Automating disease diagnosis and cause-of-death classification from medical narratives using event extraction and temporal ordering

by

Serena Jeblee

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Computer Science
University of Toronto

Abstract

Automating disease diagnosis and cause-of-death classification from medical narratives using event extraction and temporal ordering

Serena Jeblee
Doctor of Philosophy
Graduate Department of Computer Science
University of Toronto
2021

The digitization of health records has provided a wealth of data that can be used for machine learning tasks such as automated diagnosis of illness or cause of death. However, crucial contextual information such as time and duration have often been omitted from such models. Although the words and phrases describing time can be modeled with semantic representations such as word embeddings, these representations often fail to capture the temporal value of such time phrases. The goal of this work is to present models for identifying symptoms and time information in the text of different types of health documents (while explicitly modeling temporal value), determining the chronological order of the extracted symptoms, and classifying the documents according to diagnosis or cause-of-death.

Since the style and content of health text documents can be widely variable, these models are evaluated on three different types of health records: clinical notes (formal written health records), verbal autopsies (informal written records of deaths that occurred outside of health facilities), and transcripts of clinical conversations (verbal dialogues between medical staff, patients, and caregivers).

We conduct end-to-end disease classification using neural network models trained on free-text narratives. The system includes supervised learning models for event and time extraction, listwise temporal ordering of events, and classification. We find that classification models that include event timeline representations perform better than methods that use only the raw text of the narrative. Furthermore, we present several models and metrics for listwise temporal ordering, including the ability to model simultaneous events.

These models can be applied to different types of medical narratives, and can improve the performance of automated classifiers which can be used in real-world contexts to improve cause-of-death coding efficiency and clinical decision support.

Acknowledgements

I am very grateful for the generous funding for these projects from The U.S. National Institutes of Health, a Google Faculty Award, University of Toronto, and the Vector Institute.

Thanks to Aurélie Névéol for serving as the external examiner for my final thesis defence, and for your helpful feedback. Thanks also to Gerald Penn for serving on the final committee and Yuchong Zhang for serving as the meeting chair.

Thanks to all my fellow graduate students who helped keep me sane over the years, especially Akshay, Bai, Chloé, Gagandeep, Jeff, KP, Muuo, Nona, Saša, and Sean.

Many thanks to my supervisor Graeme Hirst for your relentlessly high standards, deadline reminders, subtle humor, and willingness to send me all over the world for conferences. Thank you also for your honesty, encouragement, support, wisdom, and advice. I still think that my American spellings are superior but I guess we can agree to disagree.

Thanks to Frank Rudzicz for being a constant fountain of ideas, connections, advice, and Star Trek references. I still don't know how you get so much done, but I appreciate all of your support.

Thanks to Mireille Gomes for advising me about the public health world and helping me keep my research in perspective, and for the many many Skype calls across different time zones.

Thanks to Prabhat Jha for your enthusiasm and support and of my work, and for fitting my committee meetings into your insane travel schedule.

Thanks to my undergraduate advisor David Yarowsky for introducing me to the field of natural language processing when I said that I wanted to study both linguistics and computer science.

Thanks to Chris Callison-Burch for giving me my first research project as an undergraduate and encouraging me to apply to graduate programs in NLP.

Thanks to Linkin Park, Hands Like Houses, PVRIS, Icon For Hire, The Score, Normandie, and Thessa for your amazing music that helped me get through my research and thesis writing.

Special thanks to the Weaver family for driving me through Johns Hopkins University when I was in high school and inspiring me to go there.

I couldn't have done this without the support of my amazing friends, including but not limited to: Becca, Jon, Natalie, Nora, Stephie, Pat (x2), Nick, Jenna, David, Margaret, Meg, and Fred, and my wonderful dance family: Holly, Kage, Kelsey, Brittany, Ameris, Elis, Cass, Laura, and Jennalee.

And last but certainly not least, thanks to my mom for listening to me rant about operating system woes even when you didn't understand it, and of course for your never-ending support and encouragement. Thanks to my dad for teaching me about computers when I was a kid and always telling me I could do anything I wanted if I worked hard at it. Thank you both for believing in me and feeding me for so many years. And thanks to my sister for teaching me to fight for my food, and for being a terrible influence on me in general. I hope to one day be half as fabulous as you.

Contents

List of Tables

List of Figures

List of Abbreviations

bi-LSTM	bi-directional long short term memory (LSTM) network
CHI	Calinski-Harabasz Index: an unsupervised clustering metric
CNN	convolutional neural network
CoD	cause of death
CSMFA	cause-specific mortality fraction accuracy
EPR	events per rank
GPR	gold pair recall
GPU	graphical processing unit
GRU	gated recurrent unit
LSTM	long short-term memory: a type of recurrent neural network (RNN) that can capture longer-range dependencies in the input sequence
MDS	Million Death Study: a verbal autopsy program in India
ML	machine learning
MSE	mean squared error
POA	pairwise ordering accuracy
RNN	recurrent neural network
Seq2Seq	sequence-to-sequence
Set2Seq	set-to-sequence
UMLS	Unified Medical Language System
VA	verbal autopsy

Chapter 1

Introduction

1.1 Overview

With recent advances in machine learning, natural language processing (NLP) can provide useful insights into patterns in unstructured medical text. Automated methods such as sequence tagging can be used to identify important phrases in the text, and machine learning models can be used to predict diagnosis codes or cause of death. Since temporal information such as time of occurrence and duration can provide important context for symptoms and other medically relevant events, we can automatically extract such information in order to provide a medical chronology that can be used by downstream applications.

However, the use of temporal information for automated disease diagnosis and classification of medical text has been limited by the difficulty of recognizing and using time information in medical narratives. Most existing work has used pairwise temporal relations between events, which can be sparse and inconsistent with one another. Pairwise relations also require extensive data annotation and $O(n^2)$ computation time.

In this work, the focus is instead on extracting a coherent listwise timeline from three different types of medical narratives (clinical notes, verbal autopsies, and clinical dialogues), and then learning a representation of the timeline that can be used as input to a machine learning classifier (such as a cause-of-death classifier). First, we automatically label events and times in the text, then perform listwise temporal ordering, and lastly use the resulting timeline representation to classify each document according to primary diagnosis or cause-of-death. Although there may be other types of medical narratives for which this problem is relevant, the datasets we use cover formal medical documents, informal narratives, and conversational data.

The system involves three stages: identifying events and time phrases in the text, putting the extracted events in chronological order (using an encoding of the event as an embedding vector), and performing disease classification. For each of the three stages, we evaluate a variety of machine learning models compared to human annotation. The primary focus is improving the accuracy of cause-of-death classification for verbal autopsy (VA) records. Additionally, we provide some explainability methods for the timeline-based classifier in order to help users understand which parts of the timeline had the most influence on the classification. We find that adding time information and chronology to the disease classification model improves the classification accuracy and allows us to provide a more coherent summary of the document to the user. Explicit temporal ordering can capture contextual time information about

events in a way that context-only text representation methods cannot. The listwise ordering models that we explore for temporal ordering could be applied to other domains and other types of ordering problems.

Automated disease coding can also provide insights to public health professionals about the true distribution of diseases and inform health care reform. However, current models provide only a CoD code prediction with very little explanation. We aim to make automated prediction more valuable to clinicians by providing a temporal chronology in addition to the diagnosis. This system can be used as a first pass to reduce the cost and time burden on clinicians for coding verbal autopsy records.

In addition, we present a timeline extraction model that can assist in clinical decision making by providing a chronological overview of important events in a patient's history. In contrast to previous work, we use a globally ordered timeline that can include grouping, and we demonstrate that including this listwise timeline improves disease classification accuracy. Integrating such models into clinical workflows will reduce the amount of time clinicians must spend creating and reviewing documentation and give them more time to focus on their patients.

1.2 Background

In the medical field, the advent of electronic health records has sparked an interest in using artificial intelligence to gain insights from the wealth of available data. Despite the use of structured databases, many health records necessarily contain unstructured text such as clinical notes and discharge summaries, which contain vital information about a patient's history, including symptoms, diagnoses, procedures, and medications. These events typically include context that indicates the order and duration of the events where necessary. For instance, the length and intensity of a fever is important for distinguishing diseases. In clinical notes, procedures and treatments are almost always accompanied by a date, either exact (such as "01-01-19") or relative (such as "yesterday"). This information is important for anyone reviewing the record to get a sense of the progression of the patient's symptoms and treatment. The temporal order of symptoms can be crucial to distinguishing different diseases. For example, Larsen et al. (2020) found that the order of symptoms is important in distinguishing COVID-19 from other respiratory diseases.

In fact, in the training for the Million Death Study (the verbal autopsy collection program which provided the data used in this work, to be described in Section 1.2.2 below), physicians are instructed to highlight key phrases in the text and order them chronologically as part of the cause-of-death (CoD) coding process. For our automated system, we want to focus on the same kind of information that a human expert would use to make a diagnosis.

Many of the previous machine learning methods for health either rely heavily on structured data or consider only individual words from the text, using features such as word counts and part-of-speech tags. More recent models can process sequences of text, typically within a fixed number of words, called a context window. However, especially in medical data, contextual information such as the duration of events and the temporal relationships between events is important, and is not inherently captured by window context methods.

The order and duration are especially important in the context of treatments and other medical events, which may have causal relationships. For instance, VAs often contain information about the subject's medical history which might or might not be relevant to the actual CoD depending on the type

of event and how long ago it happened. Traditional feature models, such as bag-of-words models (i.e. single word features that do not capture word sequence) treat all parts of the narrative equally, ignoring these important temporal cues.

This temporal information can be critical to making the correct diagnosis for a variety of reasons. Firstly, it can help the classification model to ignore irrelevant events. For instance, if a verbal autopsy narrative mentions a car accident, the classifier might predict "traffic accident" as the CoD, since there are many training examples of traffic accident-related deaths with exactly such phrases. However, if the car accident occurred 5 years prior to death, it is likely not the immediate cause. Knowing the time of occurrence for such events can help the classification model to focus on the most relevant events.

