

**NOVEL DATA MINING ALGORITHMS FOR ANALYSIS OF ELECTRONIC  
HEALTH RECORDS**

---

A Dissertation  
Submitted to  
the Temple University Graduate Board

---

In Partial Fulfillment  
of the Requirements for the Degree of  
DOCTOR OF PHILOSOPHY

---

by  
Ashis Kumar Chanda  
December 2022

Examining committee members:

Dr. Slobodan Vucetic, Advisory Chair, Dept. of Computer and Information Sciences  
Dr. Zoran Obradovic, Dept. of Computer and Information Sciences  
Dr. Eduard Dragut, Dept. of Computer and Information Sciences  
Dr. Kevin A. Henry, External Member, Dept. of Geography and Urban Studies

# ABSTRACT

Medical health providers use electronic health records (EHRs) to store information about patient treatment to support patient care management and securely share health information among healthcare organizations. EHRs have also been used in healthcare research in problems such as patient phenotyping, health risk prediction, and medical entity extraction. In this thesis, we focus on several important issues: (1) how to convert natural text from medical notes to vector representations suitable for deep learning algorithms, (2) how to help healthcare researchers select a patient cohort from EHRs, and (3) how to use EHRs to identify patient diagnoses and treatments.

In the first part of the thesis, we present a new method for learning vector representations of medical terms. Learning vector representations of words is an important pre-processing step in many natural language processing applications. For example, EHRs contain clinical notes that describe patient health conditions and course of treatment in a narrative style. The notes contain specialized medical terminology and many abbreviations. Learning good vector representations of specialized medical terms can improve the quality of downstream data analysis tasks on EHR data. However, the traditional approaches struggle to learn vector representations of rarely used medical terms. To overcome this problem, we developed a neural network-based approach, called `definition2vec`, that uses external knowledge contained in medical vocabularies. We performed quantitative and qualitative analysis to measure the usefulness of the learned

representations. The results demonstrate that definition2vec is superior to the state-of-the-art algorithms.

In the second part of the thesis, we describe a new visual interface that helps healthcare researchers select patient cohorts from EHR data. Process of identifying patients of interest for observational studies from EHR data is known as cohort selection, a challenging research problem. We considered a problem of cohort selection from medical claim data, which requires identifying a set of medical codes for selection. However, there are tens of thousands of unique medical codes, and it becomes very difficult for any human to decide which codes identify patients of interest. To help users in defining a set of codes for cohort identification, we developed an interactive system, called Medical Claim Visualization system (MedCV), which visualizes medical code representations. MedCV analyzes a medical claim database and allows users to reason about medical code relationships and define inclusion rules for the selection by visualizing medical codes, claims, and patient timelines. Evaluation of our system through a user study indicates that MedCV enables domain experts to define inclusion rules efficiently and with high quality.

The third part of the thesis is a study of the definition of acute kidney injury (AKI), which is a condition where kidneys suddenly cannot filter waste from the blood. AKI is a major cause of patient death in intensive care units (ICU) and it is critical to detect it early. Recently published KDIGO medical guideline proposed a clinical definition of AKI using blood serum creatinine and urine output. The KDIGO definition was developed based on the expert knowledge, but very little is known about how well it matches the medical practice. In this study, we investigated publicly available EHR data from 47,499 ICU admissions to determine the concordance between the KDIGO definition and AKI determination by the medical provider. We show that it is possible to find a formula using machine learning with much higher concordance with the medical provider AKI coding than KDIGO and discuss the medical relevance of this finding.

# ACKNOWLEDGEMENTS

This thesis could not have been written without my academic advisor Dr. Slobodan Vucetic, who is always encouraging, supportive and helpful during my doctoral study. He not only taught me how to do good research, but also guided me to learn how to overcome difficulties and pursue my goal. I want to express my sincere gratitude and profound indebtedness to him for guiding me to become a researcher.

I would especially like to thank Dr. Brian Egleston from Fox Chase Cancer Center and Dr. Zivjena Vucetic from Clinical Genomics, from whom I learned how to collaborate with researchers from other disciplines. I would also thank Dr. Zoran Obradovic, and Dr. Eduard Dragut, who provided constructive feedback and challenging questions during my dissertation and thesis writing and urged me to improve my work.

I would like to express my gratitude to three great teachers for shaping my life: school teacher Mr. Tapan Kumar Biswas, BS supervisor Dr. Md Samiullah, and MS supervisor Dr. Chowdhury Farhan Ahmed. I would like to extend my thanks to Dr. Mahmudul Hasan Nayeem and Dr. Ishtiaque Hussain who guided me to apply for the PhD program.

I sincerely thank the staffs and faculties from the Department of Computer and Information Sciences at Temple University, for creating a world-class environment for me to work in. Special thanks to Dr. Karl Morris, Dr. Xiuqi Li, Dr. Richard Beigel, Dr. Andrew Rosen, for whom I had worked as a teaching assistant. Their experience and enthusiasm were of great inspiration throughout my years of teaching. I would like to

thanks Julie Krystopa Skrocki, Charles T. Rorke, Kalee Marshall, Andrea McGady, Marah Minetola, and Michelle Rambo who helped me with tons of paper work.

I would like to thank the physicians and staffs from Tuttleman Counseling Services at Temple University, for providing me counseling services during my PhD study.

I thank Dr. Vasil Hnatyshin from Rowan University, for giving me the opportunity to teach advance computer science topics during my doctoral study. I thank my internship mentors, Dr. Yoshihisa Shinagawa and Dr. Halid Ziya Yerebakan. Those were invaluable and unforgettable experience.

I thank my labmates and my friends, Dr. Shanshan Zhang, Dr. Tian Bai, Aniruddha Maiti, Ziyu Yang, Saman Enayati, Sandro Hauri, Elizabeth Garrison, Tamara Katic, Sai Shi, Hanzi Xu, Abbey Liu, Zhuoan Zhou, Kevin Esslinger, and so on. The encouragement and good memories from them keep me going through good times and bad times.

I am grateful to my family members who always encouraged me to complete my doctoral study. I would like to thank all my friends back home in Bangladesh, here in Philadelphia and across the United States of America for being with me throughout these years.

And finally, I would be remiss if I did not mention the two extraordinary women who have touched my life. First, my mother, Nomita Chanda - nurturer, reader, and role model. And my wife, Sumona Deb - singer, painter, gardener, and without any doubt the most amazingly supportive woman I have ever known.

# TABLE OF CONTENTS

<b>ABSTRACT</b>	<b>ii</b>
<b>ACKNOWLEDGEMENTS</b>	<b>iv</b>
<b>LIST OF TABLES</b>	<b>ix</b>
<b>LIST OF FIGURES</b>	<b>xi</b>
<b>1 INTRODUCTION</b>	<b>1</b>
1.1 Improving Medical Term Embeddings Using UMLS Metathesaurus . . . .	2
1.2 Developing an Interactive Visualization System for Patient Cohort Identification from Medical Claim Data . . . . .	3
1.3 Concordance between KDIGO Definition of Acute Kidney Injury and Its Coding in Clinical Practice . . . . .	3
<b>2 IMPROVING MEDICAL TERM EMBEDDINGS USING UMLS METATHESAURUS</b>	<b>5</b>
2.1 Introduction . . . . .	5
2.2 Related Work . . . . .	7
2.3 Methods . . . . .	8
2.3.1 Problem Definition . . . . .	9
2.3.2 Skip-gram Algorithm . . . . .	9
2.3.3 Our Proposed Method: Definition2vec . . . . .	10
2.4 Evaluation . . . . .	13
2.4.1 Data Sets . . . . .	13

2.4.2	Data Processing . . . . .	14
2.4.3	Learning Medical Term Embeddings . . . . .	15
2.4.4	Downstream Evaluation: Predicting ICD-9-CM Diagnosis Codes	17
2.4.5	Downstream Evaluation: Predicting ICD-9-CM Diagnosis Codes using Small Training Data . . . . .	19
2.4.6	Semantic Similarity Evaluation: 3 Human Labeled Data Sets . . .	20
2.4.7	Semantic Similarity Evaluation: UMLS Semantic Types . . . . .	22
2.4.8	Qualitative Evaluation . . . . .	23
2.4.9	Qualitative Evaluation: Out-Of-Vocabulary (OOV) Medical Terms	25
2.5	Discussion . . . . .	26
2.5.1	Limitations . . . . .	27
2.6	Conclusions . . . . .	27
<b>3</b>	<b>DEVELOPING AN INTERACTIVE VISUALIZATION SYSTEM FOR PATIENT COHORT IDENTIFICATION FROM MEDICAL CLAIM DATA</b>	<b>28</b>
3.1	Introduction . . . . .	28
3.2	Related Work . . . . .	31
3.2.1	Medical Event Analysis . . . . .	31
3.2.2	Visualization of Medical Data . . . . .	32
3.3	Model Task Abstraction . . . . .	33
3.4	Method Overview . . . . .	34
3.4.1	Skip-gram Method . . . . .	34
3.4.2	PMI Method . . . . .	36
3.5	Our Proposed System: MedCV . . . . .	37
3.5.1	Interface Overview . . . . .	37
3.5.2	Interaction and Workflow . . . . .	40
3.6	Evaluation . . . . .	41

3.6.1	Dataset . . . . .	41
3.6.2	Training and Implementation . . . . .	42
3.6.3	Baseline . . . . .	42
3.6.4	User study . . . . .	42
3.6.5	Comparing with Gold Standard Data . . . . .	45
3.6.6	Expert Interview and Feedback . . . . .	45
3.7	Conclusions . . . . .	48
<b>4</b>	<b>CONCORDANCE BETWEEN KDIGO DEFINITION OF ACUTE KIDNEY INJURY AND ITS CODING IN CLINICAL PRACTICE</b>	<b>49</b>
4.1	Introduction . . . . .	49
4.2	Methods . . . . .	54
4.2.1	Cohort Selection . . . . .	54
4.2.2	AKI Diagnosis Codes . . . . .	56
4.2.3	KDIGO Implementation . . . . .	56
4.2.4	Concordance Analysis . . . . .	57
4.2.5	Prediction of ICD AKI from SCr . . . . .	58
4.3	Results . . . . .	59
4.3.1	Concordance for Different KDIGO Implementations . . . . .	59
4.3.2	Analysis of TP, FP, FN, TP Admissions . . . . .	60
4.3.3	Prediction of Hospital-based AKI Diagnosis from SCr . . . . .	63
4.3.4	Does Concordance Differ Based on ICU Type? . . . . .	65
4.3.5	Does Concordance Change Over Time? . . . . .	65
4.4	Conclusions . . . . .	68
4.5	Additional Results . . . . .	69
	<b>BIBLIOGRAPHY</b>	<b>72</b>

# LIST OF TABLES

2.1	Statistics of discharge summaries in the MIMIC-III training data . . . . .	12
2.2	Accuracy of ICD-9-CM diagnosis code prediction using large training data set (predicting top 2,690 ICD-9-CM diagnosis codes having frequency $\geq 10$ in training data) . . . . .	19
2.3	Accuracy of ICD-9-CM diagnosis code prediction using small training data sets (UT: number of unique medical terms, DC: number of ICD-9-CM diagnosis codes, PDC: number of predicted ICD-9-CM diagnosis codes occurring at least 10 times in training data, SG: skip-gram, D2V: definition2vec) . . . . .	21
2.4	Pearson correlation coefficient for semantic pair similarity . . . . .	22
2.5	Cluster NMI value for different models . . . . .	23
2.6	Showing top 10 nearest neighbor terms for “ <i>heart attack</i> ” in <i>definition2vec</i> and skip-gram . . . . .	23
2.7	Showing top 10 nearest neighbor terms for “ <i>bipolar disorder</i> ” in <i>definition2vec</i> and skip-gram . . . . .	24
2.8	Showing top 10 nearest neighbor terms for two OOV terms, “ <i>nicotine replacement therapy</i> ” and “ <i>gastric pains</i> ” in <i>definition2vec</i> . . . . .	25
4.1	ICD code description for AKI . . . . .	55
4.2	Concordance between ICD AKI and different implementations of KDIGO. . . . .	59
4.3	Statistics of clinical variables and outcomes in different subgroups: average number of SCr measurements, average SOFA renal score, percent of admissions with administered vasopressor, average hospital length of stay, average ICU length of stay, and percent of in-hospital mortality. (Here, vaso. = vasopressor, mort. = mortality, dial. = dialysis) . . . . .	60

4.4	Results of different predictive models for hospital-based AKI diagnosis (cohort-1: 47,499) . . . . .	63
4.5	Results of different predictive models on ICU unit types (MICU, CVICU, MISICU) . . . . .	66
4.6	Results of different predictive models on ICU unit types (SICU, TSICU, CCU) . . . . .	67
4.7	Concordance between ICD AKI and KDIGO AKI (I1 implementation) over the years . . . . .	68
4.8	Results of DT in different year ranges of MIMIC-IV database . . . . .	68
4.9	Results of different predictive models for hospital-based AKI diagnosis (Here, cohort: 46,869, FT: features, MSCr = maxSCr) . . . . .	70
4.10	Statistic of clinical variables and outcomes in different ICU units . . . . .	70
4.11	ICD code description for renal transplant . . . . .	71

# LIST OF FIGURES

2.1	An overall architecture of the predictive model with clinical note as input.	6
2.2	The framework for the skip-gram algorithm. . . . .	9
2.3	Architecture of the proposed <i>definition2vec</i> algorithm. . . . .	11
2.4	Illustrating a process for extracting definitions of medical terms. . . . .	14
3.1	A sample medical claim data of a hospitalized patient . . . . .	29
3.2	Skip-gram model architecture is shown for a claim data [The claim code description is presented in a box where ICD-9 diagnosis and CPT codes are marked in black and blue colors, respectively.] . . . . .	35
3.3	A screenshot of MedCV with four views: (a) Query view: user writes a query, such as a patient treatment or a medical code, (b): Table view: shows a ranked list of related medical codes based on two different metrics, (c) Projection view: displays medical code relationships in a 2D view responsive to user interaction, (d): Selected code view: keeps a record of medical codes selected by the user. . . . .	37
3.4	Showing table panel with code filtering option for “965”. . . . .	39
3.5	Claim view and patient view: Claim view shows 50 example claims for recently selected code by a user from table view; Patient view presents timeline for the selected claim from claim view. Here, user selects claim #48 from claim view and patient view shows the patient timeline of the selected claim #48 and highlights in blue circle. The ICD-9 diagnosis and CPT codes are marked with prefix “d_” and “h_”, respectively. . . . .	40
3.6	The workflow of our proposed MedCV software . . . . .	41
3.7	Expert users feedback on survey questions . . . . .	48
4.1	Selecting population from MIMIC-IV dataset . . . . .	54

4.2	Showing confusion matrix for KDIGO and ICD AKI . . . . .	57
4.3	The distribution of first SCr and maxSCr value in different subgroups of Cohort-1 . . . . .	62
4.4	ROC curve for a single feature with (TPR, FPR) points for three decision tree and three stages of I1 KDIGO implementation. [ROC AUC score are 0.91 and 0.88 for LR with maxSCr during hospital and first day of ICU, respectively] . . . . .	64

# CHAPTER 1

## INTRODUCTION

Data mining is the process of discovering interesting knowledge or information from large amounts of data. Electronic health records (EHRs) are complex large datasets describing the health status and treatment of patients for each of their encounters with a health system. EHRs contain both structured (i.e. lab values, medical diagnosis codes) and unstructured data (i.e. medical note data). The primary purpose of EHRs is helping providers to better manage patient care and exchange health information among healthcare organizations. In addition to their primary purpose, EHRs have also been used to solving various medical research problems such as patient phenotyping (Halpern et al., 2016a; Bai et al., 2018), health risk prediction (Choi et al., 2016b,d), prediction of medical events (Choi et al., 2016a; Bai and Vucetic, 2019), medical code extraction (Mullenbach et al., 2018), and relation extraction between medications and adverse drug effects (Christopoulou et al., 2020). In this research, we address three important research issues in EHR data analysis: 1) improving vector representations of medical terms with the help of external knowledge such as medical vocabulary, 2) reducing human labor during the selection of patients for retrospective studies with an interactive visual tool, and 3) identifying the onset of complex medical conditions such as acute kidney injury.

## 1.1 Improving Medical Term Embeddings Using UMLS Metathesaurus

Medical notes entered by medical staff in the form of free text are a particularly insightful component of EHRs. There is a great interest in applying machine learning methods on medical notes in numerous medical informatics applications (Banerjee et al., 2017; Maldonado et al., 2017). Learning vector representations, or embeddings, of terms in the notes, is an important pre-processing step in such applications. However, learning good embeddings is challenging because medical notes are rich in specialized terminology, and the number of available EHRs in practical applications is often very small.

In this research (Chanda et al., 2022), we propose a novel algorithm to learn embeddings of medical terms from a limited set of medical notes. The algorithm, called *definition2vec*, exploits external information in the form of medical term definitions. It is an extension of a skip-gram algorithm (Mikolov et al., 2013) that incorporates textual definitions of medical terms provided by the Unified Medical Language System (UMLS) Metathesaurus (Bodenreider, 2004).

To evaluate the proposed approach, we used a publicly available Medical Information Mart for Intensive Care (MIMIC-III) EHR data set (Johnson et al., 2016). We performed quantitative and qualitative experiments to measure the usefulness of the learned embeddings. The experimental results show that *definition2vec* keeps the semantically similar medical terms together in the embedding vector space even when they are rare or unobserved in the corpus. We also demonstrate that learned vector embeddings are helpful in downstream medical informatics applications.

This research shows that medical term definitions can be helpful when learning embeddings of rare or previously unseen medical terms from a small corpus of specialized documents such as medical notes.

## 1.2 Developing an Interactive Visualization System for Patient Cohort Identification from Medical Claim Data

A medical claim is a summary of a patient's visit used for billing purposes containing a list of medical codes describing the patient's conditions and treatments provided during the visit. In addition to their primary billing purpose, medical claims have been used in observational studies in medical research to examine a range of questions pertaining to effectiveness and outcomes of medical treatments. A critical and time-consuming step in observational studies is cohort identification (Nattinger et al., 2004; Winkelmayr et al., 2005), which refers to identifying patients of interest for the study from the claims data (Bleicher et al., 2012; Miller et al., 2009). We introduce software, called *MedCV* (**M**edical **C**laim **V**isualization software), to assist practitioners during cohort identification. MedCV enables users to define rules for the selection of patients by exploring and visualizing medical codes, claims, and patient timelines. We evaluated our system through a user study on a large-scale claim dataset. The results demonstrate that MedCV enables practitioners to define inclusion rules very efficiently and with high quality.

## 1.3 Concordance between KDIGO Definition of Acute Kidney Injury and Its Coding in Clinical Practice

Acute kidney injury (AKI) represents a sudden decrease in kidney function that should be treated promptly to predict kidney failure. Currently, the accepted definition of AKI is based on consensus-based criteria developed by the Kidney Disease Improving Global Outcomes (KDIGO) initiative (Kellum et al., 2012). KDIGO guideline defines AKI based on sudden increase in serum creatinine (SCr) in blood and sudden drop in urine output (UO). Although the KDIGO criteria helped to define the AKI for epidemiological and clinical research, the adoption of KDIGO criteria for clinical management of patients with AKI and its utility in clinical practice is debated and not completely understood.

To gain an insight, we study concordance between KDIGO AKI definition and ICD coding of kidney injury by medical providers. Our retrospective study includes 47,499 ICU admissions between 2008 to 2019 covering 38,676 patients of Beth Israel Deaconess Medical Center (Johnson et al., 2020). High degree of concordance between KDIGO and ICD coding in MIMIC-IV admissions would indicate that KDIGO formula matched the clinical understanding of AKI/AKF in Beth hospital during the study period. Moreover, since KDIGO was proposed in 2012, it could be expected that the concordance was stronger in years after 2012. Our results show that the concordance is relatively low and that it is possible to train a machine learning algorithm to determine AKI from SCr measurements with a much higher concordance than the KDIGO definition.

The thesis is organized as follows. We propose an approach in Chapter 2 that learns medical term representations from EHR data by exploiting medical vocabularies. In Chapter 3, we describe an interactive software that helps practitioner explore medical code relationships for cohort identification. In Chapter 4, we study concordance between AKI definition provided by KDIGO guidelines and AKI coding by a large healthcare provides.

## CHAPTER 2

# IMPROVING MEDICAL TERM EMBEDDINGS USING UMLS METATHESAURUS

### 2.1 Introduction

Health providers use Electronic Health Records (EHRs) to keep information about their patient's medical conditions and the procedures employed to treat them. While the primary purpose of EHRs is operational and administrative, EHRs have been increasingly useful in biomedical research studies such as patient phenotyping (Halpern et al., 2016a; Bai et al., 2018), health risk prediction (Choi et al., 2016b,d), prediction of medical events (Choi et al., 2016a; Bai and Vucetic, 2019), medical code extraction (Mullenbach et al., 2018), and relation extraction between medications and adverse drug effects (Christopoulou et al., 2020). Particularly, valuable parts of EHRs are medical notes, which are free text created by the medical staff to provide insights about the condition and treatment of patients. Extracting information and analysis of medical notes is an open machine learning (ML) problem. A critical pre-processing step in modern approaches for medical note analysis is medical term *embedding*, which refers to the representation of medical terms as vectors. Medical term embeddings can be used as inputs for neural networks in a range of predictive and descriptive tasks (Banerjee et al., 2017; Maldonado et al., 2017).

An overall architecture of the predictive tasks are shown in Figure 2.1. In this paper, we refer to a medical term as a single word (e.g., Parkinson) or a multi-word (e.g., Parkinson’s disease) that is linked to an entry in a medical thesaurus, such as the UMLS Metathesaurus (Bodenreider, 2004).

