216–223, 2010.

[86] Anthony N Nguyen, Michael J Lawley, David P Hansen, Rayleen V Bowman, Belinda E Clarke, Edwina E Duhig, and Shoni Colquist. Symbolic rule-based classification of lung cancer stages from free-text pathology reports. *Journal of the American Medical Informatics Association*, 17(4):440–445, 2010.

[87] Phuoc Nguyen, Truyen Tran, Nilmini Wickramasinghe, and Svetha Venkatesh. Deepr: a convolutional net for medical records. *IEEE journal of biomedical and health informatics*, 21(1):22–30, 2016.

[88] Paul Nickerson, Patrick Tighe, Benjamin Shickel, and Parisa Rashidi. Deep neural network architectures for forecasting analgesic response. In *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2966–2969. IEEE, 2016.

[89] MM Nugues and CM Roberts. Coral mortality and interaction with algae in relation to sedimentation. *Coral reefs*, 22(4):507–516, 2003.

[90] US Department of Health, Human Services, et al. Hhs proposes adoption of icd-10 code sets and updated electronic transaction standards, 2008.

[91] Kimberly J O'malley, Karon F Cook, Matt D Price, Kimberly Raiford Wildes, John F Hurdle, and Carol M Ashton. Measuring diagnoses: Icd code accuracy. *Health services research*, 40(5p2):1620–1639, 2005.

[92] World Health Organization et al. International classification of diseases:[9th] ninth revision, basic tabulation list with alphabetic index. 1978.

[93] Jeffrey Pennington, Richard Socher, and Christopher D 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.

[94] Adler Perotte, Rimma Pivovarov, Karthik Natarajan, Nicole Weiskopf, Frank Wood, and Noémie Elhadad. Diagnosis code assignment: models and evaluation metrics. *Journal of the American Medical Informatics Association*, 21(2):231–237, 2014.

[95] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018.

[96] Trang Pham, Truyen Tran, Dinh Phung, and Svetha Venkatesh. Deepcare: A deep dynamic memory model for predictive medicine. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 30–41. Springer, 2016.

[97] Arturo López Pineda, Ye Ye, Shyam Visweswaran, Gregory F Cooper, Michael M Wagner, and Fuchiang Rich Tsui. Comparison of machine learning classifiers for influenza detection from emergency department free-text reports. *Journal of biomedical informatics*, 58:60–69, 2015.

[98] Ryan Poplin, Avinash V Varadarajan, Katy Blumer, Yun Liu, Michael V McConnell, Greg S Corrado, Lily Peng, and Dale R Webster. Prediction of cardiovascular risk factors from retinal fundus photographs via deep learning. *Nature Biomedical Engineering*, 2(3):158–164, 2018.

[99] Aaditya Prakash, Siyuan Zhao, Sadid A Hasan, Vivek Datla, Kathy Lee, Ashequl Qadir, Joey Liu, and Oladimeji Farri. Condensed memory networks for clinical diagnostic inferencing. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

[100] Nathan D Ratliff, David Silver, and J Andrew Bagnell. Learning to search: Functional gradient techniques for imitation learning. *Autonomous Robots*, 27(1):25–53, 2009.

[101] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[102] Mohammed Saeed, Mauricio Villarroel, Andrew T Reisner, Gari Clifford, Li-Wei Lehman, George Moody, Thomas Heldt, Tin H Kyaw, Benjamin Moody, and Roger G Mark. Multiparameter intelligent monitoring in intensive care ii (mimic-ii): a public-access intensive care unit database. *Critical care medicine*, 39(5):952, 2011.

[103] Elyne Scheurwegs, Kim Luyckx, Léon Luyten, Walter Daelemans, and Tim Van den Bulcke. Data integration of structured and unstructured sources for assigning clinical codes to patient stays. *Journal of the American Medical Informatics Association*, 23(e1):e11–e19, 2016.

[104] 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.

[105] 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.

[106] Benjamin Shickel, Patrick James Tighe, Azra Bihorac, and Parisa Rashidi. Deep ehr: a survey of recent advances in deep learning techniques for electronic health record (ehr) analysis. *IEEE journal of biomedical and health informatics*, 22(5):1589–1604, 2017.

[107] Leslie N Smith. Cyclical learning rates for training neural networks. In *2017 IEEE winter conference on applications of computer vision (WACV)*, pages 464–472. IEEE, 2017.

[108] Ewout W Steyerberg and Yvonne Vergouwe. Towards better clinical prediction models: seven steps for development and an abcd for validation. *European heart journal*, 35(29):1925–1931, 2014.

[109] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. *arXiv preprint arXiv:1503.08895*, 2015.

[110] Harini Suresh, Nathan Hunt, Alistair Johnson, Leo Anthony Celi, Peter Szolovits, and Marzyeh Ghassemi. Clinical intervention prediction and understanding with deep neural networks. In *Machine Learning for Healthcare Conference*, pages 322–337. PMLR, 2017.

[111] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[112] Kenji Suzuki. Overview of deep learning in medical imaging. *Radiological physics and technology*, 10(3):257–273, 2017.

[113] Rajesh Talluri and Sanjay Shete. Using the weighted area under the net benefit curve for decision curve analysis. *BMC medical informatics and decision making*, 16(1):1–9, 2016.

[114] Lei Tang and Huan Liu. Scalable learning of collective behavior based on sparse social dimensions. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1107–1116, 2009.