In addition, the duration of symptoms can be an important factor in diagnosis. For example, a diagnosis of tuberculosis includes "high fever of long duration" along with chronic cough, and while cough and fever are also symptoms of influenza, influenza is specified by "high fever of short duration" (SRS Collaborators of the RGI-CGHR, 2014). Although there are other distinguishing symptoms between the two diseases, the duration of the fever can help determine whether it is an acute infection like influenza, or a chronic disease like tuberculosis.

The extracted timeline of events with temporal information can also be useful for providing interpretability to medical staff. Health records can be very long and the text is often not organized in a standard way. A timeline can provide a quick overview of the patient's main symptoms and medical history. For CoD classification, if an automated system is used for coding, a physician reviewing the output can look at the timeline to get a better sense of why the classifier predicted a certain CoD.

1.2.1 Clinical notes

Clinical notes are digital or handwritten notes about a patient, typically written by the physician or medical staff. When a patient visits a medical provider, an electronic health record (EHR) is created or updated with demographic information about the patient, non-narrative data such as lab test results, and free-text clinical notes. These notes contain critical information about the patient's visit and medical history. This can include admission/discharge information, medical procedures, prescriptions, diagnostics, and diagnosis codes (Spasic and Nenadic, 2020).

Clinical notes are typically written in a very concise and efficient format, which often results in ungrammatical text. Clinical text also contains many domain-specific terms and acronyms. Since many countries and regions lack a standard for EHR formatting, the style and structure of clinical notes can vary widely between facilities, as well as between individual clinicians. This can result in variation in abbreviations, acronyms, and terminology, even within the same note, as different sections may have different authors. In addition, the notes may contain typos and misspellings.

Clinicians typically have limited time to review a patient's record and may have difficulty finding the most relevant information, especially in records that contain very long clinical notes. This can result in delays in care, mistakes, and extra hours of work for clinicians.

NLP can be used as an analytical tool to extract and synthesize important information from clinical notes in order to save time for clinicians and assist them in providing care. This can include clinical term extraction, relation extraction, text summarization, text generation, word sense disambiguation, spelling correction, coreference resolution, diagnosis classification, and prediction of survival and other major medical events (Spasic and Nenadic, 2020). Machine learning methods can be used for automated classification using both structured variables and free-text notes for tasks such as automated coding,

readmission or mortality prediction, etc.

Although machine learning algorithms can be useful in clinical settings, the annotation required for supervised learning models can be time-consuming, and often results in models being trained on very small datasets because only a small portion of the available data is annotated (Spasic and Nenadic, 2020). However, semi-supervised models and automated data labeling can be used to leverage larger unlabeled datasets along with smaller labeled datasets.

In addition, the privacy requirements of healthcare data often prevent the use of lower-cost solutions like Amazon Mechanical Turk [1], or the annotation might require medical expertise that crowd-sourced annotators cannot offer. The availability of clinical data continues to be a roadblock for machine learning research due to privacy restrictions on making data publicly available. A notable exception is the MIMIC-III dataset (Pollard and Johnson, 2016), which is currently the only publicly available dataset of clinical notes.

In this work, we perform clinical entity recognition, time expression recognition, temporal ordering, and text classification on annotated clinical notes.

1.2.2 Verbal autopsy reports

Some material from previously published work (Jeblee et al., 2019a)

Throughout the world, approximately two-thirds of the deaths that occur annually have no medically certified cause of death (CoD), particularly in low and middle-income countries (Jha, 2014). This presents a challenge for public health funding and research because it is important to know the leading causes of death in order to work to prevent them. In addition, the distribution of CoDs that occur in medical facilities and are well-documented is different from the distribution of CoDs that are undocumented. Since these unknown deaths typically occur in more rural areas with less access to healthcare, the causes are often different.

Accurate CoD reporting is important not only for planning healthcare resources and education, but also for identifying new disease outbreaks (Gomes et al., 2017). For countries without the infrastructure to gather complete death statistics, they could be missing vital information about preventable deaths and disease outbreaks.

In order to get an estimate of these unknown CoDs, many countries employ verbal autopsy (VA) surveys, which are a low-cost option for estimating the CoD distribution in countries where such statistics are missing, and can inform public health funding and research.

A trained but non-medical surveyor visits homes where deaths have occurred and interviews the family members (and sometimes neighbors) about the circumstances surrounding the person's death. The surveyor then writes up a report which includes demographic information about the deceased and the respondent, answers to yes/no and multiple choice questions, and a free-text narrative (typically about half a page). The survey questions cover medical history, medications, known conditions, and key symptoms (Gomes et al., 2017). These reports are later coded for CoD by trained physicians, who review the demographic information, questionnaire answers, and free-text narrative, and assign a disease code corresponding to what they think is the most likely cause of death.

We refer to the demographic and questionnaire data as "non-narrative data", in the sense that they are not free-text like the narrative. The values may be represented as text, but they are a part of

[1] https://www.mturk.com/

key-value pairs, usually with limited value options. In the case of clinical notes the non-narrative data could also contain numerical data such as test results.

The WHO 2016 Verbal Autopsy Instrument (Nichols et al., 2018) is one of the most commonly used standardized VA forms, although there are many VA programs with different forms and data formats. Also, VA surveys are conducted in a wide variety of languages.

CoD surveys can also measure risk factors for common diseases, which can aid in prevention (Jha, 2014). For example, a retrospective VA study in China found that smokers had a higher rate of tobacco-related premature death (Liu et al., 1998). The Million Death Study, a VA program in India, found higher rates of malaria, traffic deaths, and snakebite deaths than what WHO had previously estimated in certain regions in India (Gomes et al., 2017). Identifying such associations that contribute to a large number of deaths allows governments to adapt public health programs to address the primary risk factors.

Since coding VA reports for CoD is costly and time-consuming, many VA programs are looking to implement at least partially automated coding. This is where natural language processing and machine learning can help – by using both the questionnaire data and the free-text narrative to estimate individual and populations-level CoDs for large datasets. This will reduce the burden on coding physicians, and allow larger volumes of data to be processed quickly and at a lower cost.

Although there have been criticisms of physician-coded VAs (Lozano et al., 2011), there is no other gold standard for VA coding that we can evaluate against, since for most VAs have no clinically confirmed CoD. Records of hospital deaths cannot be considered a gold standard for non-hospital deaths because of the differences in the distribution of CoDs, as well as the differences between the characteristics of patients who receive care in hospitals and those who die at home without medical attention (such as education level, access to hospital care, types of pathogens, etc.) (Aleksandrowicz et al., 2014; Ram et al., 2016; Berkley et al., 2005). For this reason, physician-coded VAs are often used for training and testing automated CoD coding methods.

Although several automated VA coding systems are available, some methods (such as InterVA-4 (Byass et al., 2012)) require a specific data format and variables which can make it difficult to apply to datasets collected with different protocols. Most methods use only the non-textual data, or treat the narrative as a bag-of-words, which ignores linguistic context. In addition, there have been conflicting reports about the accuracy of these systems on different datasets.

For example, for the InsilicoVA system (McCormick et al., 2016), Flaxman et al. (2018) reported much lower scores than the ones originally reported by McCormick et al. (2016). In response, Li et al. (2020) claimed that the experiment parameters were not sufficiently described, that code was not provided by Flaxman et al. (2018), and that the data format that was used favored InterVA-4 and the Tariff 2.0 method (Serina et al., 2015). Li et al. attempted to replicate Flaxman et al.'s experiments but again got different results. In response, Flaxman et al. (2020) claimed that their code was indeed available online, although the URL was not provided in the original paper. They claimed that the replication study was flawed and only attempted to reproduce one of the nine configurations they tested.

However, both parties agreed that reproducibility is highly important for automated public health systems, and that data selection and pre-processing can greatly affect the performance of such methods. Additonally, Li et al. (2020) emphasized that we cannot assume that methods developed on reports of hospital-based deaths will generalize to deaths outside of health facilities or from other regions of the world.

To this point, we aim to provide a classification method that can be applied to any VA dataset that includes free-text narratives, and we provide open-source code so that our efforts can be reproduced by others and applied to new datasets.

Chapter 2 will discuss verbal autopsy datasets in more detail, and Chapter 5 will review automated CoD coding methods.

1.2.3 Event, times, and temporal relations

In order to extract important information from medical text documents, we focus on events, time expressions, and temporal relations.

Events in medical narratives

We can describe an **event** as any sequence of words in the text that describes a relevant action, state, symptom, medication, or medical procedure. While events in the general domain are typically verbs, such as "said" or "consulted", in medical documents relevant events can also be nouns, such as "fever" or "aspirin". Styler et al. (2014) argued that any entities that can be classified as disorders, drugs, procedures, or signs or symptoms should be considered events in the medical domain. Additionally, an event can be any happening that is medically relevant, such as a car accident or the patient's occupation (as this may be relevant for their risk factors).

Event extraction is a similar task to named entity recognition (NER), which aims to identify named entities in text (usually nouns), such as proper names, places, titles, etc. Although events can be named entities, event extraction is somewhat more broad than traditional NER, because events can include a wide variety of types of entities and parts of speech.

Time expressions in medical narratives

A **time expression**, or **timex**, is a linguistic expression that refers to a time or date, temporal interval or other temporal region (Derczynski, 2017). This can include absolute dates and times (such as "March 5" or "2010") as well as relative time expressions (such as "yesterday" or "3 months ago"). A timex can also describe a duration (such as "3 weeks") or frequency (such as "twice a day").

Time expressions can indicate the exact location of a specific event in time, or they might describe the event's start time, end time, or duration (Leeuwenberg and Moens, 2019). In addition, these time phrases might describe absolute points in time, or they may be relative to another time phrase or reference point, such as a hospital admission date or date of death. Many documents have a document creation time (DCT) which can provide an anchor point for relative time expressions. For news articles, this is typically the date the article was written or published, and for medical documents it is often the admission or discharge date. In the case of verbal autopsies, the date of death is often the reference point for relative timexes in the narrative.

Sometimes time information is implied by context or it is made explicit in linguistic aspects of the event phrase, such as tense or aspect. **Tense** specifies when an event occurred relative to the time of speaking or some other reference point, such as in the past, present, or future, whereas **aspect** specifies whether an event has been completed or not (Binnick, 2012). For example, "patient has been throwing up for 2 days" implies that vomiting started 2 days ago but has not yet stopped. Aspectual information can be helpful in determining the endpoints of a time interval.