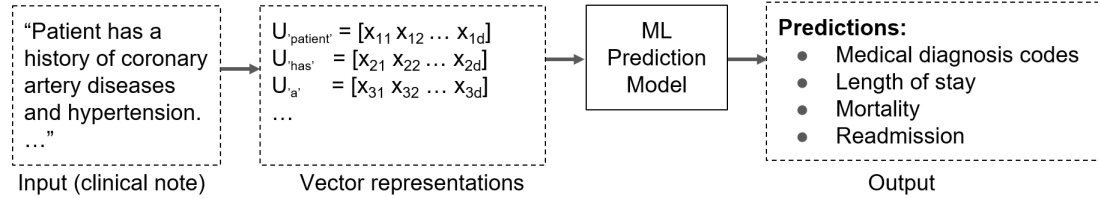


FIGURE 2.1: An overall architecture of the predictive model with clinical note as input.

Recent research has resulted in several methods for learning embeddings of medical terms, diagnosis and procedure codes, medications, and lab tests (De Vine et al., 2014; Choi et al., 2016e; Cai et al., 2018; Khattak et al., 2019). In particular, the skip-gram model (Mikolov et al., 2013) is a popular choice for learning embeddings of terms both from general-purpose corpora (e.g., Wikipedia) and from specialized corpora (e.g., medical notes) (Bai et al., 2018; Choi et al., 2016e,c) due to its simplicity and computational efficiency. The skip-gram and related embedding approaches, such as fastText (Bojanowski et al., 2016), work well when a document corpus is large and when terms that need to be embedded are frequent. However, there are many applications that rely on relatively small corpora with an abundance of specialized terms and abbreviations (Perotte et al., 2011; Coffman and Wharton, 2007; Crammer et al., 2007), where direct application of the skip-gram model does not always result in high-quality embeddings.

The main contribution of this study is summarized as follows: we propose a new algorithm, called *definition2vec* in this paper (Chanda et al., 2022), that is particularly appropriate for learning embeddings of infrequent or unobserved medical terms from a small corpus of medical notes. Our approach enhances the skip-gram algorithm by exploiting textual definitions of medical terms from existing publicly available resources,

such as the UMLS Metathesaurus. We demonstrate experimentally that our algorithm provides useful embeddings of infrequent and unobserved medical terms and that those embeddings can increase the quality of downstream medical informatics tasks.

## 2.2 Related Work

Learning embeddings of n-grams, words, terms, sentences, and paragraphs is an active research topic due to the importance of embeddings in deep learning approaches for natural language processing. Modern embedding algorithms draw inspiration from the well-known distributional hypothesis, which states that words that occur in the same contexts tend to purport similar meanings (Harris, 1954). An overview of traditional embedding approaches is provided in (Turney and Pantel, 2010). More recently, starting from seminal papers proposing skip-gram (Mikolov et al., 2013), GloVe (Pennington et al., 2014), and fastText (Bojanowski et al., 2016) algorithms, many general-purpose and specialized embedding algorithms were proposed both for processing text and various types of data objects such as sequences and graphs (Grover and Leskovec, 2016). The skip-gram algorithm (Mikolov et al., 2013) learns embeddings as a by-product of predicting context words of a target word. FastText (Bojanowski et al., 2016) is an alternative approach that treats words as sequences of n-grams that have their own embeddings and is sometimes useful in finding representations of out-of-vocabulary words.

Studying specialized approaches for embeddings of medical terms and concepts has been an active research area (Pakhomov et al., 2016; Bai et al., 2018; Wang et al., 2018; Kalyan and Sangeetha, 2020). The work on learning UMLS concept representations from medical notes and journals using the skip-gram algorithm (Choi et al., 2016e; De Vine et al., 2014) is particularly relevant to this paper. A recent study (Chiu et al., 2016) provides an extensive analysis of bio-medical word embeddings based on the skip-gram architecture. Med2Vec (Choi et al., 2016c) is another relevant work that uses a two-layer

neural network for learning embeddings of medical concepts from code occurrences and clinical narratives about patient visits. The authors of (Beam et al., 2018) proposed *cui2vec* that learns the embedding of UMLS Concept Unique Identifiers (CUIs) based on the distribution of concept co-occurrences in clinical notes. A related approach is described in (Cai et al., 2018) that focuses on temporal relations to embed medical concepts. It extends the Continuous Bag of Words (CBOW) model (Mikolov et al., 2013) to develop a time-aware attention approach for learning medical concepts. The research survey of Hahn et al. (Hahn and Oleynik, 2020) provides a detailed overview of different medical information extraction methods that rely on medical term embeddings.

Other studies used external knowledge sources in different ways to improve embeddings and downstream predictive models (Maldonado et al., 2019; Zhang et al., 2019). The authors in (Maldonado et al., 2019) combine UMLS Metathesaurus and Semantic Network information to learn concept embeddings following the Generative Adversarial Networks (GAN) framework (Goodfellow et al., 2014). Work in (Zhang et al., 2019) uses the Medical Subject Heading (MeSH) term graph (Lipscomb, 2000) to generate MeSH term sequences. While this previous work exploited known relations between medical terms, in our work, we leverage medical term definitions through an easy-to-implement and computationally efficient skip-gram extension.

## 2.3 Methods

In this section, we describe our proposed algorithm that learns the embeddings of medical terms. We first define the problem and briefly introduce the baseline skip-gram algorithm (Mikolov et al., 2013), which is the basis of our approach. Then, we describe our proposed algorithm.

### 2.3.1 Problem Definition

Let us suppose we are given a corpus of medical notes. We describe a single note  $N$  as an ordered sequence of terms,  $N = \{w_1, w_2, \dots, w_n\}$ , where  $w_i$  is a term from vocabulary  $V$  and  $n$  is the length of the note. The size of the vocabulary is  $|V|$ . A term can be a single word (e.g., Parkinson) or a multi-word (e.g., Parkinson’s disease). The objective of term embedding is to represent each term from the vocabulary as a vector, such that semantically similar terms have similar vectors.

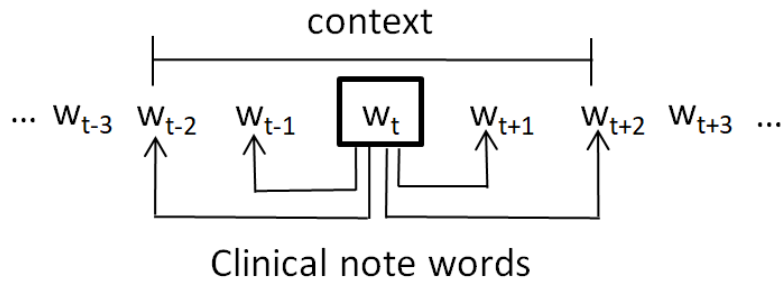


FIGURE 2.2: The framework for the skip-gram algorithm.

### 2.3.2 Skip-gram Algorithm

The skip-gram algorithm for embedding (Mikolov et al., 2013) scans the terms in a note and updates their vector representations based on their context. The context of a term is typically defined as its neighboring terms in a sequence. Given the target term  $w_t$  from the corpus, the skip-gram algorithm creates term pairs consisting of the scanned term  $w_t$  and its context terms  $w_i$ , and uses pairs  $(w_t, w_i)$  to update the likelihood of observing the context term  $w_i$  given the target term  $w_t$ . The context of  $w_t$  is defined as its neighboring terms  $C_{w_t} = (w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2})$ , if the context size is 2. Context terms  $w_i$  are selected from  $C_{w_t}$ . The log-likelihood of observing context terms for all the terms in the corpus is defined as

$$\mathcal{L} = \sum_{t, w_i \in C_{w_t}} \log p(w_i | w_t), \quad (2.1)$$

where  $p(w_i|w_t)$  is the conditional probability of context term  $w_i$  given the target term  $w_t$ . The skip-gram approach is illustrated in Figure 2.2.

In order to model  $p(w_i|w_t)$ , skip-gram assigns vectors  $U_w$  and  $V_w$  to term  $w$  from the vocabulary. The dimension of both vectors is the same. The conditional probability is defined as the following softmax function

$$p(w_i|w_t) = \frac{e^{U_{w_t} \cdot V_{w_i}}}{\sum_{w_j \in |V|} e^{U_{w_t} \cdot V_{w_j}}}, \quad (2.2)$$

where the dot product between two vectors is used to measure the similarity between two terms. A gradient descent algorithm can be used to maximize the objective function of equation (2.1). However, since the computational complexity of calculating equation (2.2) is very high due to its denominator, skip-gram uses negative sampling where the log-likelihood objective function is replaced with the negative sampling instantaneous loss for each target word  $w_t$ , defined as

$$E_t = \sum_{i \in C_{w_t}} (-\log \sigma(U_{w_t} \cdot V_{w_i})) - \sum_{w_j \in W_{neg}} -\log \sigma(U_{w_t} \cdot V_{w_j}), \quad (2.3)$$

where

$$\sigma(U_{w_t} \cdot V_{w_x}) = \frac{1}{1 + e^{-U_{w_t} \cdot V_{w_x}}}. \quad (2.4)$$

Here,  $W_{neg}$  is a set of  $K$  so-called negative terms randomly sampled from the corpus. Skip-gram uses a stochastic gradient algorithm to greedily maximize the instantaneous loss. After the training is finished, vector  $U_w$  is used as an embedding for term  $w$ .

### 2.3.3 Our Proposed Method: *Definition2vec*

The proposed *definition2vec* algorithm enhances the skip-gram approach by exploiting the textual definitions of medical terms available in public resources. Similar to skip-gram, it scans the terms in a corpus and uses stochastic gradient descent to minimize the

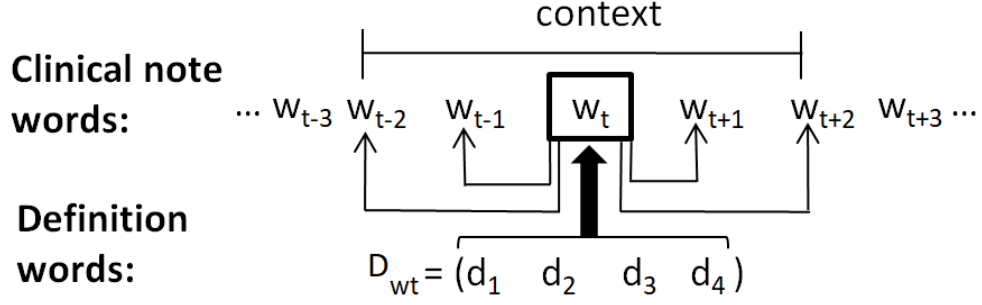


FIGURE 2.3: Architecture of the proposed *definition2vec* algorithm.

negative sampling instantaneous loss. However, when updating the embedding of a term, *definition2vec* also accounts for embeddings from its definition.

Let us assume target term  $w_t$  has its definition in a form of a word sequence  $D_{w_t} = (d_1, d_2, \dots, d_m)$ , where  $d_i$  is the  $i$ -th definition word of  $w_t$  and  $m$  is the length of the definition. We denote  $z_d$  as the vector representation of word  $d$  from the definition and  $U'_{w_t}$  as the definition-independent vector for the target term. We express the resulting target vector as

$$U_{w_t} = \frac{\text{sqr}t(f_{w_t})U'_{w_t} + \beta \frac{\sum_{d \in D_{w_t}} z_d}{|D_{w_t}|}}{\text{sqr}t(f_{w_t}) + \beta}. \quad (2.5)$$

Here,  $f_{w_t}$  is the frequency of  $w_t$  in the corpus and  $\beta$  is a hyperparameter. By using equation 2.5, our goal is to obtain the embedding of  $w_t$  that is influenced by its context and definition. Figure 2.3 illustrates the proposed approach. If a term frequently occurs in the corpus, its representation will be influenced more strongly by its contextual terms than its definition words. However, if a term is rare or unseen in the corpus, its representation will be heavily influenced by its definition words. Hyperparameter  $\beta$  determines the impact of a term's definition on its embeddings.

Our proposed algorithm scans the corpus term by term and constructs pairs of context and target terms together with their corresponding negative pairs. It follows the negative sampling idea of skip-gram and uses a stochastic gradient algorithm to minimize the instantaneous loss. The updates of context term, target term, and definition word vectors

are calculated as follows,

$$V_{w_x} = V_{w_x} - \alpha \frac{dE}{d(V_{w_x})} \quad (2.6)$$

$$\frac{dE}{d(V_{w_x})} = \frac{dE}{d(U_{w_t} V_{w_x})} \frac{d(U_{w_t} V_{w_x})}{d(V_{w_x})} \quad (2.7)$$

$$U'_{w_t} = U'_{w_t} - \alpha \frac{dE}{d(U'_{w_t})} \quad (2.8)$$

$$\frac{dE}{d(U'_{w_t})} = \sum_{w_x \in (w_i \cup W_{neg})} \frac{dE}{d(U_{w_t} V_{w_x})} \frac{d(U_{w_t} V_{w_x})}{d(U_{w_t})} \frac{d(U_{w_t})}{d(U'_{w_t})} \quad (2.9)$$

$$z_d = z_d - \alpha \frac{dE}{d(z_d)} \quad (2.10)$$

$$\frac{dE}{d(z_d)} = \sum_{w_x \in (w_i \cup W_{neg})} \frac{dE}{d(U_{w_t} V_{w_x})} \frac{d(U_{w_t} V_{w_x})}{d(U_{w_t})} \frac{d(U_{w_t})}{d(z_d)} \quad (2.11)$$

where  $\alpha$  is the learning rate.

After the training is finished, the target vector  $U_w$  becomes an embedding for term  $w$ . As a by-product of the learning procedure, we also learn the embeddings of each definition word.

Table 2.1: Statistics of discharge summaries in the MIMIC-III training data

# training notes	47,423
# of unique medical terms in training data	46,861
Average # of medical terms in a discharge summary	671
# of unique medical concepts in training data	29,740
Average # of medical concepts per discharge summary	364
Average # of definition words per medical concept	16
# of unique diagnosis codes in training data	6,717
Average # of diagnosis codes per discharge summary	11

## 2.4 Evaluation

In this section, we start by explaining the data sets and data preprocessing. Then, we describe the experimental design. Finally, we show and discuss the results of our qualitative and quantitative evaluation.

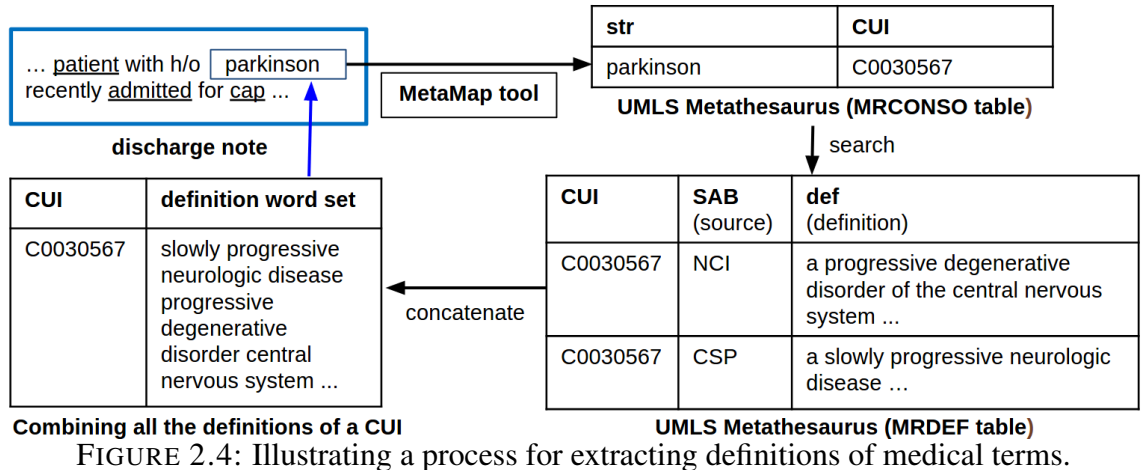
### 2.4.1 Data Sets

In our experiments, we used two data sets. The first data set is the UMLS Metathesaurus, which has textual definitions for a large number of medical terms. The second data set is MIMIC-III, which contains EHR records of a large number of Intensive Care Unit (ICU) patients with notes written in English.

**UMLS Metathesaurus:** Unified Medical Language System (UMLS) is a set of files and software that integrates multiple medical vocabularies (Bodenreider, 2004). UMLS Metathesaurus is the component of UMLS that maintains medical concepts and their textual definitions which are linked to different medical source vocabularies such as National Cancer Institute Thesaurus (NCIT) (Golbeck et al., 2003), Medical Subject Heading (MeSH) (Lipscomb, 2000), Universal Medical Device Nomenclature System (UMD) (Institute, 2018), Human Phenotype Ontology (HPO) (Robinson and Mundlos, 2010) and Mondo Disease Ontology (MONDO) (Mungall et al., 2017). UMLS Metathesaurus lists 188,050 concepts with at least one textual definition, each with its Concept Unique Identifier (CUI). Each concept has one or more medical terms associated with it, where each term has its String Unique Identifier (SUI). Each SUI can have one or more Atomic Unique Identifiers (AUI) that link the term to its definition from a particular source vocabulary. UMLS Metathesaurus has 773,692 SUIs. Although there are over 2.5 million medical concepts listed in UMLS Metathesaurus, in this study, we only consider those with at least one definition because *definition2vec* requires them.

**MIMIC-III:** Medical Information Mart for Intensive Care (MIMIC-III) is a publicly

available deidentified data set that contains EHRs of 41,127 ICU patients from Beth Israel Deaconess Medical Center recorded between 2001 to 2012 (Johnson et al., 2016). This data set contains both structured (medical codes, lab results) and unstructured (medical notes) data. MIMIC-III contains several types of medical notes such as progress notes, radiology reports, and discharge summaries. In this study, we only consider discharge summaries prepared by a health provider at the conclusion of an ICU stay. There is a total of 59,652 discharge summaries in MIMIC-III indicating that most patients have a single EHR in the data set. In our study, we are also interested in ICD-9-CM diagnosis codes (Organization, 2013) listed with each patient stay in the MIMIC-III data set. There is a total of 6,717 unique diagnosis codes listed in the data set.



### 2.4.2 Data Processing

Given a discharge summary, we performed several preprocessing steps illustrated in Figure 2.4. First, we removed digits and special characters, converted all characters into lower case, and tokenized the text. Then, we used MetaMap v16.2 (Aronson and Lang, 2010) to automatically match the tokens with UMLS CUIs. Each token can remain unmatched, become directly matched to a medical concept, or become a part of a multi-word phrase that is matched to a medical concept. If a matched concept is a multi-token such as

“*Parkinson disease*” we concatenated the tokens into a single token by adding an underscore special character such as “*Parkinson\_disease*”. Finally, we removed all unmatched tokens, such that each discharge summary becomes a sequence of tokens matched with medical concepts from UMLS Metathesaurus. This preprocessing procedure matches the previous work (De Vine et al., 2014).

To find definitions of each matched token, we performed the following steps. First, we identified the CUI of each matched token. Then, we found all AUIs corresponding to the CUI, retrieved the medical term definition of each AUI, and concatenated the definitions. Finally, we preprocessed the definition sentences to remove digits and special characters, lowercase all characters, tokenize, and remove stop words and rare words. Figure 2.4 illustrates the process that starts from a discharge note and ends with a sequence of CUI-matched tokens with their corresponding definitions.

### 2.4.3 Learning Medical Term Embeddings

After preprocessing the discharge summaries from MIMIC-III following the procedure illustrated in Figure 2.4, each medical term in the resulting corpus is linked to its definition sequence. In this subsection, we describe experimental design that was used to produce embeddings by *definition2vec* and the baseline algorithms.

Our first step was to split the set of preprocessed discharge summaries randomly into training, validation, and test sets. Similar to (Mullenbach et al., 2018), the resulting training data set contained 47,423 notes from 36,998 patients, test data had 3,372 notes from 2,755 patients, and validation set had 1,632 notes from 1,374 patients. One patient can have their discharge notes in only one of the three subsets.

We used the training data set to learn the embeddings of medical terms. In this way, we learned the embeddings of 46,861 medical terms corresponding to 29,740 medical concepts. Some statistics about the training data set are listed in Table 2.1. We trained *definition2vec* and the baselines on the preprocessed training data to learn medical term

embeddings. We used Python Gensim implementation of three popular embedding algorithms as baselines: GloVe<sup>1</sup>, skip-gram<sup>2</sup>, and fastText<sup>3</sup>.

We used the same hyperparameters for all embedding algorithms: word context neighborhood (or window size) = 5, embedding vector length (or feature size) = 100, learning rate = 0.01, number of negative samples = 5. Those same parameters had been used in previous research (Mikolov et al., 2013; Bai et al., 2017). All models were trained for 10 epochs, which was sufficient for convergence.

Glove, skip-gram, fastText, and definition2vec embeddings are non-contextualized, meaning that every term has a fixed vector representation. In contrast, recent research resulted in contextualized embeddings, where vector representation of a given term depends on the context in which it is mentioned. The most notable representative of contextualized embeddings is BERT neural network (Devlin et al., 2018), which was trained on a large corpus of general-purpose text. In particular, given an input text, the final hidden layer of BERT provides a 768-dimensional embedding for every WordPiece (Wu et al., 2016) token. Each word can be represented with the embedding of the first WordPiece token of the word. Such embedding is contextualized. A recent study (Si et al., 2019) found that the BERT contextualized embeddings can outperform context-free embeddings from skip-gram, fastText, and GloVe in several downstream tasks. Thus, in our experiments we compared BERT embeddings with the non-contextualized embeddings.

To get BERT contextual-embedding of medical terms in a discharge note, we sent the lowercased note to the BERT model and recorded the embedding of every medical term. If the discharge note has more than 512 tokens, we first divided it into subsequences shorter than 512 and concatenated medical term embeddings from all the subsequences.