[115] Kevin Ten Haaf, Jihyoun Jeon, Martin C Tammemägi, Summer S Han, Chung Yin Kong, Sylvia K Plevritis, Eric J Feuer, Harry J de Koning, Ewout W Steyerberg, and Rafael Meza. Risk prediction models for selection of lung cancer screening candidates: a retrospective validation study. *PLoS medicine*, 14(4):e1002277, 2017.

[116] Truyen Tran, Tu Dinh Nguyen, Dinh Phung, and Svetha Venkatesh. Learning vector representation of medical objects via emr-driven nonnegative restricted boltzmann machines (enrbm). *Journal of biomedical informatics*, 54:96–105, 2015.

[117] Özlem Uzuner, Ira Goldstein, Yuan Luo, and Isaac Kohane. Identifying patient smoking status from medical discharge records. *Journal of the American Medical Informatics Association*, 15(1):14–24, 2008.

[118] Özlem Uzuner, Brett R South, Shuying Shen, and Scott L DuVall. 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association*, 18(5):552–556, 2011.

[119] Ankit Vani, Yacine Jernite, and David Sontag. Grounded recurrent neural networks. *arXiv preprint arXiv:1705.08557*, 2017.

[120] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[121] Andrew J Vickers and Angel M Cronin. Everything you always wanted to know about evaluating prediction models (but were too afraid to ask). *Urology*, 76(6):1298–1301, 2010.

[122] Guoyin Wang, Chunyuan Li, Wenlin Wang, Yizhe Zhang, Dinghan Shen, Xinyuan Zhang, Ricardo Henao, and Lawrence Carin. Joint embedding of words and labels for text classification. *arXiv preprint arXiv:1805.04174*, 2018.

[123] Wei-Qi Wei and Joshua C Denny. Extracting research-quality phenotypes from electronic health records to support precision medicine. *Genome medicine*, 7(1):1–14, 2015.

[124] Nilmini Wickramasinghe. Deepr: a convolutional net for medical records. 2017.

[125] Laura K Wiley, Jeremy D Moretz, Joshua C Denny, Josh F Peterson, and William S Bush. Phenotyping adverse drug reactions: Statin-related myotoxicity. *AMIA Summits on Translational Science Proceedings*, 2015:466, 2015.

[126] Yonghui Wu, Min Jiang, Jianbo Lei, and Hua Xu. Named entity recognition in chinese clinical text using deep neural network. *Studies in health technology and informatics*, 216:624, 2015.

[127] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

[128] Xiancheng Xie, Yun Xiong, Philip S Yu, and Yangyong Zhu. Ehr coding with multi-scale feature attention and structured knowledge graph propagation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 649–658, 2019.

[129] Hua Xu, Zhenming Fu, Anushi Shah, Yukun Chen, Neeraja B Peterson, Qingxia Chen, Subramani Mani, Mia A Levy, Qi Dai, and Josh C Denny. Extracting and integrating data from entire electronic health records for detecting colorectal cancer cases. In *AMIA Annual Symposium Proceedings*, volume 2011, page 1564. American Medical Informatics Association, 2011.

[130] Keyang Xu, Mike Lam, Jingzhi Pang, Xin Gao, Charlotte Band, Piyush Mathur, Frank Papay, Ashish K Khanna, Jacek B Cywinski, Kamal Maheshwari, et al. Multimodal machine learning for automated icd coding. In *Machine Learning for Healthcare Conference*, pages 197–215. PMLR, 2019.

[131] Zhen Yang, Matthias Dehmer, Olli Yli-Harja, and Frank Emmert-Streib. Combining deep learning with token selection for patient phenotyping from electronic health records. *Scientific reports*, 10(1):1–18, 2020.

[132] Liang Yao, Chengsheng Mao, and Yuan Luo. Clinical text classification with rule-based features and knowledge-guided convolutional neural networks. *BMC medical informatics and decision making*, 19(3):31–39, 2019.

[133] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent trends in deep learning based natural language processing. *ieee Computational intelligenCe magazine*, 13(3):55–75, 2018.

[134] Mahmood Yousefi-Azar and Len Hamey. Text summarization using unsupervised deep learning. *Expert Systems with Applications*, 68:93–105, 2017.

[135] Lin Yue, Dongyuan Tian, Weitong Chen, Xuming Han, and Minghao Yin. Deep learning for heterogeneous medical data analysis. *World Wide Web*, pages 1–23, 2020.

[136] Dongdong Zhang, Changchang Yin, Jucheng Zeng, Xiaohui Yuan, and Ping Zhang. Combining structured and unstructured data for predictive models: a deep learning approach. *BMC medical informatics and decision making*, 20(1):1–11, 2020.

[137] Lei Zhang, Shuai Wang, and Bing Liu. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1253, 2018.

[138] Xingxing Zhang, Furu Wei, and Ming Zhou. Hibert: Document level pre-training of hierarchical bidirectional transformers for document summarization. *arXiv preprint arXiv:1905.06566*, 2019.

[139] Di Zhao and Chunhua Weng. Combining pubmed knowledge and ehr data to develop a weighted bayesian network for pancreatic cancer prediction. *Journal of biomedical informatics*, 44(5):859–868, 2011.

[140] Jing Zhao, Panagiotis Papapetrou, Lars Asker, and Henrik Boström. Learning from heterogeneous temporal data in electronic health records. *Journal of biomedical informatics*, 65:105–119, 2017.