Temporal relations and temporal ordering

A **temporal relation** describes either the relationship between a timex and event or the temporal relationship between two events. A **temporal relation scheme** provides a model for describing relations between events and times, as well as the relations between events, in a structured and consistent way. This includes a set of temporal relation types, but can also include rules for temporal closure and/or annotation guidelines. We will discuss methods for automatically identifying temporal relations in Chapter 4.

Since pairs of events or events and times that have no defined relation are usually not annotated, creating a complete temporal graph requires inferring the complete set of relations from the set of given annotated relations, using transitivity rules. The resulting complete graph is called the **temporal closure.**

Depending on the granularity of the model, temporal relations might specify only which event comes first (or if the events are simultaneous), or they might also include information about if and how those events overlap. This typically depends on whether events are represented as points in time or as intervals. Although a point-based model might be simpler to use, and is appropriate for some situations (for instance, if most timexes in the document are single dates or specific times), interval-based models can represent more-complex relations such as event overlap. While some events are instantaneous, most events take some amount of time (Allen and Ferguson, 1994). In the medical domain, events may be short in duration (such as a test or procedure), or long-term symptoms or conditions that last for weeks, months, or years. An interval-based model is better suited to representing the duration and overlap of events.

Temporal relations may be binary (related or not) or simple (BEFORE, AFTER, OVERLAP), or they may capture more complex relationships such as partial overlap or adjacency.

An alternative way of conceptualizing the relationship between events and timexes is **narrative containers**, which are intervals which can encompass events or act as the timeframe for those events (Pustejovsky and Stubbs, 2011).

Although more complex temporal relation schemes allow for greater specificity, these more expressive models are also less computationally tractable (Zhou and Hripcsak, 2007). Therefore it is important to find a balance between simplicity and expressiveness, and also to choose a model that is appropriate to the dataset. Especially in the medical domain, the temporal information available may be vague and incomplete, and therefore we may need to use a more coarse-grained temporal model that allows for missing or partially specified temporal intervals.

Allen's interval relations and interval temporal logic

Most current temporal annotation schemes are based on Allen's theories of temporal logic (Allen, 1984; Allen and Ferguson, 1994), which were developed to support temporal reasoning about actions and events for the purposes of planning and prediction. Allen's temporal relations are an exhaustive set of relations that model events and actions as intervals that possibly overlap with each other. There are 13 relations: BEFORE, MEETS, OVERLAPS, STARTS, DURING, FINISHES, and their inverses, as well as EQUALS. This typically requires knowing both the start and end points for each event.

Temporal reasoning can also be performed with **semi-interval logic**. Freksa (1992) introduced the idea of conceptual neighbors of Allen relations — temporal relations that can be directly transformed

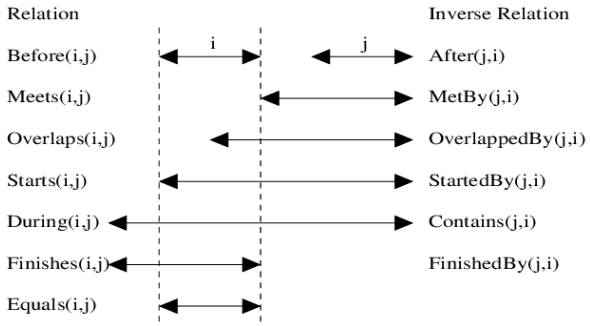

Figure 1.1: Allen's interval temporal relations (Allen and Ferguson, 1994).

into each other by changing the length of the interval (such as BEFORE and MEETS). This schema allows for representing coarser temporal knowledge where complete interval information is not available.

STAG

Setzer (2001) introduced the Sheffield Temporal Annotation Guidelines (STAG) for annotating events and time information in newswire text. STAG defines 5 temporal relations: BEFORE (and its inverse, AFTER), INCLUDES (and its inverse, INCLUDED), and SIMULTANEOUS. Temporal closure can be computed using inference rules; however, due to the vague nature of the SIMULTANEOUS and INCLUDES relation, which do not specify the nature of the overlap, the closure algorithm could result in erroneous relations. For example, a SIMULTANEOUS could mean a BEGINS, DURING, FINISHES, OVERLAPS or EQUALS relation in the Allen set. This fuzziness about the start and end points of the intervals can cause incorrect inference about transitive relations (Verhagen, 2005).

Container relations

Since medical narratives often include clusters of symptoms that occurred at roughly the same time, it may be useful to group these events together as a single chunk of time. A **narrative container**, which was first introduced by Pustejovsky and Stubbs (2011), "is the default interval containing the events being discussed in the text, when no explicit temporal anchor is given."

Essentially, a narrative container is a timex that can be interpreted by the reader as the default timeframe for events that are not otherwise temporally specified. This is often a timex other than the DCT that serves a similar function. For example, a clinical note will often list a number of events that happened around the same time. An admission date or DCT often serves as a reference point if there is not a specific timex mentioned in that portion of the text. For instance, in the phrase "on the evening of January 3rd she was admitted with a fever and treated with antibiotics", the narrative container "January 3rd" contains the events "admitted", "fever", and "treated". Narrative container relations are well suited to the medical domain because symptoms often co-occur with other symptoms or treatments, and because they can require less specific time information.

Narrative containers are often annotated as the CONTAINS or IS-CONTAINED relation. Although these relations are less specific in terms of interval endpoints, the annotation scheme is much simpler than the full set of Allen's relations.

1.2.4 Text representation for machine learning models

Text representation methods

Throughout the history of machine learning, various methods have been used to represent textual data, since mathematical models require numerical input. While early methods used numerical and class-based features, such as part-of-speech (POS) tags and other grammatical and lexical features, more recent methods have focused on representing the meaning of words and sentences.

Word embeddings are distributed vector representations of words, trained from context so that words with similar meanings will have vectors that are closer to each other in embedding space. While word indexing methods treat every word in the vocabulary as a separate item, word embedding representations allow the model to capture semantic similarities between words, which helps the model to generalize to new data. Typically, a document is represented as a matrix where each row is an embedding vector corresponding to a word in the input.

Word embeddings have become one of the most popular text representation methods for natural language processing, due to the fact that they are easy to train and reduce the need for manual feature engineering. In the next section, we provide a brief overview of current embedding methods, with a focus on the clinical domain.

Previous work on clinical text representation

Material in this section is based on previously published work (Khattak et al., 2019).

Modern text embedding models include word-level methods such as word2vec (Mikolov et al., 2013), fasttext (Bojanowski et al., 2017), and GloVe (Pennington et al., 2014), and sentence-level embeddings such as LASER (Schwenk and Douze, 2017). Khattak et al. (2019) provide an overview of current word embedding methods for clinical data.

Context-based embedding methods such as word2vec are typically learned by training a neural network to predict a focus word given its context (continuous bag of words (CBOW)) or the reverse (skip-gram). Two key parameters for training word2vec embeddings are 1) the number of the embedding dimensions (typically between 50 and 500, tuned experimentally), and 2) the length of the context window (i.e., how many words before and after the target word should be used as context for training the word embeddings, usually 5 or 10).

More recently, the best results have been obtained by using embeddings learned by large contextual language models such as ELMo (Peters et al., 2018), which uses an RNN model, and BERT (Devlin et al., 2018), which uses a transformer model (Vaswani et al., 2017). These representations are created by training a neural language model on a task such as next sentence prediction. Although training these methods from scratch requires large volumes of data, many pre-trained versions are available which can be fine-tuned on smaller amounts of data. Fine-tuning such models has been shown to generate a representation that works well for many downstream tasks.

However, such models trained on large amounts of general text might not accurately represent the meaning of clinical terms, especially if those terms are rare. There are two main approaches to including clinical information in word embeddings: training on a large corpus that includes medical terms, which relies on context to learn the semantics of such terms, and using structured resources to inject clinical knowledge into embeddings during or after training.

For the first approach, there are several pre-trained embedding models for the medical domain, trained on PubMed [2] articles or MIMIC-III (Pollard and Johnson, 2016), a publicly available database of clinical notes. These models include the PubMed ELMo model [3], BioBERT (Lee et al., 2019a), and Clinical BERT (Alsentzer et al., 2019).

For the second approach, there have been several different methods for creating clinical concept embeddings. In order to evaluate these embeddings, there are several available datasets of clinical concepts with similarity scores that can be used as a reference for evaluation pairs of terms in a trained embedding model.

The University of Minnesota Medical Residents Similarity and Relatedness Sets (UMNSRS) (Pakhomov et al., 2010) contain pairs of clinical concepts from the Unified Medical Language System (UMLS) that were manually rated by medical experts. The similarity set contains 566 pairs of concepts and the relatedness set contains 588 pairs. "Similarity" here means that the concepts are more or less synonymous (such as "heart attack" and "myocardial infarction"), whereas "relatedness" refers to concepts that are relevant to each other but do not describe the same thing, such as "heart disease" and "blood pressure". The Mayo Medical Coders Set (MayoSRS and MiniMaySRS) datasets (Pedersen et al., 2007) contains 101 clinical term pairs with relatedness scores from 3 physicians and 9 medical coders. With regards to the difference between similarity and relatedness judgments, the UMNRSRS dataset may be preferable because it explicitly encodes this distinction.

Starting with typical word embeddings, clinical knowledge can be used to refine the representations. This information can be injected at the training stage; for example, Boag and Kané (2017) injected domain knowledge into word embeddings trained on the MIMIC-III corpus (Johnson et al., 2016) by adding UMLS features to the embedding training. Using an extension of word2vec that allows training on arbitrary contexts, Levy and Goldberg (2014) trained a model using a context representation that included both the surrounding words and the identifiers of UMLS terms that match the given word in the UMLS database. This trained vector model is called Augmenting Word Embeddings with a Clinical Metathesaurus (AWE-CM) [4]. Although these embeddings had higher correlations with physician similarity judgments (.508 vs. .495) on MiniMaySRS-doctors, the default word2vec embeddings trained on MIMIC-III had higher correlation with medical coders (MiniMaySRS-coders and MayoSRS). Similarly, Patel et al. (2017) created embeddings for words and ICD-10 codes for automated medical coding review, which provided a 1% improvement of F_1 scores on an automated coding review task, where the model predicts whether the medical billing code should be accepted or sent for re-coding.