---

<sup>1</sup> <https://radimrehurek.com/gensim/scripts/glove2word2vec.html>

<sup>2</sup> <https://radimrehurek.com/gensim/models/word2vec.html>

<sup>3</sup> <https://radimrehurek.com/gensim/models/fasttext.html>

We performed two types of studies to evaluate the baseline and our proposed term embeddings, as explained next.

#### *2.4.4 Downstream Evaluation: Predicting ICD-9-CM Diagnosis Codes*

Our first evaluation was to use the embeddings in a downstream task of predicting ICD-9-CM diagnostic codes for a given discharge summary. This is a multi-label classification where the prediction model provides multiple outputs, one for each ICD-9-CM diagnosis code. The employed prediction model was Convolutional Attention for Multi-Label classification (CAML) (Mullenbach et al., 2018), which is a convolutional neural network (CNN) with the attention mechanism. In CAML, each medical term from the pre-processed discharge summary is converted to a vector according to its embedding and provided as an input to the neural network. The output of CAML is a binary vector of predictions of ICD-9-CM diagnosis codes.

For measuring the accuracy, we use “recall at 8”, micro-averaged (MIC) and macro-averaged (MAC) F1, and area under the ROC curve (AUC), similar to the previous research (Mullenbach et al., 2018). Recall at  $k$  ( $k = 8$ ), is the fraction of correctly predicted ICD-9-CM diagnosis codes among the  $k$  most confidently predicted codes. To calculate F1, we must first calculate recall and precision. Recall is a fraction of true ICD-9-CM diagnosis codes predicted by CAML. Precision is a fraction of true ICD-9-CM diagnosis codes among the predicted codes. The F1 score is measured by the harmonic mean of recall and precision. In MIC calculations, each pair (discharge note, code) is taken as a separate prediction. Then, all predictions are used to calculate the F1 accuracy. On the other hand, the MAC values are computed by first calculating F1 on each individual ICD-9-CM diagnosis code. Then, the code-specific F1 accuracies are averaged to obtain the MAC F1 accuracy (Mullenbach et al., 2018). Compared to MIC F1 accuracy, the MAC F1 accuracy places a higher emphasis on rare code predictions.

We used the CAML implementation provided by the authors <sup>4</sup>. We used the learned embeddings of our proposed method and the three non-contextualized baselines as input for CAML. The embeddings were not modified during CAML training. All trained models had identical neural network architecture and the default hyperparameters given in the original paper. Each CAML was trained on all available training data. We checked the “recall at 8”, accuracy on the validation set after each epoch as stopping criteria. If the “recall at 8” value did not increase after ten consecutive epochs, we stopped the training. For *definition2vec*, we tuned  $\beta$  value by exploring different values i.e. 1, 2, 5, 10, 20, 50, and 100. Based on the validation data, we obtained the best results for  $\beta = 10$ . The CAML model had 2,690 outputs with sigmoid neurons, corresponding to all ICD-9-CM diagnosis codes with frequency  $\geq 10$  in our training data.

To evaluate the contextualized BERT embeddings, we also used the same CAML architecture and training procedures. The only difference was the dimensionality of the embeddings, which was 768 for BERT versus 100 for the non-contextualized embeddings.

The results in Table 2.2 show accuracy measured on test data. It can be observed that *definition2vec* is more accurate than the baselines on the F1 MAC measure, while it is comparable to skip-gram and fastText on other accuracy measures. We note that the F1 MAC accuracy gives a larger weight to rare ICD-9-CM diagnosis codes than the F1 MIC measure.

The results also show that BERT contextualized embeddings are not better than the non-contextualized *definition2vec* embeddings. We think that the main reason is that BERT was pre-trained on large general-purpose corpus while the *definition2vec* and the other baseline methods (i.e., GloVe, skip-gram, and fastText) were trained on a specialized discharge note corpus.

---

<sup>4</sup> <https://github.com/jamesmullenbach/caml-mimic>

Table 2.2: Accuracy of ICD-9-CM diagnosis code prediction using large training data set (predicting top 2,690 ICD-9-CM diagnosis codes having frequency  $\geq 10$  in training data)

Model	AUC		F1		R@8
	MIC	MAC	MIC	MAC	
BERT	0.9580	0.8769	0.4516	0.0932	0.3922
GloVe	0.9703	0.8888	0.4727	0.1126	0.3938
skip-gram	0.9790	0.9316	0.4995	0.1333	0.4147
fastText	<b>0.9794</b>	0.9340	0.4950	0.1372	0.4168
definition2vec	<b>0.9794</b>	<b>0.9350</b>	<b>0.5065</b>	<b>0.1489</b>	<b>0.4173</b>

#### 2.4.5 Downstream Evaluation: Predicting ICD-9-CM Diagnosis Codes using Small Training Data

In many medical informatics applications, the available corpus is much smaller than the MIMIC-III data set. Our hypothesis is that *definition2vec* is very appropriate for small data scenarios where most of medical terms are not observed often enough to enable baseline algorithms to learn good embeddings.

We repeated the CAML experiments described in the previous subsection using smaller training data sets. In particular, we created four training data sets by randomly sampling 1,000, 2,000, 5,000, and 10,000 discharge summaries from the training data. We trained *definition2vec* and the baselines (GloVe, skip-gram, fastText) on the small data sets for 40 iterations to learn concept embedding with the same parameters as before (window size = 5, feature size = 100, learning rate = 0.01, and number of negative samples = 5).

After learning the representations of medical terms, we trained a CAML model in the same manner, using the full training data set. We only predicted ICD-9-CM diagnosis codes that occurred at least 10 times in the training data set. For each size of training data, we used validation to determine the best choice for  $\beta$  in *definition2vec* from among the following choices;  $\beta = 1, 2, 5, 10, 20, 50,$  and  $100$ . We found  $\beta = 50$  gives the best results for 1,000 and 2,000 data sets,  $\beta = 20$  is the best choice for the 5,000 data set, and  $\beta = 10$  for the 10,000 data set.

Table 2.3 shows CAML accuracy for each data set. For all four small training data

sets, *definition2vec* outperforms the baselines on all metrics. The difference between *definition2vec* and the baseline methods is particularly large on the two smallest training data sets (1,000 and 2,000) and the difference reduces on the two largest training data sets (5,000 and 10,000). Therefore, Table 2.3 results strongly support our hypothesis that *definition2vec* is particularly useful on small corpora.

In addition, we found that larger  $\beta$  in *definition2vec* were appropriate for smaller training data sets and vice versa. This result supports our hypothesis that if a term is rare or unseen in the training corpus, its representation should be heavily influenced by its definition words.

#### 2.4.6 Semantic Similarity Evaluation: 3 Human Labeled Data Sets

Several studies (De Vine et al., 2014; Wang et al., 2018) used similarity scores between pairs of medical concepts or terms to evaluate learned embeddings. For the evaluation of our learned embeddings, we used three different data sets as described below.

**Pedersen data set:** Pedersen (Pedersen et al., 2007) provides a data set of 30 UMLS medical term pairs with semantic similarity judgments by 3 physicians and 9 clinical terminologists.

**Pakhomov data set:** This data set (Pakhomov et al., 2011) consists of 101 clinical term pairs whose similarity was determined by 9 medical coders and 3 physicians from Mayo Clinic.

**UMNSRS data set:** The UMNSRS data set (Pakhomov et al., 2010) has 566 medical term pairs. Each medical term pair has a semantic similarity score determined by 8 medical residents from the University of Minnesota Medical School.

For this experiment, we treated all strings in the three data sets as medical terms and we matched them with our embeddings. To compare the embeddings, we measured the cosine similarity between them and calculated the Pearson correlation coefficient between the cosine similarity scores and the scores by the human experts. Some medical terms

Table 2.3: Accuracy of ICD-9-CM diagnosis code prediction using small training data sets (UT: number of unique medical terms, DC: number of ICD-9-CM diagnosis codes, PDC: number of predicted ICD-9-CM diagnosis codes occurring at least 10 times in training data, SG: skip-gram, D2V: definition2vec)

<b>1,000 data set</b> UT: 9,632 DC: 1,351 PDC: 138					
<b>Model</b>	AUC		F1		
	MIC	MAC	MIC	MAC	R@8
GloVe	0.8240	0.6919	0.1546	0.0266	0.3560
BERT	0.8368	0.7212	0.1675	0.0341	0.3588
SG	0.8409	0.7426	0.1440	0.0320	0.3797
fastText	0.8414	0.7720	0.1968	0.0711	0.4001
definition2vec	<b>0.8587</b>	<b>0.7958</b>	<b>0.2583</b>	<b>0.0985</b>	<b>0.4323</b>
<b>2,000 data set</b> UT: 13,551 DC: 1,932 PDC: 272					
<b>Model</b>	AUC		F1		
	MIC	MAC	MIC	MAC	R@8
GloVe	0.8505	0.7512	0.2175	0.0500	0.3306
BERT	0.8636	0.7731	0.2022	0.0466	0.3431
skip-gram	0.8709	0.7873	0.2050	0.0312	0.3455
fastText	0.8722	0.7929	0.2059	0.0362	0.3539
definition2vec	<b>0.8891</b>	<b>0.8338</b>	<b>0.2915</b>	<b>0.1055</b>	<b>0.3985</b>
<b>5,000 data set</b> UT: 19,601 DC: 3114 PDC: 500					
<b>Model</b>	AUC		F1		
	MIC	MAC	MIC	MAC	R@8
GloVe	0.9122	0.8386	0.2829	0.0805	0.3997
BERT	0.9198	0.8389	0.3016	0.1013	0.4063
skip-gram	0.9439	0.9002	0.4274	0.2056	0.4621
fastText	0.9468	0.9053	0.4291	0.2081	0.4663
definition2vec	<b>0.9475</b>	<b>0.9066</b>	<b>0.4314</b>	<b>0.2108</b>	<b>0.4696</b>
<b>10,000 data set</b> UT: 26,738 DC: 4,186 PDC: 1100					
<b>Model</b>	AUC		F1		
	MIC	MAC	MIC	MAC	R@8
GloVe	0.9496	0.8761	0.4257	0.1355	0.4352
BERT	0.9427	0.8743	0.3680	0.0970	0.3827
SG	0.9604	0.9105	0.4539	0.1796	0.4445
fastText	<b>0.9613</b>	0.9128	0.4554	0.1847	0.4472
definition2vec	<b>0.9613</b>	<b>0.9136</b>	<b>0.4564</b>	<b>0.1875</b>	<b>0.4488</b>

Table 2.4: Pearson correlation coefficient for semantic pair similarity

Data set	GloVe	skip-gram	fastText	definition2vec
Pedersen	0.2963	0.4297	0.6256	<b>0.6468</b>
Pakhomov	0.1712	0.5310	0.5732	<b>0.5888</b>
UMNSRS	0.2182	0.6058	0.6188	<b>0.6392</b>

from the three data sets do not exist in the vocabulary of our learned embeddings. Thus, we used 25, 67, and 306 medical term pairs from the three semantic similarity data sets, respectively. Since BERT is a contextual embedding model that provides different vectors for the same term in different contexts, we did not include this model as a baseline for this experiment. Table 2.4 shows the Pearson correlation coefficients for *definition2vec* and baseline methods. The results indicate that *definition2vec* better reflects the underlying semantic relationships between the medical terms.

#### 2.4.7 Semantic Similarity Evaluation: UMLS Semantic Types

UMLS semantic network has 127 different semantic types such as “*drug*”, “*virus*”, “*disease*”, and “*procedure*”, which categorize medical concepts and reveal the relationships between them. We labeled each of the embedded medical terms into one of the 127 classes. Then, we applied a k-means clustering algorithm with  $k = 127$  on the embeddings learned from the full training data set. We used normalized mutual information (NMI) to evaluate the purity of the clusters with respect to their semantic network labels. A high NMI value indicates that the clusters are pure and contain a limited set of semantic types in each cluster.

Table 2.5 compares the NMI values obtained with four different embedding algorithms. Clusters obtained with GloVe embeddings have the lowest conformity with semantic labels. Clusters obtained with *definition2vec* embeddings show the largest conformity. The clusters obtained with fastText were similar to *definition2vec*’s, with slightly less conformity. The results indicate that *definition2vec* is successful in keeping similar medical terms close together in the learned vector space.

Table 2.5: Cluster NMI value for different models

Model	NMI value
GloVe	0.1339
skip-gram	0.2130
fastText	0.2834
definition2vec	<b>0.3054</b>

Table 2.6: Showing top 10 nearest neighbor terms for “*heart attack*” in *definition2vec* and skip-gram

Large data set		Small data set	
definition2vec	skip-gram	definition2vec	skip-gram
blockage	blocked artery	myocardial infarctions	pain
heart muscle	blockage	acute mi	cough blood
heart attacks	heart blockage	infarction	scheduling
heart blockage	heart muscle	hemorrhagic stroke	aortic aneurysms
blocked heart	blocked heart	myocarditis	abuse substance
heart block diagnosis	heart muscles	hypertensive crisis	providers
block heart	blood clots lung	myocardial	skip
slow heart rate	heart function	restrictive	caregiver
slow heart rate	heart function	cardiomyopathy	caregiver
heart function	slow heart rate	ischemic change	substance abuse problem
myocardia	myocardial infarction mi	ischemia	cell phone

#### 2.4.8 Qualitative Evaluation

We learned *definition2vec* and baseline embeddings on the full training data set (47,423 summaries) and on the smallest training data set (1,000 summaries). Then, we searched the nearest neighbors in the embedding space for a range of medical terms. For a given medical term, we found its 10 nearest neighbors based on the cosine similarity between the embeddings. For example, Table 2.6 shows the nearest neighbor terms of “*heart attack*” based on learning from the full and the smallest training data sets. For the full data set, both *definition2vec* and skip-gram provide similar results, with “*blockage*”, “*heart muscle*”, “*heartblockage*”, and “*slow heart rate*” in the results of both methods. However, the results based on the smallest training data set are different. *definition2vec* finds

Table 2.7: Showing top 10 nearest neighbor terms for “*bipolar disorder*” in *definition2vec* and skip-gram

Large data set		Small data set	
definition2vec	skip-gram	definition2vec	skip-gram
schizophrenia	schizophrenia	depression	armour
schizoaffective disorder	schizoaffective disorder	psychosis	parkinson disease
major depression	depression	asthma	sildenafil
paranoid schizophrenia	major depression	hyperlipidemia	addison disease
bpad	bpad	neuropathy	ckd
psychotic disorder	multiple personality disorder	diabetic neuropathy	amenorrhea
bipolar affective disorder	seizure disorder	dyslipidemia	renal carcinoma
mood disorder	mood disorder	hypertension	obesity hypoventilation syndrome
bipolar illness	pervasive developmental disorder	malignant hypertension	oa
bipolar mood disorder	paranoid schizophrenia	anxiety	esophageal dilatation

“*myocardial infarctions*”, “*acute mi*”, “*hemorrhagic stroke*”, and “*hypertensive crisis*”, which are all the concepts related to “*heart attack*”. On the other hand, skip-gram finds “*pain*”, “*cough*”, “*blood*”, “*scheduling*”, “*skip*”, and “*cell phone*”, which are not as closely related to “*heart attack*”.

Table 2.7 shows another example with the nearest neighbors of “*bipolar disorder*”. Similar to the previous example, *definition2vec* and baseline embeddings result in similar neighborhoods when trained on the full training data set. For example, the top neighbors for both methods are “*schizophrenia*”, “*schizoaffective disorder*”, “*bpad*”, and “*mood disorder*”. However, the results obtained by learning on the smallest training data set are different. *definition2vec* finds several concepts that are related to the “*bipolar disorder*”, such as “*depression*”, “*psychosis*” and “*hyperlipidemia*”, while the nearest neighbors found by skip-gram are less related, such as “*armour*”, “*parkinson disease*”, and “*ckd*”

Table 2.8: Showing top 10 nearest neighbor terms for two OOV terms, “*nicotine replacement therapy*” and “*gastric pains*” in *definition2vec*

<b>nicotine replacement therapy</b>	<b>gastric pains</b>
nicotine replacement	stomach ache
smoking cessation therapy	stomach pain
nicotine patches	feeling bloated
nicotine transdermal patch	pain esophagus
cessation smoking	gastrointestinal pain
nicotine dependence	esophageal pain
nicotine addiction	abdominal pains
quitting smoking	low ache
nicotine lozenges	low pains
dependence nicotine	gi pain

(abbreviation of “*chronic kidney disease*”). From these results, we can conclude that *definition2vec* provides similar embeddings to skip-gram when both are trained on the full training data set, while it seems to be superior when the training data set is small.

#### 2.4.9 Qualitative Evaluation: Out-Of-Vocabulary (OOV) Medical Terms

There might be many important medical terms that do not occur in the training data, but have definitions in UMLS Metathesaurus. Since *definition2vec* learns word embeddings through medical term definitions, it can calculate the embeddings of OOV terms by taking the average of their definition word embeddings. For example, in Table 2.8 we show the top 10 neighbors of “*nicotine replacement therapy*” and “*gastric pains*” which do not occur in the full training data set. *definition2vec* properly finds “*nicotine replacement*”, “*smoking cessation therapy*”, “*nicotine patches*” among the nearest neighbors of the OOV “*nicotine replacement therapy*” term. Similarly, it properly identifies neighbors of the OOV term “*gastric pains*”. These results show that *definition2vec* can find the proper embeddings of OOV medical terms using definition word embeddings. This puts *definition2vec* at an advantage over Glove and skip-gram, which cannot provide embeddings for OOV terms. It also has an advantage over fastText, which relies purely on n-gram embeddings to calculate

the embeddings of OOV terms.

## 2.5 Discussion

Often in practice, a document corpus is too small for training language models and is only useful for learning embeddings of the most common terms. To address this issue, we extended the skip-gram algorithm to incorporate the definitions of medical terms from external publicly available resources. In our case, we relied on the UMLS Metathesaurus as the external source. We note that the proposed *definition2vec* algorithm allows other sources of medical term definitions, including web resources such as Wikipedia.

Our experiments show that *definition2vec* results in better medical term embeddings, especially when the size of a document corpus is small. This could be particularly useful in applications (Perotte et al., 2011; Coffman and Wharton, 2007; Crammer et al., 2007) where it is not feasible to have a large corpus, such as when the corpus is from a specialized medical practice, is related to the treatment of a rare medical condition, or is written in a rare language. *Definition2vec* could also be applicable to non-medical domains such as the embeddings of legal terms or specialized terms used in various scientific domains.

Recent advances in contextualized embedding represented by neural networks such as ELMo (Embedding from Language Models) (Peters et al., 1802) and BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2018) allow embeddings to depend on the context of each term’s occurrence. Although recent studies (Si et al., 2019; Chanda, 2021; Deb and Chanda, 2022) found that the BERT contextualized embeddings can be superior to context-free embeddings from skip-gram, fastText, and GloVe in some applications, our results indicate that in a small and specialized corpus setting it does not have to be the case. Another recent paper (Ji et al., 2021) also reported that BERT embeddings did not improve prediction accuracy on a medical code prediction task. We believe that this is because BERT is trained on general-purpose corpus that does

not provide sufficient information to capture useful representations of highly specialized medical terms.

### 2.5.1 Limitations

The proposed study has some limitations. For example, there are versions of BERT specialized for medical text, such as ClinicalBERT (Alsentzer et al., 2019), which was fine-tuned on all MIMIC-III medical notes. However, ClinicalBERT was not appropriate for our experiments, because we wanted to compare embeddings that could be learned on very small subsets of MIMIC-III. Thus, we had to constrain our evaluation to BERT contextualized embeddings.

Moreover, the presented experiments relied on MetaMap to match the text with medical concepts. MetaMap does not provide perfect coverage of medical terms, most often due to spelling mistakes or non-standard jargon or abbreviations. To enable matching of non-standard term variants, it might be helpful to consider character-level embedding neural networks trained to reconstruct, or mimic, an embedding from a word-level embedding model (Ha et al., 2020).

## 2.6 Conclusions

In this paper (Chanda et al., 2022), we proposed a new algorithm, *definition2vec*, which learns medical term embeddings by combining a data set of discharge summaries and definitions of medical terms. We evaluated the learned embeddings by comparing their usefulness when predicting medical codes from discharge summaries and how closely they match semantic similarities between medical terms. Our results indicate that *definition2vec* is particularly useful in downstream task when the training data set is small. Moreover, the medical term definitions are especially beneficial for the embedding of rarely seen or out-of-vocabulary medical terms. Hence, the proposed method can be useful for analysis of rare medical conditions and treatments from EHR data.

## CHAPTER 3

# DEVELOPING AN INTERACTIVE VISUALIZATION SYSTEM FOR PATIENT COHORT IDENTIFICATION FROM MEDICAL CLAIM DATA

### 3.1 Introduction

Health providers generate medical claims that list patient diagnosis and treatment for each patient visit for billing purposes (Figure 3.1). A diagnosis or a treatment is typically represented with one or more medical codes defined by a medical ontology. There are several standard medical code ontologies that are used for medical claims, such as the International Classification of Diseases (ICD) and the Current Procedural Terminologies (CPT) ontologies. Each medical code has its unique ID and a short description (e.g., ICD-9-CM code 8521 is described as “Local excision of lesion of breast”). Given the wide range of human diseases and the ways in which they are treated, the number of unique medical codes is large. For example, there are over 70,000 codes in ICD-10-PCS (the 10th ICD revision) ontology of medical procedures and slightly below 70,000 codes in ICD-10-CM ontology of medical diagnoses. Thus, it is almost impossible for any human to comprehend a list of code IDs in a medical claim, and even medical practitioners find it challenging to understand what happened during a visit simply by reading the definitions

of medical codes listed in a claim for that visit.

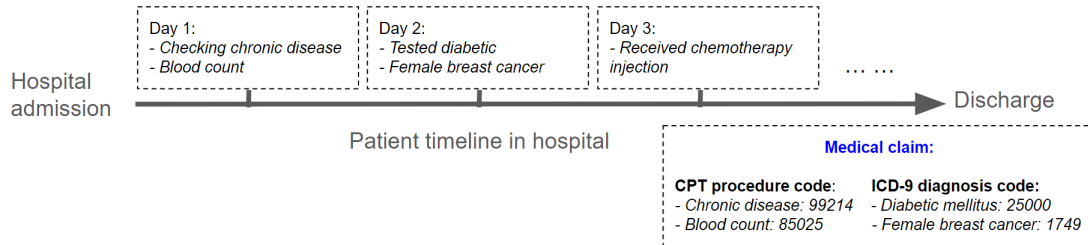


FIGURE 3.1: A sample medical claim data of a hospitalized patient

In addition to billing purposes, the medical claim data have been commonly used in retrospective studies in medical research to examine a range of questions pertaining to effectiveness and outcomes of medical treatments, such as in patient phenotyping (Warren et al., 2002; Halpern et al., 2016b) and in risk and mortality prediction (Yan et al., 2005; Klabunde et al., 2000; Krumholz et al., 2006; Warren et al., 2016). A critical and time-consuming step in retrospective studies is cohort identification (Nattinger et al., 2004; Winkelmayr et al., 2005), which refers to the process of identifying all patients with a particular property, such as patients receiving a particular type of therapy (Bleicher et al., 2012; Miller et al., 2009). Cohort identification requires a researcher to specify an inclusion criterion in a programmatic way that scans the medical claims and accurately identifies the patients. This task typically boils down to defining a set of codes that correspond to a particular diagnosis or a procedure. Cohort identification from large-scale medical claim databases which might consist of hundreds of thousands of patients and tens of millions of claims is challenging because the number of unique medical codes is extremely large (tens of thousands) and it is therefore very difficult for any human to define a query that identifies patients of interest with good coverage (not missing patients of interest) and precision (not including unrelated patients).

An illustration of difficulties that medical researchers face during cohort identification is the identification of chemotherapy treatment of cancer patients. Based on inclusion rules published in (Bleicher et al., 2012), there are 7 distinct CPT codes describing

“Ductography/galactography” including 19030 (“Injection procedure only for mammary ductogram or galactogram”), 76086 (Mammary ductogram or galactogram, singleduct, radiological supervision and interpretation), and 77054 (“Mammary ductogram or galactogram, multiple ducts, radiological supervision, and interpretation”), but not including other codes with similar ID such as 76080 (“Radiologic examination, abscess, fistula or sinus tract study, radiological supervision, and interpretation”). The anecdotal evidence based on our discussion with multiple research groups involved in retrospective studies with claims data indicates that cohort identification is the most time-consuming step that can last for weeks. It typically consists of keyword searches and manual scanning of medical ontologies coupled with scanning and manual examination of individual claims from a database of claims using custom-made and ad-hoc database searches. Those efforts often require teams consisting of physicians who can interpret the claims and programmers who can assist physicians in searching a database and writing ad-hoc code.

The contribution of our work is visualization software that helps medical researchers and clinicians define inclusion criteria that identify patients receiving a specific type of a medical treatment from medical claims data. In particular, we provide a visual interface that is able to show medical codes that are semantically similar to a query medical code. For example, if a user enters a single code for chemotherapy, our software is able to show other codes that either often co-occur with the query code in a claim, indicating synergistic procedures, or that are mutually exclusive but occur in similar types of claims, indicating alternative procedures for patient treatment. To obtain information about the relationship between pairs of codes our software relies on a popular machine learning algorithm for data embedding called word2vec (Mikolov et al., 2013). Our software is designed to make it easy for non-technical expert users to explore and select medical codes for inclusion by benefiting from tabular and scatter plot representations of codes, claims, and patient timelines.

We make three major contributions in this study. First, we propose two different

metrics to describe medical code similarity. Second, we derive a design space with a task-driven approach for medical domain experts to analyze medical code for cohort identification. Finally, we evaluated our software with help from expert users to quantify how helpful our interface is during cohort identification.

## 3.2 Related Work

### 3.2.1 *Medical Event Analysis*

In medical and statistical research based on the electronic health records and claims data, medical events are considered as a sequence of events and visualized and analyzed using sequence mining approaches to analyze patient health status (Wang et al., 2011; Tao et al., 2012; Kwon et al., 2016). Peekquence (Kwon et al., 2016) is a visual tool that mines medical event sequences using a popular sequential pattern mining approach, SPAM (Ayres et al., 2002) and presents patient event sequence aligned with respect to the mined pattern. The authors of (Tao et al., 2012) summarize patient timeline through a visual interface. To understand how the variations in sequences of events can impact medical outcomes, an exploratory visual system is discussed in (Gotz et al., 2014) for clinical episode analysis.

Medical claim data relies on tens of thousands of medical codes, where many codes are used rarely but have significant meaning in describing patient health status and treatment. For this reason, learning vector representations of medical codes became popular in healthcare research studies (Choi et al., 2016b; Nguyen et al., 2018; Choi et al., 2016e; Bai et al., 2018, 2019). For example, the authors of (Choi et al., 2016b) proposed CSM method that learns ICD code vectors from health records using skip-gram model (Mikolov et al., 2013) and authors of (Nguyen et al., 2018) used medical code representations to learn vector representation of a patient. In a recent work (Bai et al., 2019), the authors modified skip-gram model (Mikolov et al., 2013) to learn different types of medical codes

(i.e. ICD, CPT) jointly. However, this previous research (Choi et al., 2016b; Nguyen et al., 2018; Choi et al., 2016e; Bai et al., 2018, 2019) on vector representations of medical codes did not support human in the loop exploration.

### 3.2.2 *Visualization of Medical Data*

Recent research contributed several approaches that aid understanding of patient health status from health records by visualizing event series in timeline (Zhang et al., 2018; Gotz et al., 2019; Krause et al., 2015b). Type 1 diabetes treatment was analyzed in (Zhang et al., 2018), where different temporal event sequences are visualized as time series. Temporal event visualization of patient health data was also proposed in (Gotz et al., 2019) for interactive exploration.

Other research groups proposed visual tools to analyze event sequences of multiple patients to understand medical treatment (Wongsuphasawat et al., 2011; Monroe et al., 2013; Krause et al., 2015a; Rogers et al., 2019). For example, LifeFlow (Wongsuphasawat et al., 2011) provides an aggregated view on EHRs to detect patient treatment. EventFlow (Monroe et al., 2013) is another tool for simplifying patient records and allowing alignments on arbitrary points in time. Careflow (Perer and Gotz, 2013) aggregates patient event sequences by common event occurrences to find frequently observed progression patterns. Medical event embedding is used in Guo et al. (Guo et al., 2018) for visual progression analysis by dividing the hospital visit into seven stages. Another research work (Krause et al., 2015a) discussed patient cohort identification tool that used rule based query to specify temporal constraints. A user needs domain knowledge to specify such constraints to identify a cohort. Composer is a visual tool (Rogers et al., 2019) developed to help orthopedic surgeons dynamically define treatment patterns using clinical parameters (PROMIS score (Cella et al., 2010)) in patient medical histories to analyze patient-reported outcomes. However, unlike this paper, previous research focused on aggregating multiple patient event sequences to visualize commonalities, but did not

explore inclusion criteria for cohort identification.

### 3.3 Model Task Abstraction

The research goal of our design study is finding a group of patients who have undergone the same type of treatment. For this reason, we need to discover a set of medical codes that are used for a specific kind of treatment. From a large-scale dataset, the task of searching related medical codes for a treatment is challenging.

To define the similarity of medical codes we rely on word2vec approach (Mikolov et al., 2013) that represents codes as vectors such that codes that occur in similar types of medical claims have similar representation (Choi et al., 2016e; Bai et al., 2018, 2019). We also define similarity using the Pointwise Mutual Information (PMI) (Turney and Pantel, 2010) that defines codes as similar if they often co-occur in claims. There is a mathematical relationship between PMI and word2vec similarity measures (Pennington et al., 2014). Importantly, word2vec is different from PMI because it defines as similar both the codes that co-occur and codes that are mutually exclusive. Since both co-occurrence codes and context codes are important to know for patient treatment identification, we use both metric in our system.

To design a visual system that can help users analyze the relationships between different medical codes and select appropriate codes identifying the treatment, we consulted physicians at a partner cancer center during a period of over 2 years and followed an iterative software process model (Munzner, 2009) that required meeting domain experts to get their feedback about problem definition and design task analysis. The design tasks are listed below:

- **T1: Allow custom queries on claim data.** Real world medical claim data uses thousands of unique medical codes. To focus on a specific disease or procedure type, the software should support users to set their customized query and discover

data about a specific application.

- **T2: Interactively and visually explore related codes.** To provide an easy and quick overview about code similarity, the software should show a list of related codes in visual space.
- **T3: Allow users to filter out results by several criteria.** The task (T2) would display different types of metrics to evaluate medical code similarity. However, a user might only be interested in some specific criteria. The software should provide different filtering options to the user.
- **T4: Visualize the relationship between medical claims.** Users should be interested to know about a claim and its recorded medical codes. The software should also show claim data in a projected vector space.
- **T5: Visually explore patient timelines.** To have a detailed view of a patient, the software should provide a patient timeline to visualize the history of a patient to users.
- **T6: Identify medical codes for inclusion criteria for cohort selection** Finally, the users will be interested to select a code for inclusion based on the analysis and expert knowledge. The software should allow user to record his final selected codes. It should also provide an option to show patient cohort size for the selected codes.

## 3.4 Method Overview

### 3.4.1 Skip-gram Method

Word2vec Mikolov et al. (2013) has been used in medical informatics to represent medical words and codes as low dimensional vectors, or embeddings Choi et al. (2016b); Nguyen et al. (2018); Choi et al. (2016e); Bai et al. (2018, 2019); Chanda et al. (2022). In our

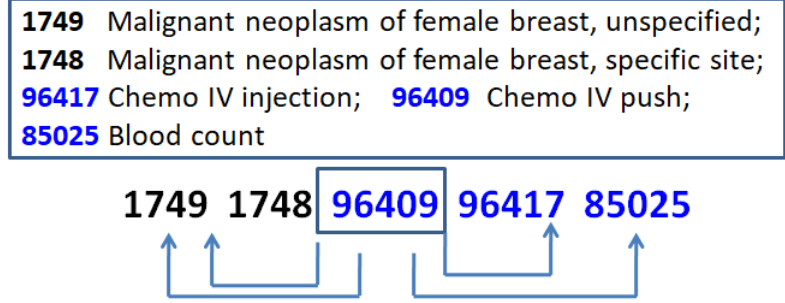


FIGURE 3.2: Skip-gram model architecture is shown for a claim data [The claim code description is presented in a box where ICD-9 diagnosis and CPT codes are marked in black and blue colors, respectively.]

application, Word2vec model (skip-gram variant) treats every claim as a document and every code in a claim as a word. Let us suppose we have a document of claims where a claim is a sequence of codes,  $N = \{... c_{t-2}, c_{t-1}, c_t, c_{t+1}, c_{t+2}, ...\}$ . Then, the skip-gram model scans the code sequence and makes pairs consisting of the scanned code and its context code and learns the likelihood of observing the context code  $c_i$  for the scanned code  $c_t$ , where context  $C_{c_t} = (c_{t-2}, c_{t-1}, c_{t+1}, c_{t+2})$ , with context size 2 and  $c_i \in C_{c_t}$ . The log-likelihood of observing  $c_i$  for  $c_t$  is defined as

$$\mathcal{L} = \sum_{c_i \in C_{c_t}} \log p(c_i | c_t), \quad (3.1)$$

where  $P(c_i | c_t)$  is the conditional probability of observing context code  $c_i$  for the given scanned word  $c_t$ . The model is explained in Figure 3.2 with an example claim.

If  $U$  is a matrix of the scanned code embeddings and  $V$  is the matrix of the context code embeddings, where  $U, V \in \mathbb{R}^{|V_{word}| \times f}$  and  $f$  is the embedding dimension, then the conditional probability is defined using softmax function as

$$p(c_i | c_t) = \frac{e^{U_{c_t} \cdot V_{c_i}}}{\sum_{c_j \in |V_{code}|} e^{U_{c_t} \cdot V_{c_j}}} \quad (3.2)$$

where  $U_{c_t}$  and  $V_{c_i}$  are vectors of codes  $c_t$  and  $c_i$ , respectively. A stochastic gradient algorithm is used to maximize the objective function to learn the scanned and context

vector representations for all the codes in the dataset (Mikolov et al., 2013). The cosine distance of the learned vector representation is used as a similarity score between two codes. The high cosine distance value means two codes are very related and they share similar context.

### 3.4.2 PMI Method

Pointwise mutual information (PMI) (Turney and Pantel, 2010) measures the likelihood that two codes co-occur in the same claim and compares it with how likely are they to co-occur if they were independent. Let us suppose,  $c_i$  and  $c_j$  are two codes in our medical claim dataset. Then, the PMI between the codes is defined as

$$PMI(c_i, c_j) = \log \frac{P(c_i|c_j)}{P(c_i)}, \quad (3.3)$$

where  $P(c_i | c_j)$  is the conditional probability of observing code  $c_i$  given code  $c_j$  and  $P(c_i)$  is the marginal probability of seeing  $c_i$  in the whole data. The probabilities are calculated based on the count of co-occurrence of the codes in dataset. Let us suppose the counts of code  $c_i$  and  $c_j$  in dataset are  $n_i$  and  $n_j$ , respectively. The code pair co-occurs in the claims  $n_{ij}$  times and the total number of times the two codes co-occur is  $n$ . Then, the probabilities can be calculated as

$$PMI(c_i, c_j) = \log \frac{P(c_i|c_j)}{P(c_i)} = \log \frac{n_{ij}/n_j}{n_i/n} = \log \frac{n_{ij} \cdot n}{n_i \cdot n_j} \quad (3.4)$$

The PMI value is useful to understand code relationships. The high PMI value means that two codes occur together in a same claim more frequently than we would expect if they are independent.

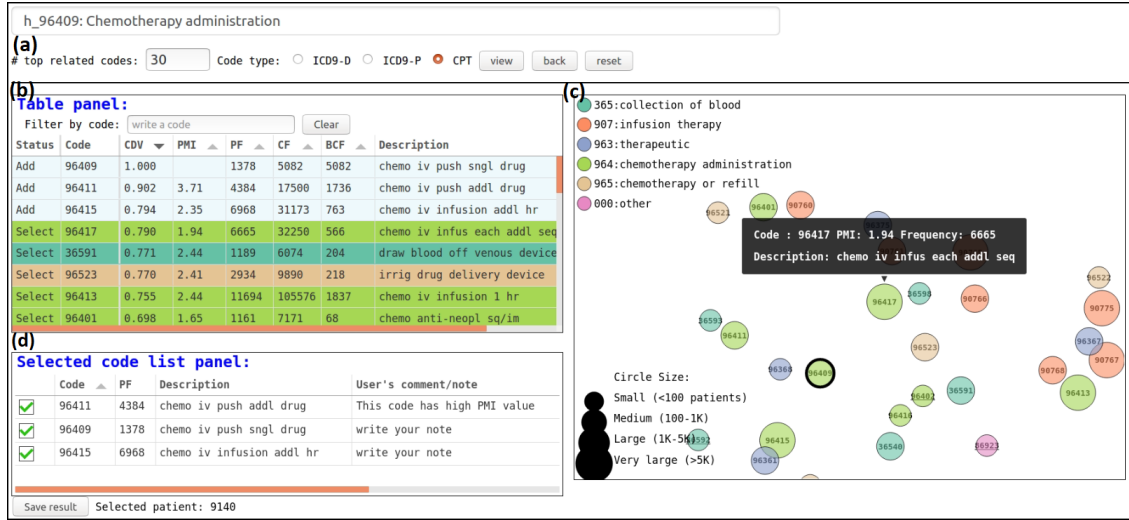


FIGURE 3.3: A screenshot of MedCV with four views: (a) Query view: user writes a query, such as a patient treatment or a medical code, (b): Table view: shows a ranked list of related medical codes based on two different metrics, (c) Projection view: displays medical code relationships in a 2D view responsive to user interaction, (d): Selected code view: keeps a record of medical codes selected by the user.

### 3.5 Our Proposed System: MedCV

MedCV is an open source browser-based interactive visualization software for physicians, medical researchers, and code interpreters that is built with JavaScript and D3.js library (Bostock et al., 2011). The main interface of our visual system, MedCV, is shown in Figure 3.3 that incorporates A) Query view, B) Table view, C) Projection view, and D) Selected code view. We also have two other views, E) Claim view and F) Patient view that opens in a separate window as shown in Figure 3.5. For our software design, we follow the popular graphical user interface design guideline “Overview first, zoom and filter, then detail-on-demand” (Shneiderman, 1996). A user will see the first four views at the beginning and the next two will open based on the user’s demand.

#### 3.5.1 Interface Overview

**Query view:** To take user’s input query and allow to set different parameters (design task T1), our system provides the query view, Figure 3.3(a). As a query, a user can type

a treatment or diagnosis name or a medical code. As discussed before, medical claims contain both ICD-9 and CPT codes and users have an option to select which types of code they are interested in exploring. Based on the user's query, our system provides a set of codes in both projection view and table list view that are related to the query code based on the cosine similarity distance.

**Projection view:** This view visualizes relationships between codes (**T2**). From the learned word2vec representations of medical codes (Mikolov et al., 2013), our system finds the top nearest neighbors of the given query code and provides a 2-dimensional t-SNE (Maaten and Hinton, 2008) projection of code vectors that allows users to explore the code relationships. The projection view is shown in Figure 3.3(c). Different colors are used to group codes based on ICD-9 or CPT code hierarchy. When the mouse hovers over a circle, concise information with code description, frequency, and PMI value is shown for the corresponding code. This interface also allows user to filter codes by clicking on the legend icon (**T3**). The underlined code in the projection view draws attention to neighboring codes with low PMI values ( $\leq 1$ ) and high word2vec similarity. The underlined codes are interesting because low PMI indicates that two codes occur in the same types of claims but do not commonly occur in the same claims.

**Table view:** The table view is an alternative way to inform users about the word2vec distance, PMI frequency values, and description of each code (**T2**). There are three types of frequency in the table.  $PF$  is the number of times a code occurs in patient.  $CF$  is the number of times a code occurs in the claim dataset, while  $BCF$  is the number of times the query code and the code in table row occur in the same claim. The table rows have exactly the same color as the color of code circled in the projection view. Moreover, the table view comes with sorting and filter by code options that further help the user with the design task (**T3**). Figure 3.4 presents a filter option of the table view.

The table view has a column, "select", that provides three options to a user, such as add, do not add, and explore. If the user thinks the code is related to the query code and

**Table panel:**

Filter by code:

Status	Code	Cosine	FMI	Freq1	Freq2	Description
Select	96522	0.842	0.81	1913	12	refill/maint pump/resvr syst
Select	96523	0.806	1.13	33319	129	irrig drug delivery device
1	0.743	2.34	8805	176	refill/maint portable pump	
2	0.705	0.05	214	1	chemotherapy injectio	
9	0.663	-0.44	486	1	chemotherapy unspecified	

FIGURE 3.4: Showing table panel with code filtering option for “965”.

wants to add it as an inclusion criterion, the “add” is selected. After adding the code, it will be listed in the selected code view table. If a user thinks code is not related to the given query, “not add” will be selected, which fades the row in the table view. If “explore” is selected, a new window opens that allows the user to observe example claims containing this code.

As a researcher, the user would be interested to know more detail by observing actual claims. For this purpose, the “explore” option of the “select” column would help user to see claim view and patient view.

**Claim view:** The claim view shows fifty randomly selected claims in a t-SNE (Maaten and Hinton, 2008) projection that contains the code selected in the table view, as shown in Figure 3.5. We use the vector of primary code in a claim to represent a claim as a vector for the t-SNE projection. The claim view enables a user to explore the claim distance in projected space and also to show some real claim data with code descriptions (**T4**).

In addition, a user would be interested to know about a patient for a specific claim to gain a better understanding of the claim. Our system also shows a patient timeline view when a user clicks on a claim circle in the claim view.

**Patient view:** The patient view presents a complete claim history of a selected patient, as shown in Figure 3.5. The x-axis shows time and the y-axis shows the primary diagnosis of a patient. The selected claim is shown in blue color in the figure. An expert user would focus on this point and look at the past and future claims to understand the health

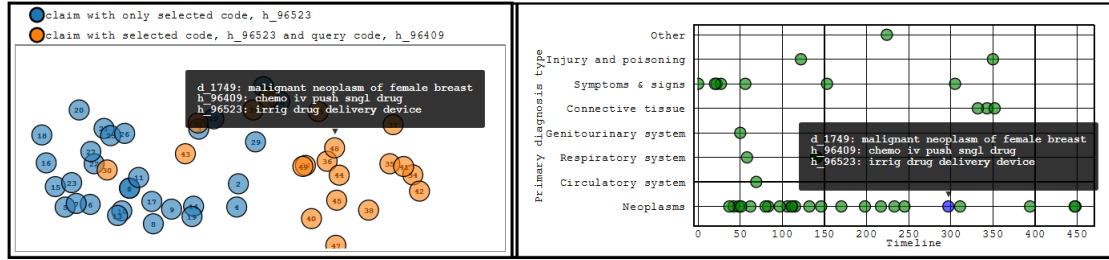


FIGURE 3.5: Claim view and patient view: Claim view shows 50 example claims for recently selected code by a user from table view; Patient view presents timeline for the selected claim from claim view. Here, user selects claim #48 from claim view and patient view shows the patient timeline of the selected claim #48 and highlights in blue circle. The ICD-9 diagnosis and CPT codes are marked with prefix “d\_” and “h\_”, respectively.

conditions of the patient and better reason about the selected claim (T5).