Embeddings can also be learned directly for medical terms. Choi et al. (2016) trained embeddings for clinical concepts from UMLS, as well as ICD-9 codes. They used a dataset of medical journal abstracts from OHSUMED[5], as well as a private dataset of medical claims.

Using a co-occurrence matrix of UMLS identifiers in clinical notes, the *cui2vec* model from Beam et al. (2018) trained GloVe and word2vec embeddings using the method from Finlayson et al. (2014). The authors also developed a set of benchmarks to measure embedding similarity for co-morbidity relationships, causative relationships, drug-condition relationships, UMLS semantic types, and human similarity judgments from UMNSRS. *cui2vec* performed better than the embeddings from (Choi et al.,

[2] https://www.ncbi.nlm.nih.gov/pubmed/
[3] Available at https://allennlp.org/elmo
[4] https://github.com/wboag/awecm
[5] http://davis.wpi.edu/xmdv/datasets/ohsumed

2016) on almost all benchmarks. The embeddings are available online [6].

In order to use contextual information, Mencia et al. (2016) used the all-in-text method (Nam et al., 2016), which creates label embeddings that are close in vector space to the embeddings of documents that have those labels. This is similar to the method from Patel et al. (2017), described above, but here the embeddings are trained from scratch. These embeddings showed higher correlation with the UMNSRS datasets than other pre-trained medical embeddings such as the PubMed vectors. The dataset is available online [7].

Zhu et al. (2018) presented a framework for clinical concept extraction using contextual word embeddings. They trained ELMo word embeddings on SNOMED-CT, in international clinical terminology resource (Rogers and Bodenreider, 2008), in addition to discharge summaries and radiology reports from the MIMIC-III dataset.

These works consistently showed that training embeddings on medical text produced better downstream performance than training on larger, more general corpora such as Google News (Mencia et al., 2016; Zhao et al., 2018), and that pre-trained models can be fine-tuned for medical-specific tasks.

However, many clinical concept embedding models are limited in the number of concepts that they can represent, and the concept embeddings are often in a different embedding space than the general word embeddings. This makes them more suited to representing individual medical terms rather than representing an entire document. For this reason, we opt to use language model-based embedding methods trained on biomedical data, such as ELMo.

Non-English text representation

Although embedding models have been trained for a variety a languages, the vast majority of clinical text representation work has been done on English. However, many verbal autopsies are conducted in other languages. From the Million Death Study we have a small dataset of narratives in Hindi that we would like to be able to use without having to translate them to English.

Currently the main limitation of working with languages other than English is the lack of resources. For English, there are several available datasets of medical documents that can be used for learning language models (such as MIMIC-III and THYME. Additionally, state-of-the-art models such as BERT (Devlin et al., 2018) and ELMo (Peters et al., 2018) have provided pre-trained models in English and a few other languages (such as French and Japanese), but are not currently available in Hindi. Such models would need to be trained from scratch for Hindi.

There are several embedding models available in Hindi, such as LASER (Schwenk and Douze, 2017) and fastText (Bojanowski et al., 2017). Our embedding and classification models for Hindi will be discussed in Section 5.4.2.

1.3 Research Questions and Hypotheses

From a clinical standpoint, time and sequence information is important for clinical decision-making and analysis (Sun et al., 2013c). Therefore we hypothesize that such temporal information should also be useful for machine learning models that operate on medical text data. We also expect different document styles to affect classification. In particular we set out to answer the following questions:

[6] http://cui2vec.dbmi.hms.harvard.edu/
[7] http://www.ke.tu-darmstadt.de/resources/medsim

Primary research question:

- **(Q1) Does information about temporal order and duration of medically relevant events improve medical text classification tasks such as cause-of-death (CoD) or disease classification?** In order to address this question, in Chapter 5 we use temporal information in machine learning classifiers for each of the 3 datasets.

Secondary research questions:

- **(Q2) How do differences in the style of medical narratives affect NLP models? (E.g. clinical notes vs. verbal autopsies vs. clinical dialogues)** Can we leverage text data of different types to improve the models? In order to characterize differences in style, formality, and temporal information in medical narratives, we analyze three different types of documents: formal health records (clinical notes), informal death records (verbal autopsies), and clinical dialogues (transcripts of audio conversations between physicians and patients). In each chapter we will examine the results on these three different types of documents, and how their inherent differences affect model performance. An overview of the details of the datasets is provided in Chapter 2. Also in Chapter 5 we experiment with pre-training on a different kind of medical narrative data to boost performance for small datasets.

- **(Q3) What kind of temporal information about events is the most useful for disease or CoD classification models (i.e. time of occurrence, duration, order, co-occurrence)?** We will examine whether focusing on important event and symptom phrases and excluding irrelevant information can improve classification. In addition, does adding temporal information to the model improve classification? And is the relative order of events enough, or does the model also benefit from absolute time and date information? Does duration information improve the results, as we expect it should? To answer these questions we experiment with various temporal features in Chapter 4.

With our focus on answering these three questions for our three data types, we introduce an end-to-end NLP system that includes event and time extraction, a listwise temporal ordering model that allows for simultaneous events, and an explainable timeline-based disease classification model. These models allow us to extract important time information from medical narratives and use it for more accurate disease and cause-of-death diagnosis.

We present the following research contributions:

- Since we want models that can be applied to raw, unannotated text, we must first identify events and time phrases in the text. In Chapter 3 we examine the performance of existing event and time taggers on all 3 datasets, and compare them to models trained specifically for each data type.

- In order to provide useful temporal information to classifiers, we investigate several different models for ordering the extracted events according to time in Chapter 4. Although most previous work has mainly used pairwise classification, we focus on a global listwise ordering, which is much simpler to annotate and interpret. We present a listwise ordering model based on a set-to-sequence model that allows for events to be grouped together. We hypothesized that the listwise models would require less detailed annotation, less feature engineering, and also provide sufficient information for the downstream classification tasks. This is the first such model to be applied to clinical text

data and the results show that classification model performance improves when time information is included.

- Once we have extracted events and determined their temporal order, it is not trivial to input this information into a classifier. In Chapter 4 we also investigate different methods of converting the extracted timelines to numerical features, including event and time embeddings.
- In addressing the third research question (what types of temporal information are the most informative) we not only want to know what information was useful to the classifier, we also want to be able to provide a human-interpretable explanation of the classifier's predictions. Especially in the medical field, it is important not to treat artificial intelligence solutions as a black box. Both patients and physicians need a reason to trust the model's output, and thus we present a model of keyword-based explainability for the output of the classifiers in Chapter 6.

1.4 Structure of thesis

The structure of the rest of the thesis is as follows:

- **Chapter 2** examines the three different types of medical text and the datasets we use for experiments, along with the relevant annotation schemas.
- **Chapter 3** discusses existing event and time phrase extraction methods, and their performance on the 3 datasets.
- **Chapter 4** presents models for listwise temporal ordering of the extracted events in all 3 datasets. Material based on (Jeblee and Hirst, 2018).
- **Chapter 5** examines methods for representing the extracted timelines as features, as well as machine learning models for disease classification, and results on several different medical narrative datasets. Material based on (Jeblee et al., 2019a).
- **Chapter 6** presents a method for generating explanations for the output of disease classifiers.
- **Chapter 7** concludes the thesis and discusses the impact and limitations of this work, along with possible avenues of future research.

Chapter 2

Datasets of medical narratives with temporal annotation

2.1 Introduction

Medical text data comes in many different formats and styles. Some documents follow a specific structure (e.g. clinical notes), and some are free-form (e.g. verbal autopsy narratives). In addition, the medical terminology contained in the document may be very formal (such as terms from the Unified Medical Language System (UMLS)[1]) or very informal (such as patients' descriptions of symptoms). Depending on the source of the data, the text quality can also be highly variable. The style and structure of the text and the data quality can have a large impact on the performance of downstream machine learning and natural language processing (NLP) models.

In this chapter, we will examine three different types of medical narratives, which vary in formality and structure. We will first discuss annotation standards for events and time expressions. We examine formal health records in the form of clinical notes in Section 2.3, we look at verbal autopsy narratives (informal cause-of-death records) in Section 2.4, and clinical dialogues in Section 2.5. We conclude by discussing the characteristics of these three datasets in Section 2.6. The three datasets introduced in this chapter will be used for experiments in the rest of the thesis.

2.2 TimeML annotation schema

In this section we will discuss annotation standards for events and temporal expressions in text, including specific modifications for annotating time phrases in medical texts.

The datasets that will be presented in this chapter use modifications of the TimeML annotation schema, presented in the next section. The THYME corpus of clinical notes used the THYME-TimeML schema, and the verbal autopsy dataset uses Simple-TimeML.

[1] https://www.nlm.nih.gov/research/umls/index.html

2.2.1 TIMEX3 and TimeML

TimeML (Pustejovsky et al., 2003) is an XML-based temporal annotation scheme that includes guidelines for annotating events, timexes, and temporal relations. All TimeML tags have a *comment* attribute, where annotators can write clarifications or explanations.

Time expressions are annotated using the **TIMEX3** annotation standard, which is the latest evolution of the TIMEX standard. The TIMEX3 annotation includes two required attributes: *tid* (the time ID), and *type*, which can be one of: DATE, TIME, DURATION, or SET. The *functionInDocument* attribute allows for specifying whether the timex is used as an anchor for other phrases in the document (for example, the DCT). The possible values are: CREATION_TIME (used for DCT), EXPIRATION_TIME, MODIFICATION_TIME, PUBLICATION_TIME, RELEASE_TIME, RECEPTION_TIME, or NONE.

Other optional attributes include *quant* and *freq* (which specify quantifier and frequency values for SET types), *value* and *valueFromFunction* (which specify a time value in ISO format according to the TIDES 2002 guidelines (Ferro and Mani, 2001), *mod* (for temporal modifiers), *anchorTimeID* (which points to another TIMEX3 as a reference in the case of a relative expression), and *beginPoint*, and *endPoint* are used to define durations in terms of other time expressions if possible.

The **SIGNAL** annotation is used for function words such as "on" or "before" that modify time expressions or describe how they relate to events or other timexes.