**Selected code view:** Whenever a user decides to add a code from the table view as an inclusion criteria for the given query, the code is listed in this view (T6). After completing a user study, this list would be the final result for a given query or patient treatment. At the below of the selected code list view, the total size of patient and claim cohort is also shown for the selected code.

### 3.5.2 Interaction and Workflow

A video demonstration of MedCV is available at online <sup>1</sup> and below we present a brief overview of the workflow of our system.

A user will interact with the above mentioned views to discover inclusion criteria for a cohort. The workflow of the different views is shown in figure 3.6. A user starts from the query view and then, she finds related code results in scatter and table views. After analyzing code results, she can add codes as inclusion criteria in the table view by selecting “add” option from the “select” button. The selected code will show in the selected code view. However, if a user is interested to see some example claims for a code, she can go to claim view from the table by selecting “explore” option, as shown in figure 3.4. From the

<sup>1</sup> <https://vimeo.com/721257633>

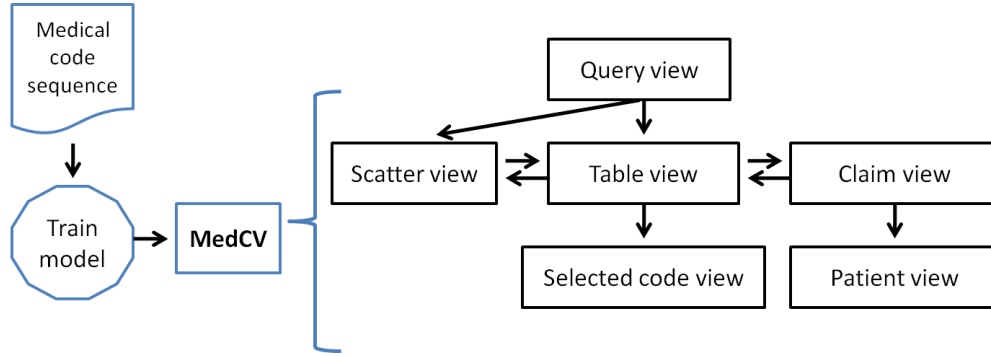


FIGURE 3.6: The workflow of our proposed MedCV software

claim view, the user can also observe the patient view of a selected claim. After examining the claim and patient view, the user would back on the table view to decide whether she is interested to add the code or not. Finally, the selected code view shows the final selected codes for cohort identification.

## 3.6 Evaluation

To evaluate our system, we performed a user study (Lam et al., 2011) to understand if the users believe that it helps them to define inclusion rules for cohort identification task and if they are confident that their rules are good.

### 3.6.1 Dataset

We used a public synthetic dataset from DE-SynPUF<sup>2</sup> that is provided by Centers for Medicare and Medicaid Services (CMS). This dataset contains three types of visit data (inpatient, outpatient and carrier claims) during three years (2008-2010). Although the dataset is synthetic, previous research indicates that it still captures some important properties about the actual patient population it has been derived from (Cai et al., 2018; Paudel et al., 2017). We extracted 66 million daily claims about 1 million synthetic patients. Each record is a set of ICD-9 and CPT codes. There are a total of 14,838 unique

<sup>2</sup> <https://www.cms.gov/Research-Statistics-Data-and-Systems/Downloadable-Public-Use-Files/SynPUFs>

ICD-9 diagnosis codes, 7,195 unique ICD-9 procedure codes, and 11,565 unique CPT codes.

### *3.6.2 Training and Implementation*

We trained word2vec (skip-gram approach)<sup>3</sup> and PMI<sup>4</sup> on this synthetic dataset. We used all the default parameters for the training (i.e. vector length = 100, learning rate = 0.01, epoch = 40, and number of negative samples = 5), other than the window size, which is set to a large number to include all codes in a claim as context. We used MySQL and PHP in back-end to load the training data. Moreover, JavaScript and D3.js is used for front-end development.

### *3.6.3 Baseline*

To evaluate the quality of medical code selection, we used gold standard mappings for procedures related to breast cancer treatments that were manually derived by clinical researchers (Bleicher et al., 2012). For each of breast cancer treatments, a list of the corresponding CPT and ICD-9 procedure codes is provided (the detailed code lists are provided in the Appendix of (Bleicher et al., 2012)).

### *3.6.4 User study*

In our user study, we asked users to identify a breast cancer treatment such as excisional biopsy, mastectomy, or chemotherapy. Our observations of users had two objectives. One was to understand how users interact with the software. Another way was to see how their code selection after only a few minutes of using the software compares to the gold standard, which took domain experts several weeks to compile through a manual selection process.

---

<sup>3</sup> <https://radimrehurek.com/gensim/models/word2vec.html>

<sup>4</sup> <https://radimrehurek.com/gensim/models/phrases.html>

**Participants:** We recruited four medical experts (three females, and one male) to use our software and perform a use case study. Three of them were researchers from a cancer research center who work in medical data analysis fields and very familiar with medical codes. We had another experts from a medical hospital who is a physician and also experienced in patient data analysis. Three of the users have more than 5 years of expertise with claims data, medical codes, and cohort selection; one user has one year of experience in these tasks.

**Tasks and procedures:** We prepared an introductory document and video to explain the goals and tasks of our software to the users. We encouraged our users to “think-aloud” during the sessions. We recorded each session for future analysis.

**User 1:** Our first user was interested in chemotherapy and selected CPT code 96409 (“chemotherapy administration”). She looked at the projection view at first and she easily identified the highlighted query code and neighbor codes of 96409. She found the mouse hover option useful to see code description, frequency, and PMI values. She selected the legend icon to filter out codes with prefix 964 from the projection view (**T3**). From the code descriptions and cosine distance values, she decided to add codes with prefix 964 to the selected code list. She used the table view to add codes of the 964 group, such as 96409, 96411, 96413 as shown in Figure 3.3.

Next, she focused on the codes with prefix 965. She used the filter option of the table view to see only codes that start with 965, as shown in 3.4. She then added 96540 and 96542 to the selected list based on the cosine distance, even though they had low frequency. However, she was not sure about code 96523 (“irregular drug delivery device”), because the code description did not seem related to chemotherapy although it had good cosine distance and PMI values.

In this situation, she was really interested to see some examples of code 96523 in actual claims (**T4**). Hence, she clicked the “explore” from the select option of the table view and opened the claim view, as shown in Figure 3.5. She hovered with the mouse over claim

circles in claim view to see medical codes listed in it. In addition, she wanted to learn about the patients who had that claim. Thus, she clicked on a claim, and our system showed the patient timeline for the selected claim (**T5**). The Figure 3.5 shows the patient of claim 48. The mouse hover option also shows claim codes of each visit day in patient timeline. After observing the patient timeline she focused on claims that had cancer as the main diagnosis (the “Neoplasms” line in the patient view), and noted that code 96523 did not show up in almost any of the claims. After analyzing the claim and patient view, our user decided “not to add” this code in the inclusion criteria. She said this view is really useful to go in-depth. Next, she also checked other codes, but she did not find anything worth adding to the inclusion criteria. Finally, she concluded her task by selecting 16 codes as inclusion criteria for chemotherapy treatment (**T6**).

**User 2:** User 2 was interested in finding a cohort for “local excision of breast” treatment. He entered ICD-9 procedure codes, 8521 for local excision of breast and selected CPT codes to see related codes of the query code. He observed the software result and found couples of CPT codes such as 19120, 19295, 19296, 76098, 19290, 38525, and 19316 that are related to the given code.

Then, he sorted the table list based on the PMI values. Then, he noticed that he selected 76098 (“x-ray exam breast specimen”) and 19290 (“place needle wire breast”) CPT codes before, but the codes had very low PMI values. After checking the PMI values, he changed his mind and deleted codes 76098 and 19290 from the selected code list. Finally, he was happy with his 5 selected codes. He said that he felt PMI value is more useful than cosine distance value in cohort identification because PMI is calculated based on the co-occurrence of codes.

**User 3:** Our third user also selected chemotherapy treatments for her user study. She liked the projection view. She studied not only cosine distance and PMI value but also code frequencies. After a long observation and careful analysis, she selected a total of 11 codes as inclusion criteria for the cohort identification. She stated that the “select” and

“filter” options were the most important features of our MedCV system for her.

**User 4:** Our fourth user started with a different CPT code, 96411 (“chemotherapy IV push”). She found the third frequency value of the table view interesting since it reveals the co-occurrence of the query code and the corresponding table row code. Our fourth user was also not sure about 96521 and 96523. She said these codes are challenging for cohort identification. She used the claim view and patient timeline to analyze these codes. She eventually decided not to add any code from the 965 prefix group. She believed the idea of using underlined text for the code that has low PMI value ( $\leq 1$ ), is useful, as shown in Figure 3.3(c) for code 82378. She finally selected 11 codes for the cohort identification.

### *3.6.5 Comparing with Gold Standard Data*

For the “chemotherapy” treatment, there are 21 codes listed in the gold standard. All four participants identified at least 10 codes, ranging from 11 to 16. None of the participants selected a code that was not in the list. This is an impressive result, considering that 4 of the codes in the gold standard did not occur in more than 10 claims in the dataset.

For the “local excision of breast” treatment, the gold standard included 3 CPT codes. Our participant found all 3 CPT codes and selected 2 additional codes that were not in the gold standard. After our consultation with the authors of Bleicher et al. (2012), we concluded that those 2 additional codes are indeed related to biopsy or excision of breast (i.e. 19316 “suspension of breast”, 38525 “biopsy/removal lymph nodes”). Thus, we do not consider the selection to be incorrect.

### *3.6.6 Expert Interview and Feedback*

We also validated the utility of MedCV by recording the subjective feedback of our expert users. At the end of each user session, we asked a set of structured questions to understand how users felt about their interaction with MedCV. We asked them whether the tool allowed them to make reliable judgments about their task. Based on their answers

and their behavior during the experiment, we group the user feedback into the following categories:

**i) Effectiveness in cohort selection:** All of our users thought that MedCV is effective in finding related codes and displaying code similarity based on different metrics. Showing examples of claim data and patient timeline also played an important role in effective decision making. However, our second user believed that there should be subsequent validation methods using other data, such as demographics and survival, that would further inform users in code selection.

**ii) Human-computer interaction:** Our participants appreciated the level of interaction and flexibility that we allowed them to search codes and analyze their similarity. Our third participant found that the filter option is really helpful in finding code and saving time. She commented that “This filter option helps to save time and look over all of the group codes together very easily”. The users liked the tooltips text on user interaction that shows additional information of a user action. They said this option helps them to get the necessary information quickly and easily.

**iii) Advantages over the state of the art:** Our expert users believe the visual system has many advantages over the current state of the art. Using this tool, users can easily avoid the large scale data handling problems. They thought the tool is useful in finding related medical codes and analyzing them for patient cohort selection. It also presents patient timeline to understand patient health status and consequences of claim data. Our fourth user liked the patient view the most and she commented that “I like the timeline aspect in the patient view the most. It tells me what happened to a patient if he had the claim or any claims related to the claim”.

**iv) Potential for adoption:** The users believe the tool has potential for extensions such that it could be adopted to related tasks. For example, our first participant worked in cost-effectiveness studies in the medical domain and she believes our software could be a potentially useful tool in this research field. She said - “it could potentially be useful

to categorize claims as either attributable or non-attributable to a particular diagnosis for studying costs of care.” Similarly, our fourth user works on prostate cancer data analysis. She is also interested to see how our software would work on her research data.

**v) Shortcomings:** Our expert users also noted some points that they think we should consider for further improvements. Our second participant observed that there is a learning curve to become familiar with the software, but once that is reached, it is easy to use. For this reason, we create an introductory video that explains the functionality of our software with a sample user study. For our third user, the t-SNE plot was not clear. She could not understand the code relationships from the plot. Our fourth user told us that the underlined code text of the projection view for low PMI should be highlighted in a different way, so that the user would be interested to explore the code. After her comment, we highlight the underline code text with a blue colored circular border.

In our question set, we asked users to fill out 15-question survey where Q1-Q10 are objective and Q11-Q15 are subjective questions. The questions are designed to investigate how our MedCV tool helps from the visualization wise, method wise, and feature wise. In the question set, users rate different views of MedCV on a scale from 1 to 10. Similarly, we also requested them to rate our visual tool on different criteria, such as effectiveness, flexibility, advantages, and future adoption. Figure 3.7 shows the average feedback score of our expert users on software designs and performance. The result tells us that most of the users believe table view plays an important role in design tasks, while some of them found the scatter plot view confusing. At the same time, users were confident about the effectiveness, advantages, and future adoption of our software. However, we need to improve the flexibility options, so that expert users can perform exploratory analysis even more efficiently and with less cognitive load. In future, we will find more expert users as volunteers for testing our MedCV tool and getting feedback to improve the current version.

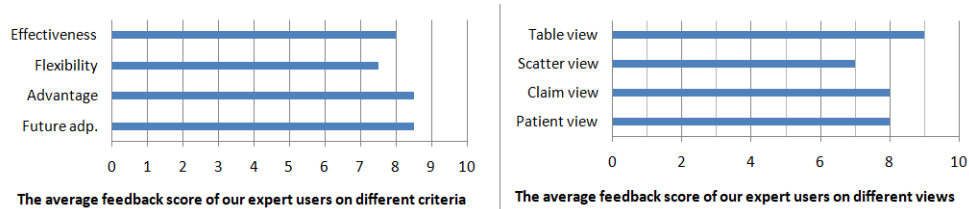


FIGURE 3.7: Expert users feedback on survey questions

### 3.7 Conclusions

In this paper, we described a visual tool, MedCV that aims to assist researchers and clinicians to define inclusion criteria for patient cohort identification from medical claim data. We demonstrated that the visualization of codes, claims, and patient timelines positively impact the reasoning of domain experts during the cohort identification task. For the evaluation of our software, we conducted with multiple expert users to perform user studies and we reported the qualitative results. Future research will consider ways to enable users to analyze patient demographic information in addition to medical claims. It will also focus on integrating our software with statistical tools used in retrospective studies to produce a one-stop resource available to medical researchers working with medical claims data.

## CHAPTER 4

# CONCORDANCE BETWEEN KDIGO DEFINITION OF ACUTE KIDNEY INJURY AND ITS CODING IN CLINICAL PRACTICE

### 4.1 Introduction

Acute kidney injury (AKI) represents a sudden decrease in kidney function characterized by the accumulation of urea and other waste products, imbalances in extracellular volume and electrolytes, and reduced urinary output (Palevsky, 2020; Lameire et al., 2021). Clinical manifestation of AKI spans a continuum of kidney diseases from decreased glomerular filtration rate and reduced kidney function to complete organ failure (Palevsky, 2020; Levey et al., 2020; Lameire et al., 2021; Kellum et al., 2021). The current clinical definition of AKI recognizes that smaller reductions in kidney function that do not result in overt organ failure are of substantial clinical relevance and are associated with increased morbidity and mortality; thus, the term AKI has largely replaced acute renal/kidney failure (ARF/AKF) (Palevsky, 2020; J. et al., 2020).

Although the etiology and pathophysiology of AKI vary, the condition occurs commonly in critically ill hospitalized patients and is associated with increased morbidity, mortality, and medical costs (Griffin et al., 2020; Hoste et al., 2018; Kellum et al.,

2021). Traditionally, clinical conditions underlying AKI have been categorized as prerenal (due to decreased renal perfusion pressure), intrinsic renal (pathology of the kidney vessels, glomeruli, or tubules-interstitium), or postrenal (obstruction of urinary flow). The primary cause of AKI in hospitalized patients, accounting for about 45% of AKI cases, includes acute tubular necrosis (ATN) from ischemia related to sepsis, hypotension, or exposure to nephrotoxic medication such as antibiotics, imaging contrast, ACEi/ARB, and non-steroid anti-inflammatory medication (Griffin et al., 2020; Erdbruegger and Okusa, 2018). Prerenal disease due to hypervolemic (heart failure with reduced ejection fraction, decompensated liver disease, use of medications causing vasoconstriction or affecting renal flow) or hypovolemic state (acute hemorrhage, diarrhea) accounts for an additional 25% of AKI cases (Erdbruegger and Okusa, 2018). Furthermore, hospitalized patients often have additional risk factors and comorbidities for AKI, such as older age, male sex, diabetes, heart failure, and hypoalbuminemia (Hoste et al., 2018). Because both the presence and severity of AKI are associated with worse outcomes, including increased mortality, increased length of hospital and ICU stay, a requirement for dialysis, and an increased risk of subsequent chronic kidney disease, improvements in early detection of AKI can lead to appropriate management and better patient outcomes.

Early and accurate detection of AKI still presents a diagnostic challenge due to a variety of clinical presentations and a lack of uniformity in criteria defining AKI. Patients with AKI may present with typical signs and symptoms of reduced kidney function, such as edema, hypertension, and/or decreased urinary output. However, some patients with AKI may lack overt clinical symptoms and a loss of kidney function is detected only by the laboratory tests (Fatehi and Hsu, 2017). Serum creatinine (SCr) is a biomarker most commonly used in clinical practice as a surrogate for kidney excretory function and remains the only lab value used in formal definitions of AKI (Fatehi and Hsu, 2017; Kellum et al., 2021). Currently, the accepted definition of AKI is based on consensus-based criteria sequentially developed by the Kidney Disease Improving Global Outcomes

(KDIGO) initiative (Kellum et al., 2012; Levey et al., 2020; Lameire et al., 2021). Criteria for AKI include a sudden decrease in glomerular filtration rate manifested by an increase in serum creatinine or decrease in urine output (oliguria) within 48 hours to 7 days, with the severity (stage) of AKI determined by the severity of increase in serum creatinine or oliguria. Specifically, KDIGO guidelines (Kidney Disease: Improving Global Outcomes Work Group 2012) define AKI as follows

- increase in serum creatinine by  $\geq 0.3$  mg/dL within 48 hours, or
- increase in serum creatinine to  $\geq 1.5$  times baseline, which is known or presumed to have occurred within the prior seven days, or
- urine volume  $< 0.5$  mL/kg/hour for six hours.

Using the same KDIGO criteria AKI is staged as follows

- Stage 1: Increase in serum creatinine to 1.5 to 1.9 times baseline, or increase in serum creatinine by  $\geq 0.3$  mg/dL, or reduction in urine output to  $< 0.5$  mL/kg/hour for 6 to 12 hours
- Stage 2: Increase in serum creatinine to 2.0 to 2.9 times baseline, or reduction in urine output to  $< 0.5$  mL/kg/hour for  $\geq 12$  hours
- Stage 3: Increase in serum creatinine to 3.0 times baseline, or increase in serum creatinine to  $\geq 4.0$  mg/dL, or reduction in urine output to  $< 0.3$  mL/kg/hour for  $\geq 24$  hours, or anuria for  $\geq 12$  hours, or the initiation of kidney replacement therapy, or, in patients  $< 18$  years, decrease in estimated glomerular filtration rate (eGFR) to  $< 35$  mL/min/1.73 m

Current KDIGO criteria evolved from previous efforts to define AKI. RIFLE (Risk, Injury, Failure, Loss, and End Stage Renal Disease) criteria were initially proposed by the Acute Dialysis Quality Initiative (ADQI) in 2004 and used a 50% increase in serum

creatinine or 25% decrease in eGFR as a threshold for the lowest severity or “risk” stage and the increasing stages of disease severity were labeled “injury,” “failure,” “loss,” and “End Stage Renal Disease” (Bellomo et al., 2004). Between 2005-2007, Acute Kidney Injury Network (AKIN) Classification System proposed the terminology of AKI to represent the entire spectrum of acute renal failure, eliminated RIFLE’s Loss and ESRD stages and eGFR criteria, and given studies showing mortality associated with small increases in creatinine included absolute increase in serum creatinine of  $\geq 0.3$  mg/dl, a  $\geq 50$  % percentage increase in serum creatinine, or a reduction in urine output of less than 0.5 ml/kg/hr as a criteria for Stage 1 AKI (Mehta et al., 2007). Therefore, the current KDIGO AKI definition built on the AKIN criteria specifying the timing of increase in *SCr* of either 0.3 mg/dl within 48 hours or a 50% increase within 7 days.

Harmonized definition of AKI using KDIGO criteria improved epidemiological and clinical research allowing for consistent definition of population inclusion criteria and/or endpoints for clinical studies (Palevsky, 2020; Birkelo et al., 2022). The research allowed for evaluating the burden of disease in the critically ill patients and in the community, risk factors and outcomes (Hoste et al., 2018). KDIGO has been used in several papers as the ground truth for AKI (Kate et al., 2016; Li et al., 2018; Sun et al., 2019). Other papers report on prospective and retrospective studies that indicate that KDIGO can be used as a predictor of outcomes, including mortality (Bıyık et al., 2016).

However, there are a number of challenges with the KDIGO definition of AKI. Specifically, KDIGO AKI criteria do not distinguish between the multiple etiologies that cause AKI (Palevsky, 2020) and using urine output to define or stage AKI is not based on robust evidence (Palevsky, 2020) and it is impractical to use outside of ICU. Perhaps the biggest limitation is that the KDIGO formula can be interpreted and/or implemented in numerous ways (Wiersema et al., 2020). Because kidney function using KDIGO criteria is assessed based on change in baseline *SCr*, in patients presenting with AKI who do not have a baseline measurement of *Scr*, KDIGO formula cannot be applied (Palevsky, 2020).

Several workarounds have been proposed to best approximate the true baseline creatinine value, including using the first presenting *SCr* measurement (ad-hoc working group of ERBP: et al., 2012) or an average of creatinine values measured 7 to 365 days prior to hospitalization (Siew et al., 2012). Beyond clinical practice where access to baseline *SCr* is likely impractical, Guthrie et al found that there is wide variation and a lack of transparency in how KDIGO AKI definition is applied in clinical and epidemiological research (Guthrie et al., 2021) which reduces the ability to generalize and compare results across different studies.

Although the KDIGO criteria helped to define the AKI for epidemiological and clinical research, the adoption of KDIGO criteria for clinical management of patients with AKI and its utility in clinical practice is debated and not completely understood (Birkelo et al., 2022). Interestingly, in critical care, renal function is only one of the organs evaluated to quickly understand a patient's overall functional status. In that setting, a simple SOFA score is routinely used to assess the performance of neurologic, blood, liver, kidney, and blood pressure/hemodynamics to describe a sequence of complications in the critically ill. Renal component of SOFA score does not use KDIGO AKI criteria and is instead based on *SCr* threshold with the lowest score (best functional status) of 1 assigned at *SCr* threshold of 1.2-1.9 mg/dL and the highest score of 4 at *SCr* threshold of > 5.0 mg/dL. Similarly, the official diagnosis and procedure codes (ICD-9 and ICD-10) do not recognize AKI. The codes closest to AKI are a group of codes with prefix 584 in ICD-9 and prefix N17 in ICD-10 that refer to acute kidney failure.

To gain an insight into the clinical uptake of KDIGO criteria for the diagnosis of AKI, we study concordance between KDIGO AKI detection and ICD coding of kidney injury by medical providers. Our retrospective study includes 47,499 ICU admissions between 2008 to 2019 covering 38,676 patients of Beth Israel Deaconess Medical Center (Johnson et al., 2020). High degree of concordance between KDIGO and ICD coding in MIMIC-IV admissions would indicate that KDIGO formula matched the clinical understanding of

AKI/AKF in Beth hospital during the study period. Moreover, since KDIGO was proposed in 2012, it could be expected that the concordance was higher after 2012.

## 4.2 Methods

This section describes the cohort for this study and the methods used to analyze the data.

### 4.2.1 Cohort Selection

Fourth edition of the Medical Information Mart for Intensive Care (MIMIC-IV) is a de-identified medical data set that contains electronic health records of BIDMC intensive care unit (ICU) patients between 2008 and 2019 (Johnson et al., 2020). This data set records various types of patient health information, such as patient demographics, charted events, lab events, procedures, and ICD-9 or ICD-10 codes. One patient could have a record for multiple admissions and each admission might contain one or more visits to ICU. There is a total of 69,619 ICU visits in 64,975 admissions of 50,048 patients. Our cohort included

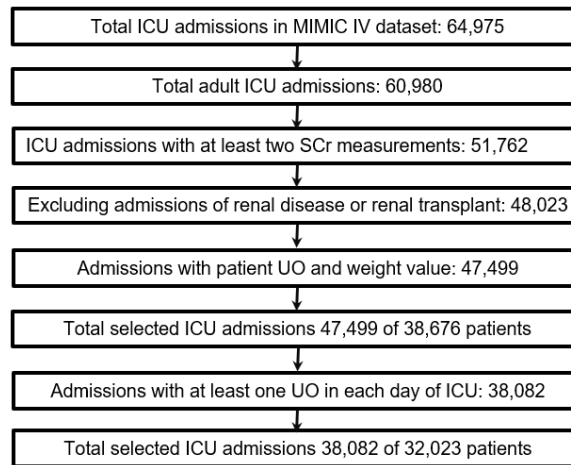


FIGURE 4.1: Selecting population from MIMIC-IV dataset

adult patients ( $\text{age} \geq 18$ ) who have at least two serum creatinine (SCr) measurements and at least one urine output (UO) during the admission, have recorded weight, have exactly one ICU admission during the admission, and have no record of renal transplant at discharge

Table 4.1: ICD code description for AKI

<b>ICD-9 code</b>	<b>Code description</b>
584	Acute kidney failure
584.5	Acute kidney failure with lesion of tubular necrosis
584.6	Acute kidney failure with lesion of renal cortical necrosis
584.7	Acute kidney failure with lesion of renal medullary [papillary] necrosis
584.8	Acute kidney failure with other specified pathological lesion in kidney
584.9	Acute kidney failure, unspecified
<b>ICD-10 code</b>	<b>Code description</b>
N17	Acute kidney failure
N170	Acute kidney failure with lesion of tubular necrosis
N171	Acute kidney failure with lesion of renal cortical necrosis
N172	Acute kidney failure with medullary necrosis
N178	Other acute kidney failure
N179	Acute kidney failure, unspecified

(ICD codes for identifying renal transplant are given in the Appendix). The resulting cohort (Cohort-1) consists of 47,499 admissions covering 38,676 patients.

UO measurements are unreliable, which makes it difficult to use UO in KDIGO criteria. There are only 7,949 admissions in Cohort-1 with at least one *UO* measurement during each hospitalization day. There are only 2,405 admissions with at least one *UO* measurement during each 6-hour time window of the ICU stay. Thus, we decided to create Cohort-2 from Cohort-1 by retaining only the 38,082 admissions with at least one *UO* measurement during each day of ICU stay. Figure 4.1 summarizes the inclusion criteria for Cohort-1 and Cohort-2. We extracted the following information for each admission: log of UO and SCr measurements, admission and discharge times and dates, mortality in

hospital, ICD procedure and diagnosis codes, and ICU type. We note that the exact date of admission is obfuscated due to de-identification, such that a year of admission is mapped to a year range (there are four ranges: 2008-2010, 2011-2013, 2014-2016, 2017-2019).

#### 4.2.2 AKI Diagnosis Codes

As explained in the introduction, ICD-9 and ICD-10 diagnosis codes do not contain term “acute kidney injury.” Terms “acute kidney injury” (or AKI) and “acute kidney failure” (or AKF) have been used interchangeably in the literature and clinical practice (Palevsky, 2020; J. et al., 2020). Previous research has explicitly used the ICD-9 and ICD-10 codes for AKF as indication of AKI diagnosis (Keddis et al., 2012; Vlasschaert et al., 2011; Li et al., 2009; Jamal et al., 2020). Following this previous research, we use codes listed in Table 4.1 to identify patients diagnosed with AKI. There are 13,176 admissions (27.7%) in Cohort-1 with AKI diagnosis.

#### 4.2.3 KDIGO Implementation

The Kidney Disease Improving Global Outcomes (KDIGO) (Kellum et al., 2012) uses two criteria based on increase in *SCr* and one criterion based on *UO*. If at least one of the three criteria is satisfied, KDIGO detects AKI in a patient. As noted in the previous research (Wiersema et al., 2020), there are multiple ways to implement KDIGO criteria depending on the definition of *SCr* baseline and use of *UO* values. Since MIMIC-IV does not list any *SCr* values prior to admission (excluding a small subset of patients with a previous recorded admission), we had to rely only on *SCr* values recorded during the admission. We implemented three versions of KDIGO previously recommended in (Wiersema et al., 2020):

**I** can be applied on Cohort-1 and Cohort-2 data. It uses the two *SCr* criteria and ignores the *UO* criterion from KDIGO. For every *SCr* measurement, it compares the measurement to the previous *SCr* measurements during the last 48 hours (criterion 1) and

		ICD AKI coding	
		ICD AKI	No ICD AKI
KDIGO	KDIGO positive	True positive (TP)	False positive (FP)
	KDIGO negative	False negative (FN)	True negative (TN)

FIGURE 4.2: Showing confusion matrix for KDIGO and ICD AKI

the last 7 days (criterion 2). It uses the first *SCr* value in a 7-day window as the baseline (criterion 2). If there are no *SCr* measurements during the time window, the criterion is ignored. If AKI is detected by either criterion for any *SCr* measurement, the whole admission is labeled as AKI according to KDIGO.

**I2** differs from **I1** by the definition of baseline; instead of the first *SCr* measurement within a 7-day window it uses the lowest *SCr* within the window (Coca et al., 2007).

**I3** can be applied on Cohort-2. It differs from **I1** by including criterion 3 that uses *UO*, which was modified as recommended in (Wiersema et al., 2020) to consider 24-hour window instead of the 6-hour window:

- (criterion 3) Urine volume < 0.3 ml/kg/h for 24 hours

#### 4.2.4 Concordance Analysis

The main objective of this paper is to study concordance between the hospital-provided AKI ICD diagnosis (Section 4.2.2) and KDIGO AKI definition (Section 4.2.3). To study the concordance, we calculate  $2 \times 2$  confusion table illustrated in Figure 4.2, where rows correspond to KDIGO AKI definition and columns to ICD AKI diagnosis. The four elements of the confusion table are 1) True Positives (TP): count of admissions where both KDIGO and ICD recognize AKI, 2) False Positives (FP): count of admissions where AKI is predicted by KDIGO but AKI ICD is not listed, 3) False Negatives (FN): count of admissions where AKI is not predicted by KDIGO but AKI ICD is listed, and 4) True

Negatives (TN): count of admissions where neither KDIGO nor ICD recognize AKI.

We calculate the following concordance measures from the confusion table:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \quad (4.1)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (4.2)$$

$$\text{Recall (or TPR)} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (4.3)$$

$$\text{F1} = \frac{2 \times (P \times R)}{(P + R)} \quad (4.4)$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \quad (4.5)$$

Accuracy is a fraction of admissions where KDIGO and ICD agree. F1 measures a balance between precision (fraction of KDIGO AKI positive admissions agreed by ICD) and recall or true positive rate (TPR) (fraction of ICD AKI positive admissions agreed by KDIGO). False positive rate (FPR) is a fraction of ICD AKI negative admissions agreed by KDIGO AKI.

#### 4.2.5 Prediction of ICD AKI from SCr

In order to get a better insight into concordance between KDIGO criteria and ICD AKI coding, we trained several models to predict ICD AKI coding from SCr measurements. We created the following three variables from the SCr time series:

**[KDIGO-I]:** Maximum increment of *SCr* during any 48 hour period. We note that if KDIGO-I is larger than 0.3 at any point during the hospital visit, KDIGO criterion 1 detects AKI.

**[KDIGO-R]:** Maximum *SCr* increase ratio during any 7 day period. We note that if KDIGO-R is larger than 1.5 at any point during the hospital visit, KDIGO criterion 2 detects AKI.

**[maxSCr]:** Maximum *SCr* during the whole hospital stay.

We used logistic regression (LR) and decision tree (DT) models with different combinations of the three *SCr* variables to predict ICD AKI. We randomly divided Corpus-1 admissions into train, validation, and test sets with 70-15-15 proportion, respectively.

To train DT, we used the post pruning method on the validation data. Other hyperparameters for LR and DT were the default choices in sklearn Python package <sup>1</sup>.

Table 4.2: Concordance between ICD AKI and different implementations of KDIGO.

	<b>TP</b>	<b>FP</b>	<b>FN</b>	<b>TN</b>	<b>Acc (%)</b>	<b>Prec</b>	<b>Recall</b>	<b>FPR</b>	<b>F1</b>
	Population: 47,499								
I1	7,397	6,411	5,779	27,912	<b>74.3</b>	0.53	0.56	0.19	<b>0.55</b>
I2	7,589	7,939	5,587	26,384	71.5	0.48	0.57	0.23	0.53
	Population: 38,082								
I1	6,405	5,676	4,435	21,566	73.4	0.53	0.59	0.21	0.55
I3	7,906	13,106	29,34	14,136	57.8	0.37	0.72	0.48	0.49

## 4.3 Results

### 4.3.1 Concordance for Different KDIGO Implementations

The first two rows of Table 4.2 compare KDIGO implementations I1 and I2 on Cohort-1 data. We can see that implementation I1 results in slightly higher concordance. Concordance between ICD AKI and KDIGO implementation I1 shows that there are sizeable FP (6,411) and FN (5,779) disagreements, relatively low accuracy 74.3%, similar precision (0.53) and recall (0.56), and 0.55 F1.

The last two rows of Table 4.2 compare KDIGO implementations I1 and I3 on Cohort-2 data. We can see that implementation I1 results in a significantly higher concordance. It

<sup>1</sup> <https://scikit-learn.org/stable/>

is interesting to observe that implementation I3 (which adds UO-based criterion 3) resulted in a much higher number of KDIGO AKI positives (7,906+13,106) than implementation I1 (6,405+5,676). However, most of the KDIGO AKI positives added by I3 were FP (did not agree with ICD AKI), which resulted in decreased concordance.

Based on the results in Table 4.2, in the rest of the paper, we will use KDIGO implementation I1 (that uses the first *SCr* in the 7-day window as the baseline) and will focus on Cohort-1. I1 implementation of KDIGO detected 13,808 (7,397+6,411) AKI admissions with prevalence of 29.08% on Cohort-1.

Table 4.3: Statistics of clinical variables and outcomes in different subgroups: average number of *SCr* measurements, average SOFA renal score, percent of admissions with administered vasopressor, average hospital length of stay, average ICU length of stay, and percent of in-hospital mortality. (Here, vaso. = vasopressor, mort. = mortality, dial. = dialysis)

	<b>Total</b>	<b>ICD +</b>	<b>ICD -</b>	<b>KD +</b>	<b>KD -</b>	<b>TP</b>	<b>FP</b>	<b>FN</b>	<b>TN</b>
LOS	8.7	11.0	7.8	12.5	7.2	13.9	10.9	7.3	7.1
LOS ICU	3.4	4.4	2.9	5.1	2.7	5.8	4.2	2.6	2.7
# <i>SCr</i>	10.9	15.9	9.0	17.8	8.1	21.0	14.0	9.6	7.8
SOFA renal	0.58	1.48	0.22	1.2	0.3	1.76	0.55	1.13	0.15
Dial. (%)	1.5	4.7	0.2	4.3	0.3	7.6	0.6	1.0	0.2
Vaso. (%)	33.1	42.0	29.8	51.3	25.5	51.1	51.6	30.4	24.6
Mort. (%)	8.2	16.7	5.0	16.9	4.6	23.0	10.0	8.6	3.8

#### 4.3.2 Analysis of TP, FP, FN, TP Admissions

This subsection provides analysis of the four concordance groups. From Table 4.3, we observe that admissions with ICD AKI diagnosis have longer stay in hospital (11.0 versus 7.8 days) and ICU (4.4 versus 2.9 days) with more *SCr* measurements (15.9 versus 9.0). They have much higher in-hospital mortality (16.7% versus 5.0%). There is much larger incidence of dialysis treatment (4.7% versus 0.2%) and slightly larger use of vasopressor (42.0% versus 29.8%). We report vasopressor use because vasopressor therapy is used for

AKI but also for heart failure, low blood pressure problems, and sepsis.

Table 4.3 also lists (Sequential Organ Failure Assessment) (SOFA) renal score calculated from the maximum  $SCr$  during the first ICU day as follows:

If  $\max SCr > 5.0$ , then score 4,

If  $3.5 < \max SCr \leq 5.0$ , then score 3,

If  $2.0 < \max SCr \leq 3.5$ , then score 2,

If  $1.2 < \max SCr \leq 2.0$ , then score 1,

If  $\max SCr \leq 1.2$ , then score 0.

It can be seen that SOFA renal scores are much higher in ICD positives (1.48 versus 0.22), indicating reduced kidney function upon ICU admission.

Comparison of AKI positive and negative admissions based on KDIGO criteria shows that positives have much higher in-hospital mortality (17.0% versus 4.9%), longer length of stay in hospital (12.6 versus 7.1 days) and ICU (12.6 versus 7.1 days), and higher use of vasopressor (51.3% versus 26.0%). Interestingly, the difference in SOFA renal score is not as large (1.2 versus 0.4) as between AKI positives and negatives based on ICD diagnosis.

Among the four concordance groups, it is evident that TP admissions had the most severe cases, with the highest mortality, length of stay, SOFA renal scores, and the highest incidence of dialysis. FP admissions had slightly higher mortality and LOS but lower SOFA renal scores than FN admissions. Finally, TN admissions had the lowest mortality and SOFA renal scores.

Table 4.3 reveals interesting relationship between SOFA renal scores, KDIGO criteria, and AKI diagnosis by ICD. To gain further insight, Figure 4.3 compares distributions of first and maximum  $SCr$  of each admission in the four concordance subgroups and the whole Cohort-1. Each boxplot shows minimum (bottom tick), first quartile (bottom of the box), median (orange line), mean (green dashed line), third quartile (top of the box), and maximum (top tick).

As expected, the maximum  $SCr$  is higher than the first  $SCr$  in TP and FP admissions

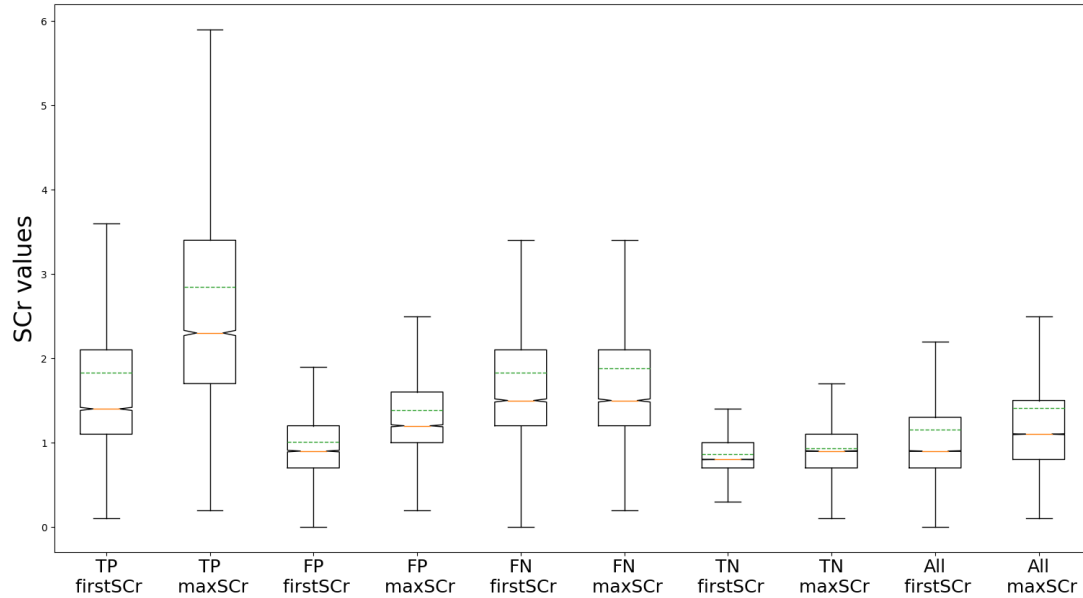


FIGURE 4.3: The distribution of first SCr and maxSCr value in different subgroups of Cohort-1

because the increase in SCr triggers KDIGO criteria. TP admissions had the highest first SCr (median 1.40 and mean 1.83), and substantially higher maximum SCr (median 2.30 and mean 2.85) than the other three subgroups. Unlike TP, the first SCr for FP admissions was relatively low (median 0.90 and mean 1.01), with the maximum SCr (median 1.20 and mean 1.39) not substantially higher than the first SCr. In 3,409 of the 6,411 FP admissions, SCr remained below 1.2 (corresponding to SOFA renal score 0), and in 2,332 FP admissions, maximum SCr was between 1.2 and 2 (corresponding to SOFA renal score 1).

The difference between the first and maximum SCr in TN and FN admissions was negligible (mean increased from 0.87 to 0.93 in FN and from 1.83 to 1.88 in TN). This result is expected because KDIGO is not triggered when there is a lack of significant SCr increase. However, it is important to observe that the first SCr values in TN (mean 1.83) were substantially higher than the first and maximum SCr values in FP (mean 1.01 and 1.39). In 72.2% of the FN admissions, the first SCr value was also the maximum SCr

value.

The FN and FP results in Figure 4.3 strongly indicate that the maximum SCr value during the hospital stay was strongly correlated with AKI diagnosis provided by the hospital in Cohort-1. The next subsection shows machine learning results aimed at better understanding the relationship between KDIGO criteria and ICD AKI diagnosis.

Table 4.4: Results of different predictive models for hospital-based AKI diagnosis (cohort-1: 47,499)

	<b>Features</b>	<b>TP</b>	<b>FP</b>	<b>FN</b>	<b>TN</b>	<b>Acc%</b>	<b>Prec</b>	<b>Recall</b>	<b>FPR</b>	<b>F1</b>
I1	-	1,092	947	902	4,184	74.05	0.53	0.55	0.18	0.54
LR	maxSCr [1.78]	1,171	253	823	4,878	84.90	0.82	0.58	0.05	0.68
$DT_1$	maxSCr [1.35]	1,611	694	383	4,437	84.88	0.69	0.80	0.13	0.74
DT	maxSCr	1,487	525	507	4,606	85.52	0.73	0.75	0.10	0.74
DT	KDIGO -I,R	871	286	1,123	4,845	80.22	0.75	0.43	0.05	0.55
DT	KDIGO -I,R, maxSCr	1,503	514	491	4,617	<b>85.89</b>	0.74	0.75	0.10	<b>0.75</b>

### 4.3.3 Prediction of Hospital-based AKI Diagnosis from SCr

Table 4.4 compares concordance of several machine learning models with hospital-based ICD AKI coding. LR refers to logistic regression on maxSCr variable.  $DT_1$  refers to a DT using maxSCr variable with a single question of type “is maxSCr larger than a threshold?” Three DT rows refer to DTs grown and pruned with three different combinations of SCr variables. We note that all the results are calculated on test data, which are 15% of randomly selected Cohort-1 admissions.