The TimeML **EVENT** annotation is used for labeling events in text and classifying their attributes. The EVENT label is used for individual mentions of events, and the MAKEINSTANCE annotation represents an instance of an event. Each EVENT has an id attribute called *eid*, and a *class* attribute, which can have one of the following values: OCCURRENCE, PERCEPTION, REPORTING, ASPECTUAL, STATE, I_STATE, or I_ACTION.

The MAKEINSTANCE annotation maps the event ID to an eventInstanceID, and adds the following attributes: *pos* (part of speech), *tense*, *aspect*, *cardinality*, *polarity*, and *modality*. The *polarity* attribute can be POS (positive) or NEG (negative), and is positive by default. In our work, we use only the *polarity* attribute. The possible values for the other attributes can be found in the full TimeML specification[2].

The **TLINK** annotation captures a pairwise temporal relation between an event or timex (specified by *timeID* or *eventInstanceID*) and another entity (either an event or timex), specified by either the *relatedToTime* or *relatedToEvent* attribute. The *relType* attribute specifies the relation type. TimeML typically uses the 13 Allen relations, although in practice any temporal relation scheme could be used. Each document can have a DCT, which is specified by a TIMEX3 annotation with the feature *functionInDocument="CREATION_TIME"*. We include relations to the DCT in the dataset we use for our experiments.

TimeML also includes a subordination link annotation (SLINK) and aspectual link (ALINK), but these are not used in our work.

Inter-annotator agreement for temporal relations types in TimeML was found to be only .71 Cohen's kappa for the largest TimeML corpus (Derczynski, 2016), which shows that TimeML annotation is a fairly difficult task, even for expert annotators.

[2] http://www.timeml.org/publications/timeMLdocs/timeml_1.2.1.html

2.2.2 THYME-TimeML

In the news domain, several TimeML-annotated corpora are publicly available, such as the TimeBank corpus (Pustejovsky et al., 2006) and the AQUAINT corpus (Graff, 2002).

In addition to news datasets, there are currently several datasets of clinical notes annotated with a variation of TimeML, such as the THYME corpus (Styler et al., 2014), which includes clinical notes from brain and colon cancer patients, and the Informatics for Integrating Biology and the Bedside (i2b2) (Sun et al., 2013a) datasets. The THYME dataset (discussed in more detail in Section 2.3) is annotated with THYME-TimeML, a modification of TimeML developed for annotating clinical data, and the i2b2 dataset in annotated with a further modification of THYME-TimeML.

Each document in the THYME corpus has an explicit DCT attribute (*Doctime*), and each section can also have its own *Sectiontime*, since clinical notes may be updated multiple times. Events can also have three additional attributes: *contextual modality* ("Actual", "Hypothetical", "Hedged", or "Generic"), *contextual aspect* ("Intermittent"), and *permanence* ("Permanent" or "Finite", roughly corresponding to chronic vs. acute), although *permanence* was not actually annotated in the THYME dataset.

THYME-TimeML also includes a provision for annotating "specific temporal spans related to an implicit EVENT" as preoperative, postoperative, or intraoperative. (Tourille, 2018).

The main difference in temporal relation annotation for THYME-TimeML is the addition of the CONTAINS relation type. TLINK types are restricted to CONTAINS, OVERLAP, BEGINS-ON, ENDS-ON, and BEFORE (this set excludes SIMULTANEOUS relations, which have low occurrence, and inverse relations such as AFTER and DURING, which can be inferred). The annotation guidelines prioritize linking all events to their temporal container, and then linking container expressions, such that many relations can be inferred rather than annotated directly.

Similarly, Sun et al. (2013a) presented annotation guidelines used for the 2012 i2b2 project, which included a challenge for extracting temporal information from clinical narratives. The i2b2 2012 dataset, which was introduced as part of the Sixth i2b2 Natural Language Processing Challenge for Clinical Records (Sun et al., 2013b), contains 310 discharge summaries, which are annotated for events, time expressions, and temporal relations. The annotation scheme was based on THYME-TimeML, but SIGNALs, SLINKs, and ALINKs were removed, event attributes were modified, and section times were added. Event attributes that differ from standard TimeML included *type* ("problem", "test", "treatment", "clinical department", "evidential", "occurrence"), *modality* ("factual", "hypothetical", "hedged", "conditional"), and *polarity* ("positive", "negative").

However, Tissot et al. (2015) analyzed the manual annotations in the THYME dataset and found that some annotations were incorrect or inconsistent, such as DURATIONs being inaccurately labeled as DATEs, or function words being included in the annotation span, despite the guidelines stating that they should not be included. In addition, they found that a number of time phrases were not annotated at all, with the most common type of missing entities being SET.

This shows that annotating clinical notes for time information is a difficult task, and requires clear instructions with examples in order to provide an accurate standard with which to evaluate automated methods.

Example clinical note
HISTORY OF PRESENT ILLNESS AND HOSPITAL COURSE: For details, please refer to clinic notes and OP notes. In brief, the patient is a 47-year-old female with a posttraumatic AV in the right femoral head. She came in consult with Dr. X who after reviewing the clinical and radiological findings recommended she undergo a right total hip arthroplasty and removal of old hardware. After being explained the risks, benefits, alternative options, and possible outcomes of surgery, she was agreeable and consented to proceed and therefore on the day of her admission, she was sent to the operating room where she underwent a right total hip arthroplasty and removal of hardware without any complications. She was then transferred to PACU for recovery and postop orthopedic floor for convalescence, physical therapy, and discharge planning. DVT prophylaxis was initiated with Lovenox. Postop pain was adequately managed with the aid of Acute Pain team. Postop acute blood loss anemia was treated with blood transfusions to an adequate level of hemoglobin. Physical therapy and occupational therapy were initiated and continued to work with her towards discharge clearance on the day of her discharge. DISPOSITION: Home. On the day of her discharge, she was afebrile, vital signs were stable. She was in no acute distress. Her right hip incision was clean, dry, and intact. Extremity was warm and well perfused. Compartments were soft. Capillary refill less than two seconds. Distal pulses were present. PREDISCHARGE LABORATORY FINDINGS: White count of 10.9, hemoglobin of 9.5, and BMP is pending. DISCHARGE INSTRUCTIONS: Continue diet as before.

Table 2.1: Excerpt from a sample clinical note from MTSamples [3]

2.3 Formal medical narratives: clinical notes

Clinical notes are formal text documents written by medical professionals, typically as part of a patient's health record. The notes may document admission or procedure dates, test results, exam findings, summaries of a patient visit, observations from the medical staff, plans for treatment, etc. These documents often have many formal medical terms, abbreviations, and acronyms. Although these notes typically include temporal information such as dates of admission and procedures, inferring the correct context often requires domain knowledge. In addition, the structure and style of clinical notations can vary widely between clinics and between medical professionals.

The **THYME** (Temporal Histories of Your Medical Event) dataset is an annotated dataset consisting of clinical notes for brain and colon cancer patients (Styler et al., 2014). The full dataset includes 1254 clinical and pathology notes, annotated with the THYME-TimeML temporal annotation schema. The goal is to be able to use the annotated events and times to construct a timeline of the patient's history from the narrative.

Events in the THYME dataset are restricted to a single word. This is somewhat of a limitation as, in some cases, a longer phrase might better capture the details of the event, such as "pain in the left side of the chest" instead of just "pain". However, to our knowledge, all of the currently available clinical datasets with temporal relation annotations have events annotated as single words.

For our models, we use the annotated set of clinical notes, which includes a training set of 208 records, a development set of 103 records, and a test set of 106 records. Although the number of records available is very small, many of the documents are long and include hundreds of events. For our final experiments we perform 10-fold cross-validation using all THYME records in order to have a larger training set.

Since all available clinical note datasets contain private medical data, we cannot show real examples. Table 2.1 shows a portion of a constructed clinical note written as a sample by a medical transcriptionist (although this example is much shorter than the average note). This example is a discharge summary, which is slightly different in structure from the inpatient notes in the THYME dataset, but the style of the text and clinical entities are similar. In the THYME dataset, notes typically have many sections, which each have their own datetime.

We can see that the note contains many medical terms, including acronyms such as "PACU", abbreviations such as "postop", procedures such as "right total hip arthroplasty", and laboratory findings such as "hemoglobin of 9.5". Although the note is written in natural language, the information is concise and contains very little irrelevant text. These notes also typically have subsections indicated by text in all-caps, although the actual section headings vary depending on the type of note and the conventions of individual clinics and staff.

2.4 Informal medical narratives: verbal autopsies

The **Million Death Study (MDS)** is a program in India to collect verbal autopsy (VA) records of recent community deaths (Westly, 2013; Jha, 2014). The program began in 2001 with the goal of surveying deaths that occurred in 1.3 million households, and has coded over 600,000 deaths so far. Narratives are collected in a variety of languages, primarily English and Hindi, but also including Marathi, Punjabi, and other Indian languages (see Figure 2.1 for the distribution of languages in the dataset we use). The survey includes demographic information and information about risk factors such as genetics and environmental factors, especially for preventable causes of death (Gomes et al., 2017).

In 2017, the All India Institute of Medical Sciences (AIIMS) created the MINErVA (Mortality in India Established through Verbal Autopsy) platform to continue the task of conducting VA surveys in India (Krishnan et al., 2020). The new system has coded over 75,000 VA forms so far, and aims to move towards electronic data capture.

Compared to previous methods, the MDS reduced the percentage of deaths surveyed before age 70 that were recorded as "Ill-defined". For verification, a random re-sampling of the original population was conducted with new interviews, and this re-sampling found a similar CoD distribution to the original survey. The MDS also includes geospatial mapping, which allows for analyses of how different CoDs correlate to different regions, and how these patterns change over time (Gomes et al., 2017).

In the dataset we use in this work, the forms were handwritten, scanned, and eventually transcribed into a digital format[4]. The non-English narratives were translated into English at the time of transcription. Each collected record is then coded for CoD separately by two physicians, using the International Classification of Diseases version 10 (ICD-10) (World Health Organization, 2008). If the two assigned codes do not agree, the physicians are given the chance to view each other's assigned codes and key phrases and decide if they want to update their own. If the codes still do not match after this reconciliation process, a third, more senior physician reviews the record and determines the ICD-10 code. Along with the ICD-10 code, physicians enter a list of key phrases explaining their diagnoses, which can come directly from the narrative or be written in by the physician.

ICD codes are organized in a hierarchical structure, with a letter indicating the category of disease, such as infectious diseases (A and B) or neoplasms (C and D), and a number indicating the specific type

[4]As of 2018, the MDS has adopted electronic VA forms (Gomes et al., 2017)

of disease. For example, A00 is Cholera, and A01 is typhoid fever. Since there are thousands of possible individual ICD-10 codes, we group them into broader categories for classification (shown in Table 2.2).

The dataset is de-identified, which means that the names of people and places are replaced with "XXX". However, since the best methods identify around 95% of personal health information (PHI) in clinical notes, there are many instances where names and other personally identifiable information has been missed by the de-identification process and is still present in the text (Abdalla et al., 2020). It is also important to note that the removal of names from the narratives does not constitute anonymization. Because our classification algorithms rely on time information, dates have not been changed or removed, so the record includes the person's birth and death date along with their region of residence and several other demographic factors, which may provide enough information to re-identify some of the subjects. For this reason, the dataset is confidential.

In addition to the primary MDS dataset, we also have a set of VA records collected for a randomized control trial (RCT) conducted by the Centre for Global Health Research (CGHR) in two states of India: Punjab and Gujarat. The purpose of the study was to compare physician coding of VAs to several automated coding methods. A total of 9374 deaths were surveyed, with 4651 dual-coded by physicians according to the MDS protocol, and 4723 coded by automated methods (Jha et al., 2019). The records for physician coding were collected using MDS forms, and the records for automated coding were collected electronically using a form based on the WHO 2014 standard VA instrument (Nichols et al., 2018). However, only the physician-coded dataset included free-text narratives, and thus we use only this subset for our experiments, which we refer to as the RCT dataset.

Population-level concordance with the true distribution is calculated using cause-specific mortality fraction accuracy (CSMFA), which is a measure of how similar the predicted cause distribution is to the true distribution (Murray et al., 2011b). The RCT study found a maximum CSMFA of 97% for adult deaths, 87% for child deaths, and 84% for neonatal deaths. However, the average individual-level concordance of the algorithms to physician coding was only 62%, lower than the previously reported 76% (Jha et al., 2019).

We combine 5,532 records from the RCT physician-coded dataset with the original MDS data for training and testing our models, and we refer to this dataset as MDS+RCT.

2.4.1 Annotation of verbal autopsies: Simple TimeML

For the MDS VA dataset, we use a simplified version of TimeML for two reasons: to increase annotation speed, and because VA narratives are often more vague than clinical notes and therefore very few attributes are discernible for most entities.

For TIMEX3 annotations we include only the "type" attribute, and for EVENTs we include the following attributes: "polarity" (whether the event is negated or not), "relatedToTime" (the ID of the time phrase related to the event), and "signalID" (the ID of the associated SIGNAL phrases, if there is one). We also introduce a "rank" attribute, which is an integer starting from 1 specifying the event's position in the chronological timeline, where 1 is the earliest occurring event. Any number of events can have the same temporal rank value.

For simplicity, we also collapse the concept of events and event instances, and therefore drop the MAKEINSTANCE annotation. Although this could result in duplicate events and prevents us from doing coreference resolution, in the VA dataset most events are only mentioned once since the narratives are fairly short. In cases where there are duplicate mentions of the same event, they can be assigned the

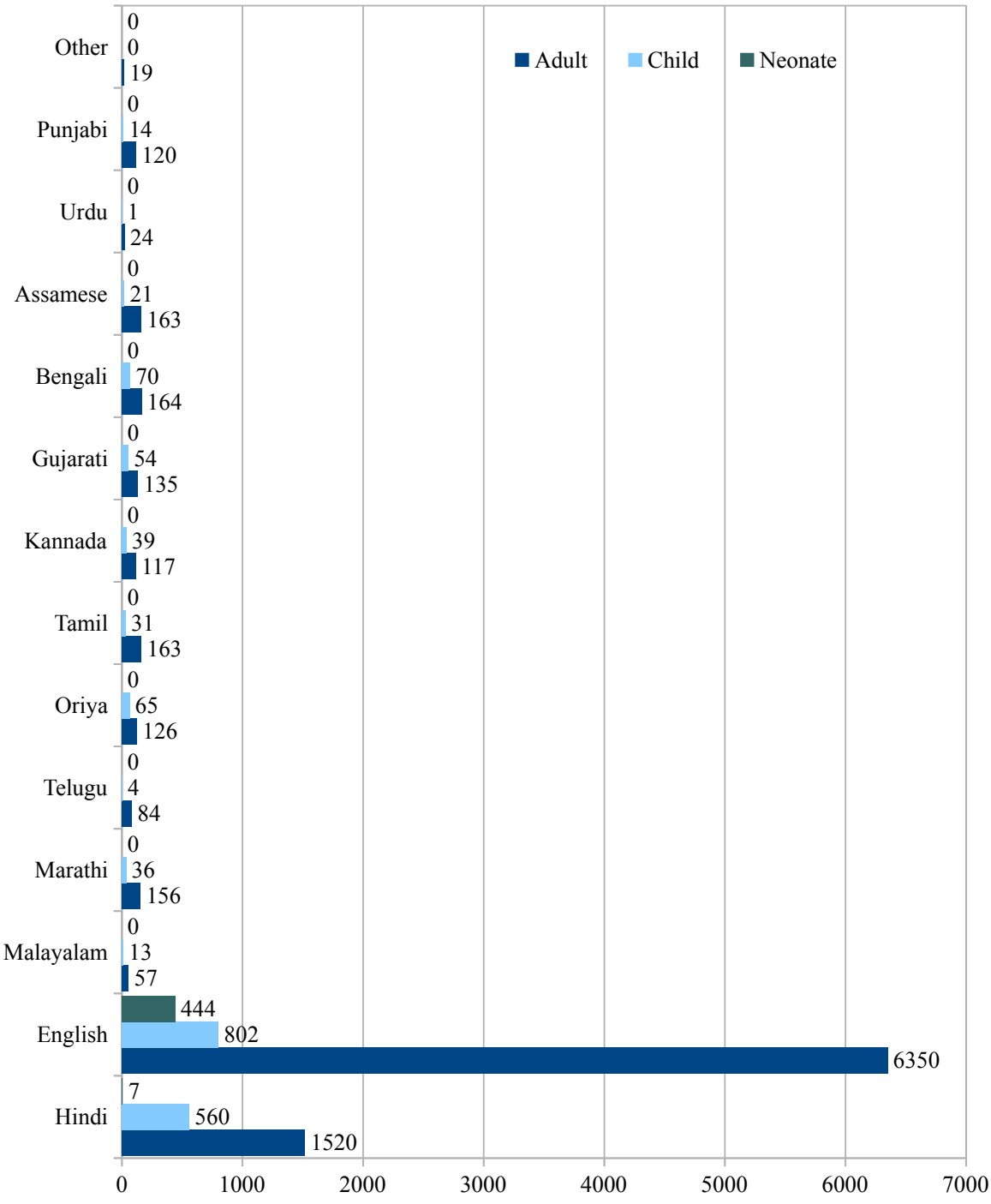

Figure 2.1: Distribution of original narrative languages in the MDS+RCT dataset.

Num	Category	Num	Category
	Adult		Child
1	Acute respiratory infections	1	Pneumonia
2	Tuberculosis	2	Diarrhoea
3	Diarrhoeal	3	Malaria
4	Unspecified infections	4	Other infections
5	Maternal	5	Congenital anomalies
6	Nutrition	6	Non-communicable diseases
7	Chronic respiratory diseases	7	Injuries
8	Neoplasms	8	Nutritional
9	Ischemic heart disease	9	Other
10	Stroke	10	Ill-defined
11	Diabetes	11	Cancer
12	Other cardiovascular diseases		
13	Liver and alcohol		
14	Other non-communicable diseases		
15	Road traffic incidents		
16	Suicide		
17	Other injuries		
18	Ill-defined		
	Neonate		
1	Prematurity and low birthweight		
2	Neonatal infections		
3	Birth asphyxia and birth trauma		
4	Congenital anomalies		
5	Ill-defined		

Table 2.2: Cause-of-death categories used for the MDS data.

same rank value.

2.4.2 English VA narratives

MDS+RCT dataset

The main verbal autopsy dataset we use consists of approximately 16,000 records from the MDS and RCT datasets, and covers adult, child, and neonatal deaths. A subset of 700 records were annotated for event and time information as described above. The distribution of annotated and unannotated records by CoD category is shown in Figure 2.2.

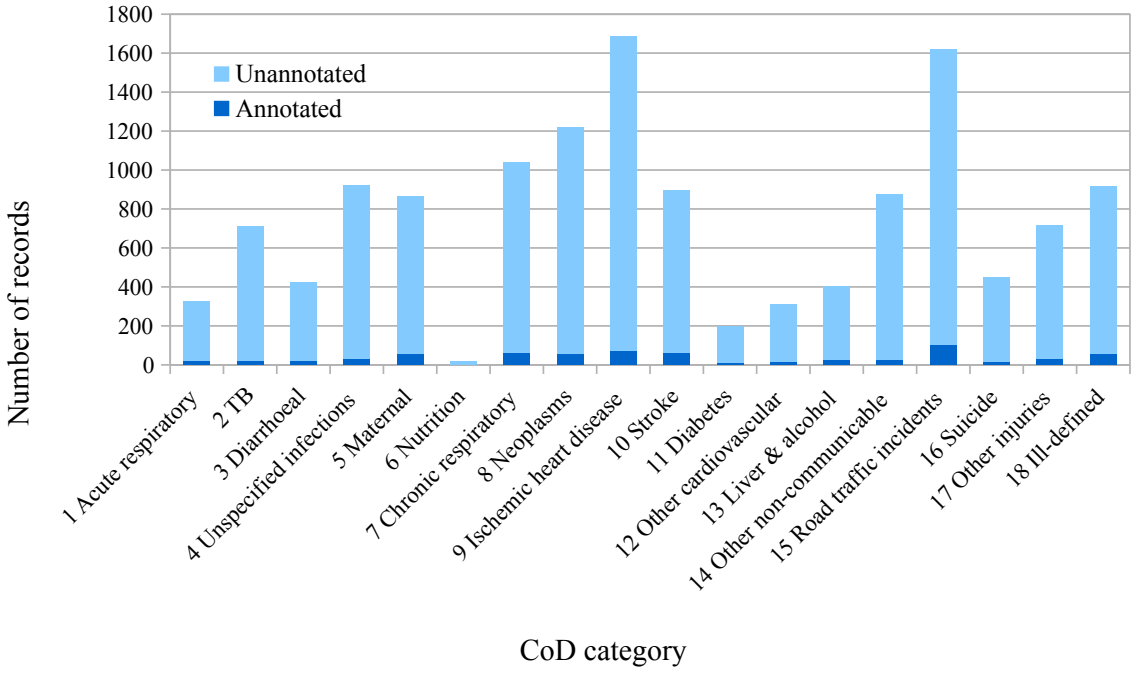

Figure 2.2: CoD distribution of records of adult deaths from the MDS+RCT dataset, including annotated and unannotated records.