The results show that the three machine learning models trained only with maxSCr variable achieve much higher concordance than KDIGO criteria. Moreover, DT trained on KDIGO-I and KDIGO-R variables achieves slightly higher but comparable concordance to KDIGO. Finally, adding KDIGO-I and KDIGO-R to maxSCr increases DT concordance

only slightly compared to DT trained only on SCr. These results strongly suggest that SCr is a superior predictor of hospital-based AKI diagnosis compared to KDIGO criteria.

The number in parenthesis in LR and  $DT_1$  models is decision threshold for maxSCr variable. The threshold for LR was determined as maxSCr value that results in LR output 0.5 (equal probability of positive and negative AKI). The threshold for  $DT_1$  is available explicitly from the first question. If maxSCr is higher than the threshold the machine learning model is predicting AKI. The result shows that 1.78 threshold results in better F1 score than 1.35 threshold (0.74 versus 0.68) indicating better balance between precision and recall. Higher threshold results in higher precision but lower recall. We note that both thresholds are between 1.2 and 2.0, which corresponds to SOFA renal score 1. **ROC**

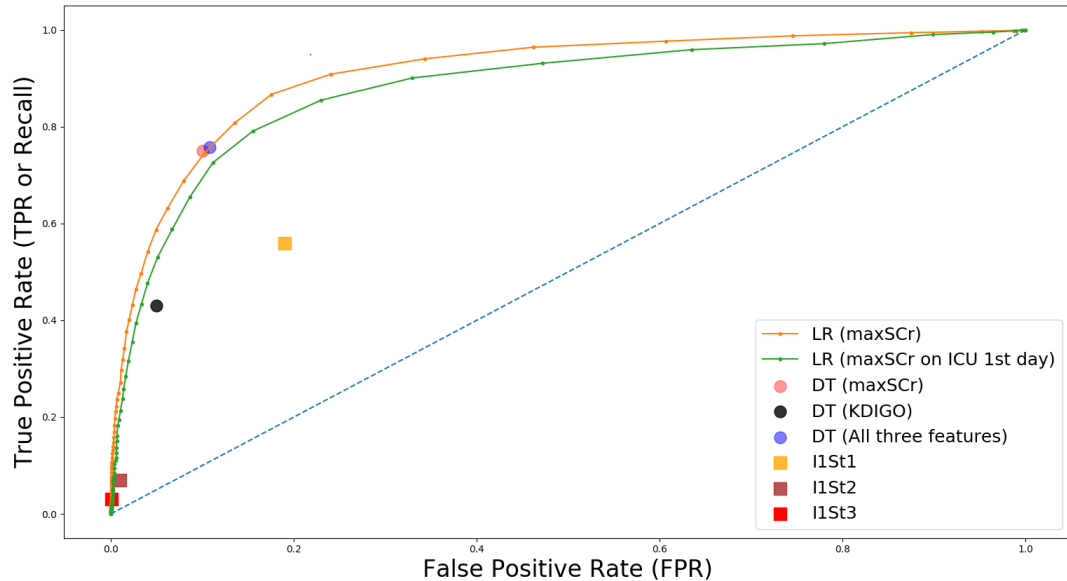


FIGURE 4.4: ROC curve for a single feature with (TPR, FPR) points for three decision tree and three stages of I1 KDIGO implementation. [ROC AUC score are 0.91 and 0.88 for LR with maxSCr during hospital and first day of ICU, respectively]

**curve analysis** is a useful tool for jointly studying True Positive Rate (TPR) or recall and False Positive Rate (FPR) of different prediction models. While some models, such as DT provide a single combination of TPR and FPR values, other models provide different combinations depending on threshold. In particular, the orange line in Figure 4.4 shows

how TPR and FPR change as maxSCr threshold is increased from a very small to a very large value. This particular plot is called the ROC curve. The green line shows the ROC curve for SCr1 (referring to SCr during first ICU day). The dotted blue line represents the ROC curve of a random classifier. A good model should have an ROC curve that is higher than the blue line and is as close to TPR = 1 and FPR = 0. In addition to ROC curves, Figure 4.4 also contains TPR and FPR combinations for KDIGO and three DT models.

We can observe that maxSCr ROC curve is superior to SCr1 curve. KDIGO is significantly below the orange ROC curve, which again indicates its lower concordance with hospital-based AKI diagnosis. DTs that use maxSCr variables lie almost exactly on the orange ROC curve, while DT that uses KDIGO variables is below it.

#### *4.3.4 Does Concordance Differ Based on ICU Type?*

Six types of ICUs cover more than 95% of all MIMIC-IV ICU admissions: MICU (Medical ICU), CVICU (Cardiac Vascular ICU), MISICU (Medical/Surgical ICU), SICU (Surgical Intensive Care Unit), TSICU (Trauma Surgical Intensive Care Unit), and CCU (Coronary Care Unit). Table 4.5 and 4.6 summarizes concordances of different models on test data from Cohort-1 for each of those six types of ICU.

The results show that for all 6 types of ICU, KDIGO concordance is always significantly lower than for maxSCr variable. Looking at the F1 scores, KDIGO had the smallest concordance on CVICU (0.42) and TSICU (0.46) and the highest concordance on CCU (0.63). DT1 achieved the highest F1 concordance on MICU (0.81) and MISICU (0.80).

#### *4.3.5 Does Concordance Change Over Time?*

Table 4.7 shows the concordance between the KDIGO definition and hospital-based AKI diagnosis over time, calculated for four different year ranges. It can be seen that accuracy and F1 score improved over time, from 0.53 during 2008-2010 to 0.62 during 2017-2019.

Table 4.5: Results of different predictive models on ICU unit types (MICU, CVICU, MISICU)

Method	Features	TP	FP	FN	TN	Acc	Prec	Rec	FPR	F1
	MICU (10,008)									
KDIGO	-	270	135	330	767	69.0%	0.66	0.45	0.14	0.53
LR	maxSCr [1.4]	417	79	183	823	82.5%	0.84	0.69	0.08	0.76
DT1	maxSCr [1.25]	496	122	104	780	84.9%	0.80	0.82	0.13	0.81
DT	maxSCr	493	122	107	780	84.7%	0.80	0.82	0.13	0.81
DT	KDIGO -I, R	290	105	310	797	72.3%	0.73	0.48	0.11	0.58
DT	KDIGO -I, R, max	486	108	114	794	85.2%	0.81	0.80	0.12	0.81
	CVICU (8,356)									
KDIGO	-	145	357	33	719	68.9%	0.28	0.81	0.33	0.42
LR	maxSCr [2.13]	81	29	97	1047	89.9%	0.73	0.45	0.02	0.56
DT1	maxSCr [1.75]	122	56	56	1020	91.0%	0.68	0.68	0.05	0.68
DT	maxSCr	93	38	85	1038	90.1%	0.70	0.52	0.03	0.60
DT	KDIGO -I, R	83	42	95	1034	89.0%	0.66	0.46	0.03	0.54
DT	KDIGO -I, R, max	93	38	85	1038	90.1%	0.70	0.52	0.03	0.60
	MISICU (8,151)									
KDIGO	-	244	105	269	605	69.4%	0.69	0.47	0.14	0.56
LR	maxSCr [1.45]	341	54	172	656	81.5%	0.86	0.66	0.07	0.75
DT1	maxSCr [1.25]	398	82	115	628	83.8%	0.82	0.77	0.11	0.80
DT	maxSCr	398	82	115	628	83.8%	0.82	0.77	0.11	0.80
DT	KDIGO -I, R	227	64	286	646	71.38%	0.78	0.44	0.09	0.56
DT	KDIGO -I, R, max	398	82	115	628	83.8%	0.82	0.77	0.11	0.80

Table 4.6: Results of different predictive models on ICU unit types (SICU, TSICU, CCU)

Method	Features	TP	FP	FN	TN	Acc	Prec	Rec	FPR	F1
	SICU (7,200)									
KDIGO	-	131	127	82	740	80.6%	0.50	0.51	0.15	0.55
LR	maxSCr [1.96]	104	32	109	835	86.9%	0.76	0.48	0.03	0.59
DT1	maxSCr [1.45]	157	75	56	792	87.8%	0.67	0.73	0.08	0.70
DT	maxSCr	110	35	103	832	87.2%	0.75	0.51	0.04	0.61
DT	KDIGO-I, R	76	35	137	832	84.0%	0.68	0.35	0.04	0.46
DT	KDIGO -I, R, max	119	43	94	824	87.3%	0.73	0.55	0.04	0.63
	TSICU (5,530)									
KDIGO	-	99	150	80	777	79.2%	0.39	0.55	0.16	0.46
LR	maxSCr [1.88]	95	29	84	898	89.7%	0.76	0.53	0.03	0.62
DT1	maxSCr [1.35]	143	109	36	818	86.8%	0.56	0.79	0.11	0.66
DT	maxSCr	88	25	91	902	89.5%	0.77	0.49	0.02	0.60
DT	KDIGO-I, R	63	33	116	894	86.5%	0.65	0.35	0.03	0.45
DT	KDIGO -I, R, max	88	25	91	902	89.5%	0.77	0.49	0.02	0.60
	CCU (5,522)									
KDIGO	-	260	150	143	552	73.4%	0.63	0.64	0.21	0.63
LR	maxSCr [1.85]	257	63	146	639	81.0%	0.80	0.63	0.08	0.71
DT1	maxSCr [1.45]	335	116	68	586	83.3%	0.74	0.83	0.16	0.78
DT	maxSCr	335	116	68	586	83.3%	0.74	0.83	0.16	0.78
DT	KDIGO-I, R	220	64	183	638	77.4%	0.77	0.54	0.09	0.64
DT	KDIGO -I, R, max	335	116	68	586	83.3%	0.74	0.83	0.16	0.78

Although this might be an indication that KDIGO definition started to be noticed in clinical practice, the machine learning model results paints a different picture.

Table 4.8 shows concordances for three types of DT models, where a separate DT

Table 4.7: Concordance between ICD AKI and KDIGO AKI (I1 implementation) over the years

Year	# Adm.	# ICD AKI	TP	FP	FN	TN	Acc (%)	F1
2008-10	14,207	4,497	2,330	1,897	2,167	7,813	71.3	0.53
2011-13	13,184	3,414	1,859	1,873	1,555	7,897	74.0	0.52
2014-16	12,268	3,115	1,806	1,697	1,309	7456	75.5	0.55
2017-19	7,840	2,150	1,402	944	748	4,746	<b>78.4</b>	<b>0.62</b>
Total	47,499	13,176	7,397	6,411	5,779	27,912	74.0	0.55

Table 4.8: Results of DT in different year ranges of MIMIC-IV database

KDIGO Period	Year	Adm.	AKI	maxSCr		KDIGO-I,R		KDIGO-I, R, maxSCr	
				Acc. (%)	F1	Acc. (%)	F1	Acc. (%)	F1
Before	2008-10	14,207	4,497	83.86	0.75	76.78	0.54	83.4	0.74
During	2011-13	13,184	3,414	86.55	0.74	80.03	0.51	86.65	0.73
After	2014-16	12,268	3,115	87.83	0.73	82.18	0.52	87.56	0.73
After	2017-19	7,840	2,150	<b>88.35</b>	<b>0.80</b>	81.55	0.62	<b>88.35</b>	<b>0.80</b>

model was trained on admissions from each year range. The table shows the change in concordance of three DT models over time. F1 concordance of DT using KDIGO variables increases from 0.54 during 2008-2010 to 0.62 during 2017-2019. This increase in concordance is almost identical to KDIGO definition. However, F1 concordance of DT using maxSCr variable increased from 0.75 during 2008-2010 to 0.80 during 2017-2019. Moreover, addition of KDIGO variables to maxSCr did not result in increased concordance of DT model in any of the four year ranges. If KDIGO definition had increased recognition in clinical practice, we would expect the KDIGO variables to be increasingly helpful in improving concordance compared to DT that only uses maxSCr variable.

#### 4.4 Conclusions

In this study, we analyzed the concordance between KDIGO definition of AKI and its coding clinical practice of a large hospital system in Boston covering the period from

2008 to 2019. Our results show that the concordance is relatively low and that it is possible to train a machine learning model with a significantly higher concordance than KDIGO definition. That machine learning formula and our analysis of false positives and negatives reveals that it is more likely that AKI is recognized in clinical practice when the SCr is higher than a threshold rather than when the large change in SCr occurs. We believe that our study will improve the understanding of the differences between KDIGO definition of AKI and its coding in the scientific community and clinical practice.

#### 4.5 Additional Results

We ran some additional experiments to our study on concordance between KDIGO definition on acute kidney injury (AKI) and its coding (Chapter 4). For example, we ran one experiment where we used the *maxSCr* during the first day of ICU as feature in different models and we have a population of 46,869 admissions with that. The results are given in Table 4.9. We find the threshold of LR and  $DT_1$  is 1.56 and 1.25, respectively, which is close to SOFA renal score 1.

We also used the SOFA renal as a predictive formula where a patient is denoted as AKI if the *maxSCr* value satisfies the SOFA renal condition. We consider four predictive formula for the four SOFA condition such as SOFA1, SOFA2, SOFA3, SOFA4, and the results are shown in Table for 4.9. The SOFA1 has 83% accuracy and 0.71 F1 score which is comparable to the LR where the trained model's threshold was 1.56. We also report the lab values and outcomes for the six ICU unit types in the last six columns of Table 4.10. We observe that 70.3% patients of the CVICU had vasopressor therapy. The CVICU unit also has a shorter ICU stay and smaller mortality rate than the other ICU units.

We used a list of medical codes for renal transplant patient detection (Schroeder et al., 2014; Tschida et al., 2013), as shown in Table 4.11.

Table 4.9: Results of different predictive models for hospital-based AKI diagnosis (Here, cohort: 46,869, FT: features, MSCr = maxSCr)

Method	FT	TP	FP	FN	TN	Acc	Prec	Rec	FPR	F1
LR	MSCr [1.56]	1,018	261	900	4,852	83.4%	0.79	0.53	0.05	0.63
$DT_1$	MSCr [1.25]	1,392	570	526	4,543	84.4%	0.70	0.72	0.11	0.71
DT	MSCr	1,255	441	663	4,672	84.3%	0.74	0.65	0.08	0.69
SOFA1	MSCr	1,517	793	401	4,320	83.0%	0.65	0.79	0.15	0.71
	$\geq 1.2$									
SOFA2	MSCr	681	119	1,237	4,994	80.7%	0.85	0.35	0.02	0.50
	$\geq 2$									
SOFA3	MSCr	180	20	1,738	5,093	75.0%	0.90	0.09	0.003	0.16
	$\geq 3.5$									
SOFA4	MSCr	54	9	1,864	5,104	73.3%	0.85	0.02	0.001	0.05
	$\geq 5$									

Table 4.10: Statistic of clinical variables and outcomes in different ICU units

Variables	Total (47,499)	MICU (10,008)	CVIU (8,356)	MSIU (8,151)	SIU (7,200)	TSICU (5,530)	CCU (5,522)
LOS	8.7	8.2	8.2	9.3	9.4	10.0	7.3
LOS ICU	3.4	3.2	2.8	3.0	3.7	3.9	3.2
# SCr	10.9	11.21	10.36	11.69	10.99	11.27	10.9
SOFA renal	0.58	0.76	0.48	0.68	0.43	0.41	0.84
Dialysis	1.5%	2.5%	1.1%	1.3%	1.3%	1.2%	1.9%
Vasopressor	33.1%	27.5%	70.3%	24.2%	21.8%	28.6%	31.3%
Mortality	8.2%	10.7%	2.2%	11.0%	8.3%	7.9%	9.9%

Table 4.11: ICD code description for renal transplant

<b>ICD-9 code</b>	<b>Code description</b>
55.6	Ulcerative enterocolitis.
55.61	Renal Autotransplantation
55.69	Ulcerative colitis, unspecified
99681	Complications of transplanted kidney
V420	Kidney replaced by transplant
00.91	Transplant from live related donor
00.92	Transplant from live non-related donor
00.93	Transplant from cadaver convert
<b>CPT code</b>	<b>Code description</b>
50380	Under Renal Transplantation Procedures
50365	Renal allotransplantation implantation of graft; with recipient nephrectomy.
50360	Renal allotransplantation implantation of graft; excluding donor and recipient nephrectomy.

# BIBLIOGRAPHY

- ad-hoc working group of ERBP: Fliser, D., Laville, M., Covic, A., Fouque, D., Vanholder, R., Juillard, L., and Van Biesen, W. (2012), “A European Renal Best Practice (ERBP) position statement on the Kidney Disease Improving Global Outcomes (KDIGO) clinical practice guidelines on acute kidney injury: part 1: definitions, conservative management and contrast-induced nephropathy,” *Nephrology Dialysis Transplantation*, 27, 4263–4272.
- Alsentzer, E., Murphy, J. R., Boag, W., Weng, W.-H., Jin, D., Naumann, T., and McDermott, M. (2019), “Publicly available clinical BERT embeddings,” *arXiv preprint arXiv:1904.03323*.
- Aronson, A. R. and Lang, F.-M. (2010), “An overview of MetaMap: historical perspective and recent advances,” *Journal of the American Medical Informatics Association*, 17, 229–236.
- Ayres, J., Flannick, J., Gehrke, J., and Yiu, T. (2002), “Sequential pattern mining using a bitmap representation,” in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 429–435.
- Bai, T. and Vucetic, S. (2019), “Improving medical code prediction from clinical text via incorporating online knowledge sources,” in *The World Wide Web Conference*, pp. 72–82.
- Bai, T., Chanda, A. K., Egleston, B. L., and Vucetic, S. (2017), “Joint learning of representations of medical concepts and words from EHR data,” in *2017 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2017, Kansas City, MO, USA, November 13-16, 2017*, pp. 764–769.
- Bai, T., Chanda, A. K., Egleston, B. L., and Vucetic, S. (2018), “EHR phenotyping via jointly embedding medical concepts and words into a unified vector space,” *BMC medical informatics and decision making*, 18, 123.
- Bai, T., Egleston, B. L., Bleicher, R., and Vucetic, S. (2019), “Medical concept representation learning from multi-source data,” in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 4897–4903, AAAI Press.

- Banerjee, I., Madhavan, S., Goldman, R. E., and Rubin, D. L. (2017), “Intelligent word embeddings of free-text radiology reports,” in *AMIA Annual Symposium Proceedings*, vol. 2017, p. 411, American Medical Informatics Association.
- Beam, A. L., Kompa, B., Fried, I., Palmer, N. P., Shi, X., Cai, T., and Kohane, I. S. (2018), “Clinical Concept Embeddings Learned from Massive Sources of Multimodal Medical Data,” *arXiv preprint arXiv:1804.01486*.
- Bellomo, R., Ronco, C., Kellum, J. A., Mehta, R. L., and Palevsky, P. (2004), “Acute renal failure—definition, outcome measures, animal models, fluid therapy and information technology needs: the Second International Consensus Conference of the Acute Dialysis Quality Initiative (ADQI) Group,” *Critical care*, 8, 1–9.
- Birkelo, B. C., Pannu, N., and Siew, E. D. (2022), “Overview of Diagnostic Criteria and Epidemiology of Acute Kidney Injury and Acute Kidney Disease in the Critically Ill Patient,” *Clinical Journal of the American Society of Nephrology*, 17, 717–735.
- Bıyık, M., Ataseven, H., Bıyık, Z., Asil, M., Çifçi, S., Sayın, S., Demir, A., and Tombul, H. Z. (2016), “KDIGO (Kidney Disease: Improving Global Outcomes) criteria as a predictor of hospital mortality in cirrhotic patients,” .
- Bleicher, R., Ruth, K., Sigurdson, E., Ross, E., Wong, Y., Patel, S., Boraas, M., Topham, N., and Egleston, B. (2012), “Preoperative Delays in the US Medicare Population With Breast Cancer,” *Journal of Clinical Oncology*, 30, 4485–4492.
- Bodenreider, O. (2004), “The unified medical language system (UMLS): integrating biomedical terminology,” *Nucleic acids research*, 32, D267–D270.
- Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016), “Enriching word vectors with subword information,” *arXiv preprint arXiv:1607.04606*.
- Bostock, M., Ogievetsky, V., and Heer, J. (2011), “D<sup>3</sup> data-driven documents,” *IEEE transactions on visualization and computer graphics*, 17, 2301–2309.
- Cai, X., Gao, J., Ngiam, K. Y., Ooi, B. C., Zhang, Y., and Yuan, X. (2018), “Medical Concept Embedding with Time-Aware Attention,” *arXiv preprint arXiv:1806.02873*.
- Cella, D., Riley, W., Stone, A., Rothrock, N., Reeve, B., Yount, S., Amtmann, D., Bode, R., Buysse, D., Choi, S., et al. (2010), “Initial adult health item banks and first wave testing of the patient-reported outcomes measurement information system (PROMIS™) network: 2005–2008,” *Journal of clinical epidemiology*, 63, 1179.
- Chanda, A. K. (2021), “Efficacy of BERT embeddings on predicting disaster from Twitter data,” *arXiv preprint arXiv:2108.10698*.