The dataset of VA narratives in English includes some narratives that have been translated into English from various other languages. The quality of the translation is highly variable and sometimes results in English text with poor grammar, unclear wording, misspellings, and occasional untranslated terms. Even the narratives that were originally written in English often suffer from these problems. This can result in text that is difficult to interpret. In addition, many pre-trained NLP tools that expect well-formed text, such as dependency parsers, will fail on VA narrative text.

Table 2.3 shows several examples of verbal autopsy narratives from the MDS dataset of adult deaths. In the first example, the narrative actually contains the CoD ("heart failure"), whereas the second example only contains symptoms, not the actual CoD. We can also see that there are several misspellings and grammatical issues. In the first example, we can infer that "person was head" really should be "person was dead", but this typo is difficult to correct automatically because "head" is also a valid word.

Another phenomenon that occurs frequently in the MDS narratives is the inconsistent and incorrect usage of gender pronouns. The subjects of the narrative are often referred to with the the wrong pronouns,

Narrative	**Physician-certified CoD category**
Heart failure. The patient death due to breathlessness. The person suffering paralysis and stroke lost on year with chest pain very pressure after then person was head.	Cardiovascular disease
One day 13/03/01 he fell ill with some fever and chest pain who called the Doctor. On 15/03/01 the deceased was crying in the chest pain and high fever. We were ready to shift. The patient to the Hospital, some water came out from the deceased mouth and closed his eyes and passed away.	Acute respiratory infections

Table 2.3: Two example narratives from the MDS dataset (adult deaths).

even while the correct pronouns are used in other parts of the same narrative. For example, one narrative states: "When his temperature did not come down, she was admitted to XXX Hospital". However, it is clear from the context of the narrative that there is a single subject. This is a frequent source of error in narratives translated from Hindi to English, since Hindi has a commonly used gender-neutral third-person pronoun, and English does not.

Some narratives are also vague or uninformative. Especially for older subjects, the narrative may state something like "the person died from old age". In some cases the symptoms themselves (such as cough and fever) do not provide enough information to distinguish between possible causes. And in some cases, the narrative provides conflicting accounts of what happened, such as from the family members vs. neighbours. However, since these ambiguities are part of the nature of verbal autopsies, it is important for machine learning models to handle them. In this case, we have a best estimate of CoD from multiple physicians, so our goal is to develop models that can replicate a human diagnosis.

For CoD classification, the ICD-10 codes are grouped into broader CoD categories (18 for adult deaths, 9 for child deaths, and 6 for neonatal deaths). These categories are based on the distribution of CoDs in the collected MDS data, and include an explicit "Ill-defined" class, which is used to categorize deaths where the record does not have enough information to make a diagnosis. We refer to these as the MDS CoD categories (shown in Table 2.2). Figures 2.3, 2.4, and 2.5 show the distribution of the MDS categories in the MDS+RCT dataset.

The MDS categories are an alternative to the larger set of standard CoD categories used by the World Health Organization (WHO) VA survey. The WHO provides a hierarchical classification of ICD-10 codes for CoD diagnoses, with a total of 64 categories (although only 35 of them appear in the MDS data) (see appendix). However, the MDS categories are tailored to the MDS dataset and they provide a more even distribution and a smaller number of categories for classification (18 vs. 35). However, the WHO categories are an international standard used by other datasets. For this reason, the WHO categories may be better for generalizing our models to other datasets, but the MDS categories are more suited to classifying the datasets we currently have. Thus, we will examine both in Chapter 5. Figures 2.6, 2.7, and 2.8 show the distribution of WHO categories in the MDS+RCT dataset.

Table 2.4 shows the number of records for each age group in the MDS and RCT datasets. The adult age group includes deaths between 15–69 years of age, the child age group includes deaths from 29 days to 14 years, and the neonatal group includes deaths before 29 days. Note that the number of categories is much smaller for neonatal deaths as there are a smaller number of CoDs, and also fewer records.

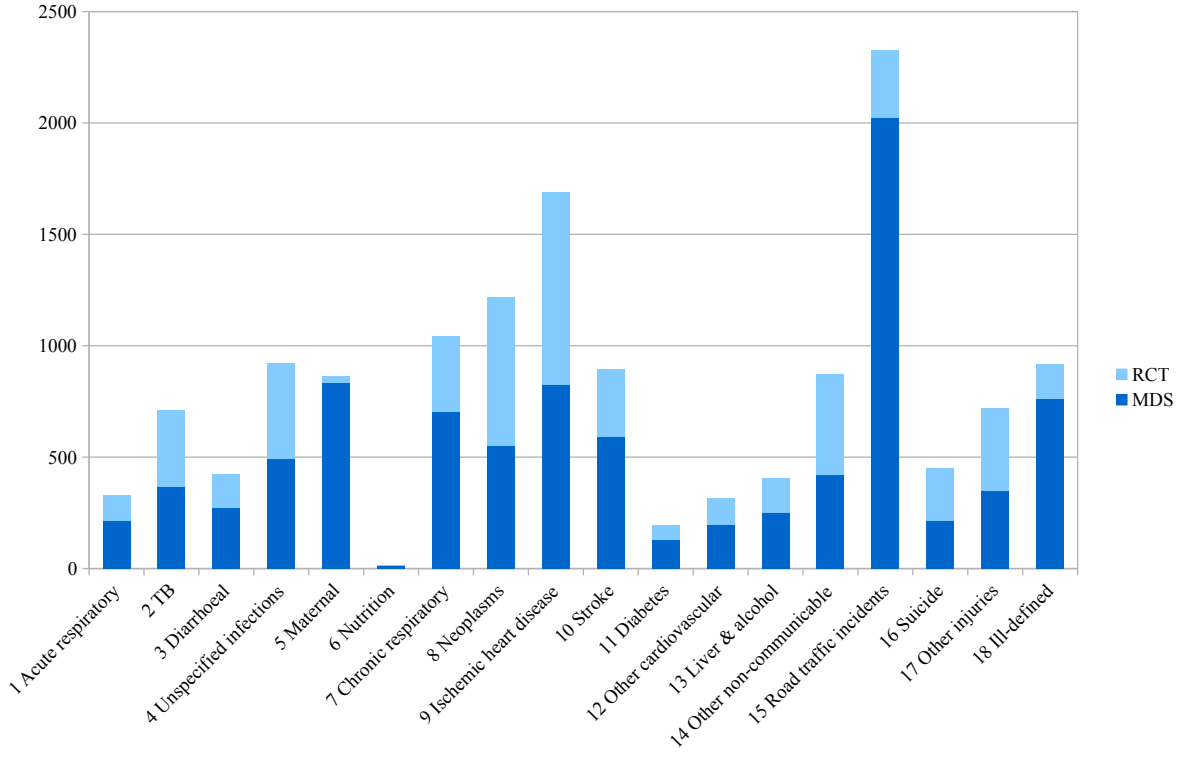

Figure 2.3: CoD distribution of records of adult deaths from the MDS+RCT dataset.

Dataset	**Adult**	**Child**	**Neonate**	**Total**
MDS	9,208	1,717	451	11,376
RCT	5,105	256	451	5,532
MDS+RCT	14,313	1,973	622	16,908

Table 2.4: Dataset sizes for each age group in the MDS and RCT datasets.