- Chanda, A. K., Bai, T., Yang, Z., and Vucetic, S. (2022), “Improving medical term embeddings using UMLS Metathesaurus,” *BMC Medical Informatics and Decision Making*, 22, 1–12.
- Chiu, B., Crichton, G., Korhonen, A., and Pyysalo, S. (2016), “How to train good word embeddings for biomedical NLP,” in *Proceedings of the 15th workshop on biomedical natural language processing*, pp. 166–174.
- Choi, E., Bahadori, M. T., Schuetz, A., Stewart, W. F., and Sun, J. (2016a), “Doctor ai: Predicting clinical events via recurrent neural networks,” in *Machine Learning for Healthcare Conference*, pp. 301–318.
- Choi, E., Schuetz, A., Stewart, W. F., and Sun, J. (2016b), “Medical concept representation learning from electronic health records and its application on heart failure prediction,” *arXiv preprint arXiv:1602.03686*.
- Choi, E., Bahadori, M. T., Searles, E., Coffey, C., Thompson, M., Bost, J., Tejedor-Sojo, J., and Sun, J. (2016c), “Multi-layer representation learning for medical concepts,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1495–1504, ACM.
- Choi, E., Schuetz, A., Stewart, W. F., and Sun, J. (2016d), “Using recurrent neural network models for early detection of heart failure onset,” *Journal of the American Medical Informatics Association*, 24, 361–370.
- Choi, Y., Chiu, C. Y.-I., and Sontag, D. (2016e), “Learning low-dimensional representations of medical concepts,” *AMIA Summits on Translational Science Proceedings*, 2016, 41.
- Christopoulou, F., Tran, T. T., Sahu, S. K., Miwa, M., and Ananiadou, S. (2020), “Adverse drug events and medication relation extraction in electronic health records with ensemble deep learning methods,” *Journal of the American Medical Informatics Association*, 27, 39–46.
- Coca, S. G., Bauling, P., Schiffner, T., Howard, C. S., Teitelbaum, I., and Parikh, C. R. (2007), “Contribution of acute kidney injury toward morbidity and mortality in burns: a contemporary analysis,” *American Journal of Kidney Diseases*, 49, 517–523.
- Coffman, A. and Wharton, N. (2007), “Clinical Natural Language Processing: Auto-Assigning ICD-9 Codes,” *Overview of the Computational Medicine Center’s*.
- Crammer, K., Dredze, M., Ganchev, K., Talukdar, P. P., and Carroll, S. (2007), “Automatic code assignment to medical text,” in *Proceedings of the workshop on bionlp 2007: Biological, translational, and clinical language processing*, pp. 129–136, Association for Computational Linguistics.

- De Vine, L., Zuccon, G., Koopman, B., Sitbon, L., and Bruza, P. (2014), “Medical semantic similarity with a neural language model,” in *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*, pp. 1819–1822, ACM.
- Deb, S. and Chanda, A. K. (2022), “Comparative analysis of contextual and context-free embeddings in disaster prediction from Twitter data,” *Machine Learning with Applications*, 7, 100253.
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018), “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*.
- Erdbruegger, U. and Okusa, M. D. (2018), “Etiology and diagnosis of prerenal disease and acute tubular necrosis in acute kidney injury in adults,” *Uptodate Waltham, MA*. <https://www.uptodate.com/contents/etiology-and-diagnosis-of-prerenal-disease-and-acute-tubular-necrosis-in-acute-kidney-injury-in-adults>. Last updated May.
- Fatehi, P. and Hsu, C.-y. (2017), “Evaluation of acute kidney injury among hospitalized adult patients,” *UpToDate, topic last updated Oct, 2*.
- Golbeck, J., Frago, G., Hartel, F., Hendler, J., Oberthaler, J., and Parsia, B. (2003), “The National Cancer Institute’s thesaurus and ontology,” *Journal of Web Semantics First Look 1\_1\_4*.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014), “Generative adversarial nets,” in *Advances in neural information processing systems*, pp. 2672–2680.
- Gotz, D., Wang, F., and Perer, A. (2014), “A methodology for interactive mining and visual analysis of clinical event patterns using electronic health record data,” *Journal of biomedical informatics*, 48, 148–159.
- Gotz, D., Zhang, J., Wang, W., Shrestha, J., and Borland, D. (2019), “Visual analysis of high-dimensional event sequence data via dynamic hierarchical aggregation,” *IEEE transactions on visualization and computer graphics*, 26, 440–450.
- Griffin, B. R., Liu, K. D., and Teixeira, J. P. (2020), “Critical care nephrology: core curriculum 2020,” *American Journal of Kidney Diseases*, 75, 435–452.
- Grover, A. and Leskovec, J. (2016), “node2vec: Scalable feature learning for networks,” in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 855–864.
- Guo, S., Jin, Z., Gotz, D., Du, F., Zha, H., and Cao, N. (2018), “Visual progression analysis of event sequence data,” *IEEE transactions on visualization and computer graphics*, 25, 417–426.

- Guthrie, G., Guthrie, B., Walker, H., James, M. T., Selby, N. M., Tonelli, M., and Bell, S. (2021), “Developing an AKI consensus definition for database research: findings from a scoping review and expert opinion using a Delphi process,” *American Journal of Kidney Diseases*.
- Ha, P., Zhang, S., Djuric, N., and Vucetic, S. (2020), “Improving Word Embeddings through Iterative Refinement of Word-and Character-level Models,” in *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 1204–1213.
- Hahn, U. and Oleynik, M. (2020), “Medical information extraction in the age of deep learning,” *Yearbook of Medical Informatics*, 29, 208–220.
- Halpern, Y., Horng, S., Choi, Y., and Sontag, D. (2016a), “Electronic medical record phenotyping using the anchor and learn framework,” *Journal of the American Medical Informatics Association*, 23, 731–740.
- Halpern, Y., Horng, S., Choi, Y., and Sontag, D. (2016b), “Electronic medical record phenotyping using the anchor and learn framework,” *Journal of the American Medical Informatics Association*, p. ocw011.
- Harris, Z. S. (1954), “Distributional structure,” *Word*, 10, 146–162.
- Hoste, E. A., Kellum, J. A., Selby, N. M., Zarbock, A., Palevsky, P. M., Bagshaw, S. M., Goldstein, S. L., Cerdá, J., and Chawla, L. S. (2018), “Global epidemiology and outcomes of acute kidney injury,” *Nature Reviews Nephrology*, 14, 607–625.
- Institute, E. (2018), “The Universal Medical Device Nomenclature System,” .
- J., J., Wong You Cheong, P., Nikolaidis, G., Khatri, V. S., Dogra, M., Dhakshinamoorthy Ganeshan, S., Goldfarb, J. G., et al. (2020), “American College of Radiology ACR Appropriateness Criteria for Renal Failure,” *Expert Panel on Urologic Imaging*.
- Jamal, S., Khan, M. Z., Kichloo, A., Edigin, E., Bailey, B., Aljadah, M., Hussaian, I., Rahman, A. U., Ahmad, M., and Kanjwal, K. (2020), “The effect of atrial fibrillation on inpatient outcomes of patients with acute pancreatitis: a two-year national inpatient sample database study,” *The Journal of Innovations in Cardiac Rhythm Management*, 11, 4338.
- Ji, S., Hölttä, M., and Marttinen, P. (2021), “Does the Magic of BERT Apply to Medical Code Assignment? A Quantitative Study,” *arXiv preprint arXiv:2103.06511*.
- Johnson, A., Bulgarelli, L., Pollard, T., Horng, S., Celi, L. A., and Mark, R. (2020), “MIMIC-IV (version 0.4),” *PhysioNet*.

- Johnson, A. E., Pollard, T. J., Shen, L., Li-wei, H. L., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L. A., and Mark, R. G. (2016), “MIMIC-III, a freely accessible critical care database,” *Scientific data*, 3, 160035.
- Kalyan, K. S. and Sangeetha, S. (2020), “SECNLP: A survey of embeddings in clinical natural language processing,” *Journal of biomedical informatics*, 101, 103323.
- Kate, R. J., Perez, R. M., Mazumdar, D., Pasupathy, K. S., and Nilakantan, V. (2016), “Prediction and detection models for acute kidney injury in hospitalized older adults,” *BMC medical informatics and decision making*, 16, 39.
- Keddis, M. T., Khanna, S., Noheria, A., Baddour, L. M., Pardi, D. S., and Qian, Q. (2012), “Clostridium difficile infection in patients with chronic kidney disease,” in *Mayo Clinic Proceedings*, vol. 87, pp. 1046–1053, Elsevier.
- Kellum, J. A., Lameire, N., Aspelin, P., Barsoum, R. S., Burdmann, E. A., Goldstein, S. L., Herzog, C. A., Joannidis, M., Kribben, A., Levey, A. S., et al. (2012), “Kidney disease: improving global outcomes (KDIGO) acute kidney injury work group. KDIGO clinical practice guideline for acute kidney injury,” *Kidney international supplements*, 2, 1–138.
- Kellum, J. A., Romagnani, P., Ashuntantang, G., Ronco, C., Zarbock, A., and Anders, H.-J. (2021), “Acute kidney injury,” *Nature reviews Disease primers*, 7, 1–17.
- Khattak, F. K., Jeblee, S., Pou-Prom, C., Abdalla, M., Meaney, C., and Rudzicz, F. (2019), “A survey of word embeddings for clinical text,” *Journal of Biomedical Informatics: X*, 4, 100057.
- Klabunde, C., Potosky, A., Legler, J., and Warren, J. (2000), “Development of a comorbidity index using physician claims data,” *Journal of clinical epidemiology*, 53, 1258–1267.
- Krause, J., Perer, A., and Stavropoulos, H. (2015a), “Supporting iterative cohort construction with visual temporal queries,” *IEEE transactions on visualization and computer graphics*, 22, 91–100.
- Krause, J., Razavian, N., Bertini, E., and Sontag, D. (2015b), “Visual Exploration of Temporal Data in Electronic Medical Records.” in *AMIA*.
- Krumholz, H., Wang, Y., Mattera, J., Wang, Y., Han, L., Ingber, M., Roman, S., and Normand, S. (2006), “An administrative claims model suitable for profiling hospital performance based on 30-day mortality rates among patients with heart failure,” *Circulation*, 113, 1693–1701.
- Kwon, B. C., Verma, J., and Perer, A. (2016), “Peekquence: Visual analytics for event sequence data,” in *ACM SIGKDD 2016 Workshop on Interactive Data Exploration and Analytics*, vol. 1.

- Lam, H., Bertini, E., Isenberg, P., Plaisant, C., and Carpendale, S. (2011), “Empirical studies in information visualization: Seven scenarios,” *IEEE transactions on visualization and computer graphics*, 18, 1520–1536.
- Lameire, N. H., Levin, A., Kellum, J. A., Cheung, M., Jadoul, M., Winkelmayer, W. C., Stevens, P. E., Caskey, F. J., Farmer, C. K., Fuentes, A. F., et al. (2021), “Harmonizing acute and chronic kidney disease definition and classification: report of a Kidney Disease: Improving Global Outcomes (KDIGO) Consensus Conference,” *Kidney international*, 100, 516–526.
- Levey, A. S., Eckardt, K.-U., Dorman, N. M., Christiansen, S. L., Hoorn, E. J., Ingelfinger, J. R., Inker, L. A., Levin, A., Mehrotra, R., Palevsky, P. M., et al. (2020), “Nomenclature for kidney function and disease: report of a Kidney Disease: Improving Global Outcomes (KDIGO) Consensus Conference,” *Kidney international*, 97, 1117–1129.
- Li, T., Carls, G. S., Panopalis, P., Wang, S., Gibson, T. B., and Goetzl, R. Z. (2009), “Long-term medical costs and resource utilization in systemic lupus erythematosus and lupus nephritis: a five-year analysis of a large Medicaid population,” *Arthritis Care & Research: Official Journal of the American College of Rheumatology*, 61, 755–763.
- Li, Y., Yao, L., Mao, C., Srivastava, A., Jiang, X., and Luo, Y. (2018), “Early prediction of acute kidney injury in critical care setting using clinical notes,” in *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 683–686, IEEE.
- Lipscomb, C. E. (2000), “Medical subject headings (MeSH),” *Bulletin of the Medical Library Association*, 88, 265.
- Maaten, L. v. d. and Hinton, G. (2008), “Visualizing data using t-SNE,” *Journal of machine learning research*, 9, 2579–2605.
- Maldonado, R., Goodwin, T. R., Skinner, M. A., and Harabagiu, S. M. (2017), “Deep learning meets biomedical ontologies: knowledge embeddings for epilepsy,” in *AMIA Annual Symposium Proceedings*, vol. 2017, p. 1233, American Medical Informatics Association.
- Maldonado, R., Yetisgen, M., and Harabagiu, S. M. (2019), “Adversarial Learning of Knowledge Embeddings for the Unified Medical Language System,” *AMIA Summits on Translational Science Proceedings*, 2019, 543.
- Mehta, R. L., Kellum, J. A., Shah, S. V., Molitoris, B. A., Ronco, C., Warnock, D. G., and Levin, A. (2007), “Acute Kidney Injury Network: report of an initiative to improve outcomes in acute kidney injury,” *Critical care*, 11, 1–8.
- Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013), “Efficient Estimation of Word Representations in Vector Space,” *CoRR*, abs/1301.3781.

- Miller, D. C., Saigal, C. S., Warren, J. L., Leventhal, M., Deapen, D., Banerjee, M., Lai, J., Hanley, J., and Litwin, M. S. (2009), “External validation of a claims-based algorithm for classifying kidney-cancer surgeries,” *BMC health services research*, 9, 92.
- Monroe, M., Lan, R., Lee, H., Plaisant, C., and Shneiderman, B. (2013), “Temporal event sequence simplification,” *IEEE transactions on visualization and computer graphics*, 19, 2227–2236.
- Mullenbach, J., Wiegrefe, S., Duke, J., Sun, J., and Eisenstein, J. (2018), “Explainable Prediction of Medical Codes from Clinical Text,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pp. 1101–1111.
- Mungall, C. J., McMurry, J. A., Köhler, S., Balhoff, J. P., Borromeo, C., Brush, M., Carbon, S., Conlin, T., Dunn, N., Engelstad, M., et al. (2017), “The Monarch Initiative: an integrative data and analytic platform connecting phenotypes to genotypes across species,” *Nucleic acids research*, 45, D712–D722.
- Munzner, T. (2009), “A nested model for visualization design and validation,” *IEEE transactions on visualization and computer graphics*, 15, 921–928.
- Nattinger, A., Laud, P., Bajorunaite, R., Sparapani, R., and Freeman, J. (2004), “An algorithm for the use of Medicare claims data to identify women with incident breast cancer,” *Health services research*, 39, 1733–1750.
- Nguyen, D., Luo, W., Venkatesh, S., and Phung, D. (2018), “Effective identification of similar patients through sequential matching over icd code embedding,” *Journal of medical systems*, 42, 94.
- Organization, W. H. (2013), “International Classification of Diseases,Ninth Revision, Clinical Modification (ICD-9-CM),” .
- Pakhomov, S., McInnes, B., Adam, T., Liu, Y., Pedersen, T., and Melton, G. B. (2010), “Semantic similarity and relatedness between clinical terms: an experimental study,” in *AMIA annual symposium proceedings*, vol. 2010, p. 572, American Medical Informatics Association.
- Pakhomov, S. V., Pedersen, T., McInnes, B., Melton, G. B., Ruggieri, A., and Chute, C. G. (2011), “Towards a framework for developing semantic relatedness reference standards,” *Journal of biomedical informatics*, 44, 251–265.
- Pakhomov, S. V., Finley, G., McEwan, R., Wang, Y., and Melton, G. B. (2016), “Corpus domain effects on distributional semantic modeling of medical terms,” *Bioinformatics*, 32, 3635–3644.

- Palevsky, P. M. (2020), "Definition and staging criteria of acute kidney injury in adults," *In: Motwani S, ed. UpToDate. Waltham, Mass.: UpToDate.*
- Paudel, R., Eberle, W., and Talbert, D. (2017), "Detection of anomalous activity in diabetic patients using graph-based approach," in *The Thirtieth International Flairs Conference.*
- Pedersen, T., Pakhomov, S. V., Patwardhan, S., and Chute, C. G. (2007), "Measures of semantic similarity and relatedness in the biomedical domain," *Journal of biomedical informatics*, 40, 288–299.
- Pennington, J., Socher, R., and Manning, C. (2014), "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543.
- Perer, A. and Gotz, D. (2013), "Data-driven exploration of care plans for patients," in *CHI'13 Extended Abstracts on Human Factors in Computing Systems*, pp. 439–444, ACL.
- Perotte, A. J., Wood, F., Elhadad, N., and Bartlett, N. (2011), "Hierarchically supervised latent Dirichlet allocation," in *Advances in Neural Information Processing Systems*, pp. 2609–2617.
- Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018), "Deep contextualized word representations. arXiv 2018," *arXiv preprint arXiv:1802.05365*, 12.
- Robinson, P. N. and Mundlos, S. (2010), "The human phenotype ontology," *Clinical genetics*, 77, 525–534.
- Rogers, J., Spina, N., Neese, A., Hess, R., Brodke, D., and Lex, A. (2019), "Composer—visual cohort analysis of patient outcomes," *Applied clinical informatics*, 10, 278–285.
- Schroeder, E. B., Goodrich, G. K., Newton, K. M., Schmittiel, J. A., and Raebel, M. A. (2014), "Implications of different laboratory-based incident diabetic kidney disease definitions on comparative effectiveness studies," *Journal of comparative effectiveness research*, 3, 359–369.
- Shneiderman, B. (1996), "The eyes have it: A task by data type taxonomy for information visualizations," in *Proceedings 1996 IEEE symposium on visual languages*, pp. 336–343, IEEE.
- Si, Y., Wang, J., Xu, H., and Roberts, K. (2019), "Enhancing clinical concept extraction with contextual embeddings," *Journal of the American Medical Informatics Association*, 26, 1297–1304.

- Siew, E. D., Ikizler, T. A., Matheny, M. E., Shi, Y., Schildcrout, J. S., Danciu, I., Dwyer, J. P., Srichai, M., Hung, A. M., Smith, J. P., et al. (2012), “Estimating baseline kidney function in hospitalized patients with impaired kidney function,” *Clinical Journal of the American Society of Nephrology*, 7, 712–719.
- Sun, M., Baron, J., Dighe, A., Szolovits, P., Wunderink, R. G., Isakova, T., and Luo, Y. (2019), “Early Prediction of Acute Kidney Injury in Critical Care Setting Using Clinical Notes and Structured Multivariate Physiological Measurements.” in *MedInfo*, pp. 368–372.
- Tao, C., Wongsuphasawat, K., Clark, K., Plaisant, C., Shneiderman, B., and Chute, C. G. (2012), “Towards event sequence representation, reasoning and visualization for EHR data,” in *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*, pp. 801–806.
- Tschida, S., Aslam, S., Khan, T. T., Sahli, B., Shrank, W. H., and Lal, L. S. (2013), “Managing specialty medication services through a specialty pharmacy program: the case of oral renal transplant immunosuppressant medications,” *Journal of Managed Care Pharmacy*, 19, 26–41.
- Turney, P. D. and Pantel, P. (2010), “From Frequency to Meaning: Vector Space Models of Semantics,” *J. Artif. Intell. Res.*, 37, 141–188.
- Vlasschaert, M. E., Bejaimal, S. A., Hackam, D. G., Quinn, R., Cuerden, M. S., Oliver, M. J., Iansavichus, A., Sultan, N., Mills, A., and Garg, A. X. (2011), “Validity of administrative database coding for kidney disease: a systematic review,” *American journal of kidney diseases*, 57, 29–43.
- Wang, T. D., Wongsuphasawat, K., Plaisant, C., and Shneiderman, B. (2011), “Extracting insights from electronic health records: case studies, a visual analytics process model, and design recommendations,” *Journal of medical systems*, 35, 1135–1152.
- Wang, Y., Liu, S., Afzal, N., Rastegar-Mojarad, M., Wang, L., Shen, F., Kingsbury, P., and Liu, H. (2018), “A comparison of word embeddings for the biomedical natural language processing,” *Journal of biomedical informatics*, 87, 12–20.
- Warren, J., Harlan, L., Fahey, A., Virnig, B., Freeman, J., Klabunde, C., Cooper, G., and Knopf, K. (2002), “Utility of the SEER-Medicare data to identify chemotherapy use,” *Medical care*, 40, IV–55.
- Warren, J. L., Mariotto, A., Melbert, D., Schrag, D., Doria-Rose, P., Penson, D., and Yabroff, K. R. (2016), “Sensitivity of Medicare claims to identify cancer recurrence in elderly colorectal and breast cancer patients,” *Medical care*, 54, e47.
- Wiersema, R., Jukarainen, S., Eck, R. J., Kaufmann, T., Koeze, J., Keus, F., Pettilä, V., van der Horst, I. C., and Vaara, S. T. (2020), “Different applications of the KDIGO

- criteria for AKI lead to different incidences in critically ill patients: a post hoc analysis from the prospective observational SICS-II study,” *Critical Care*, 24, 1–8.
- Winkelmayer, W., Schneeweiss, S., Mogun, H., Patrick, A., Avorn, J., and Solomon, D. (2005), “Identification of individuals with CKD from Medicare claims data: a validation study,” *American Journal of Kidney Diseases*, 46, 225–232.
- Wongsuphasawat, K., Guerra Gómez, J. A., Plaisant, C., Wang, T. D., Taieb-Maimon, M., and Shneiderman, B. (2011), “LifeFlow: visualizing an overview of event sequences,” in *Proceedings of the SIGCHI conference on human factors in computing systems*, pp. 1747–1756.
- Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. (2016), “Google’s neural machine translation system: Bridging the gap between human and machine translation,” *arXiv preprint arXiv:1609.08144*.
- Yan, Y., Birman-Deych, E., Radford, M., Nilasena, D., and Gage, B. (2005), “Comorbidity indices to predict mortality from medicare data: results from the national registry of atrial fibrillation,” *Medical care*, 43, 1073–1077.
- Zhang, Y., Chanana, K., and Dunne, C. (2018), “IDMVis: Temporal event sequence visualization for type 1 diabetes treatment decision support,” *IEEE transactions on visualization and computer graphics*, 25, 512–522.
- Zhang, Y., Chen, Q., Yang, Z., Lin, H., and Lu, Z. (2019), “BioWordVec, improving biomedical word embeddings with subword information and MeSH,” *Scientific data*, 6, 1–9.