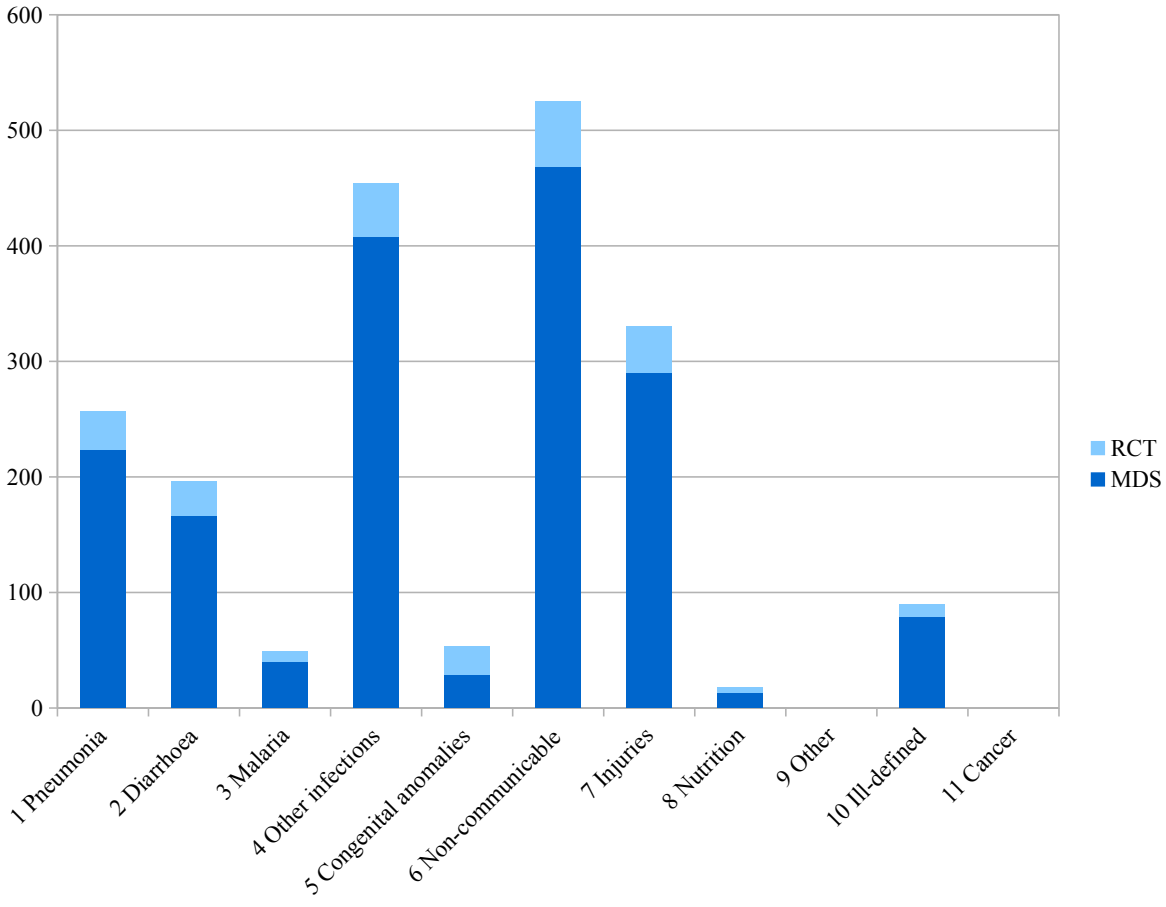

Figure 2.4: CoD distribution of records of child deaths from the MDS+RCT dataset.

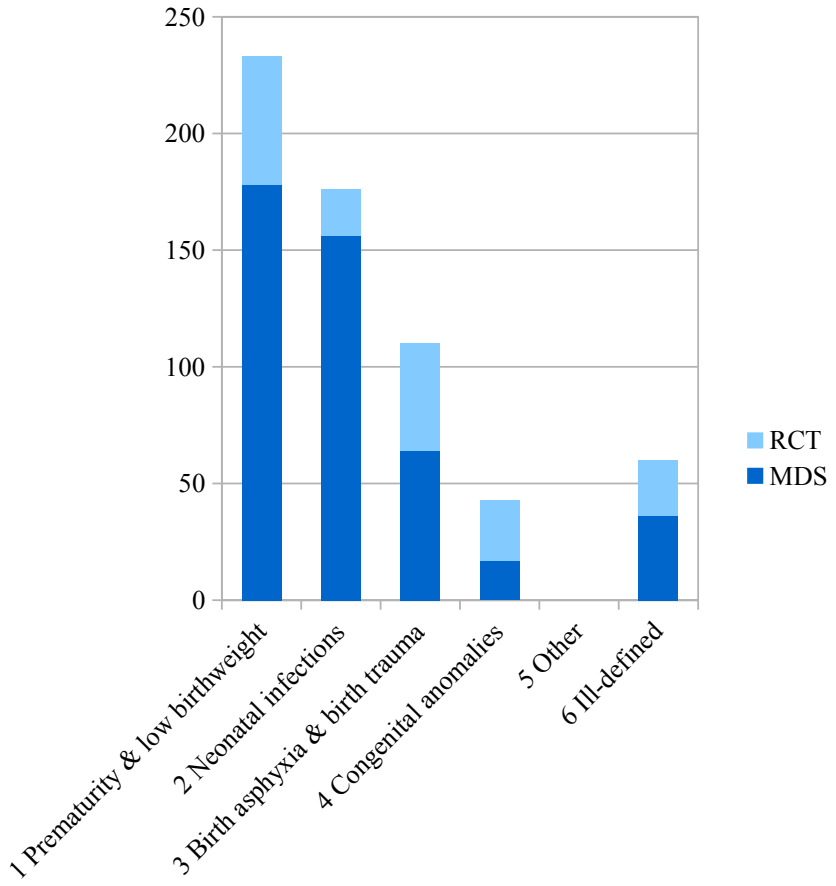

Figure 2.5: CoD distribution of records of neonatal deaths from the MDS+RCT dataset.

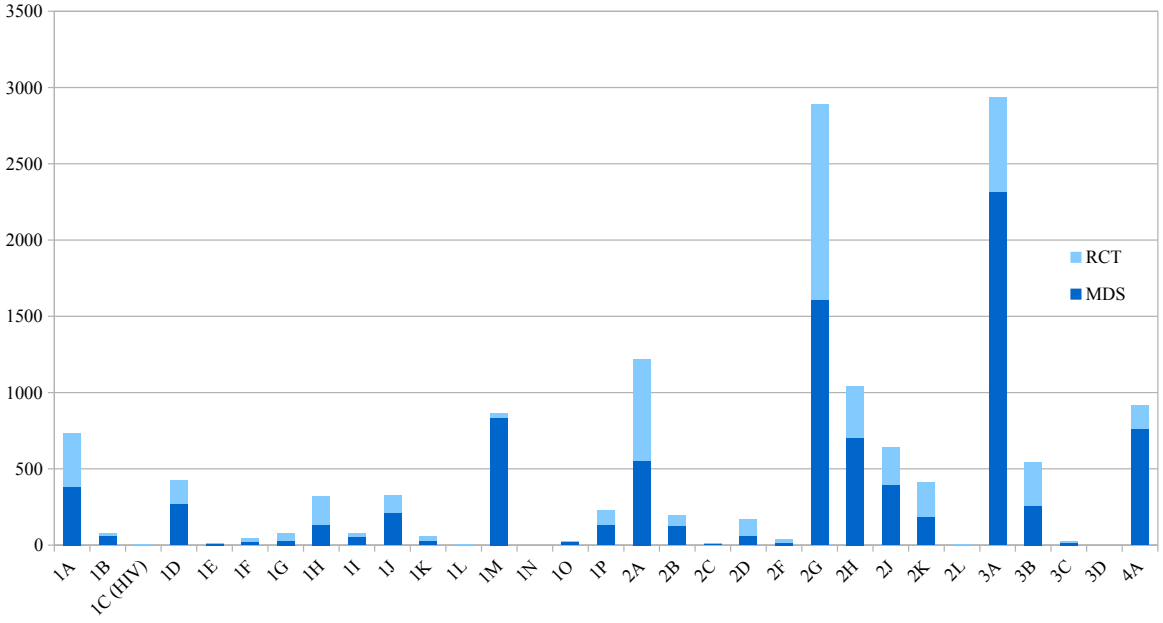

Figure 2.6: WHO CoD distribution of records of adult deaths from the MDS+RCT dataset.

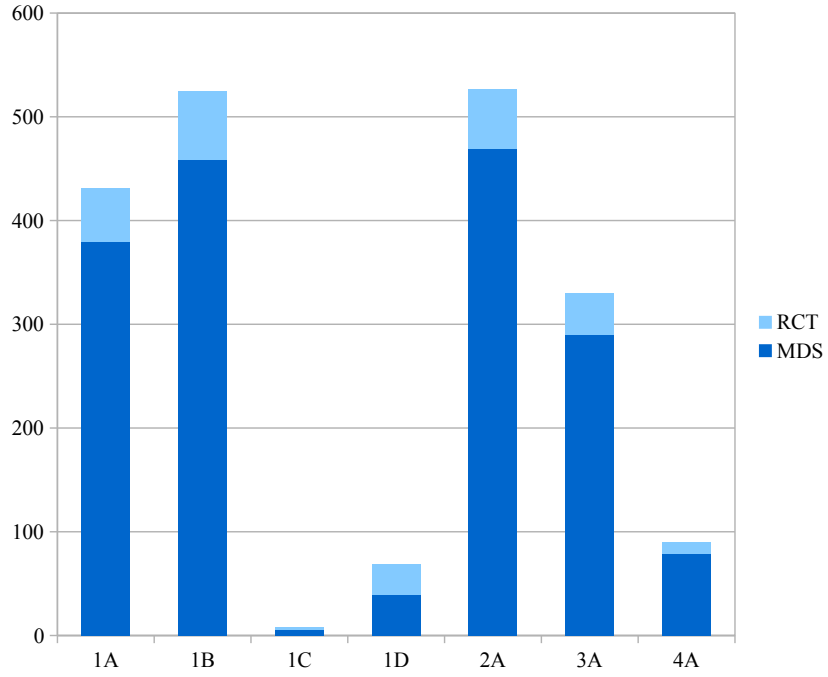

Figure 2.7: WHO CoD distribution of records of child deaths from the MDS+RCT dataset.

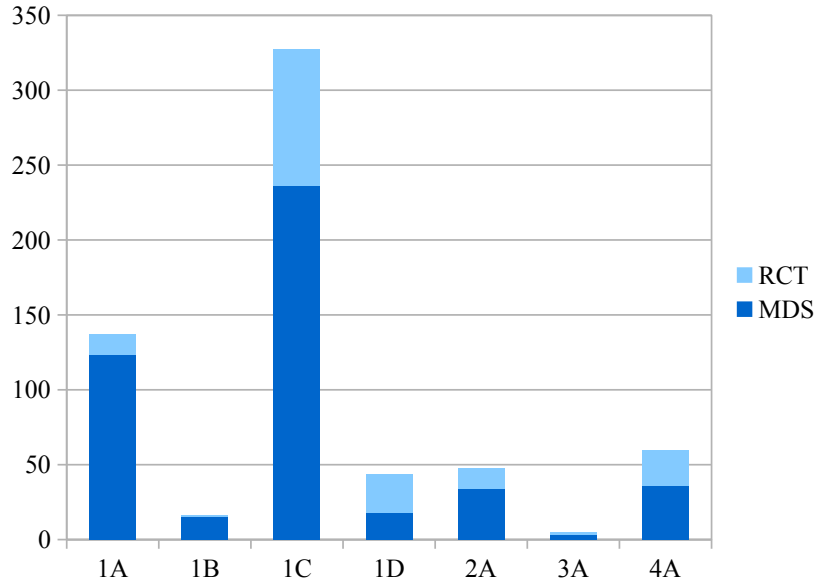

Figure 2.8: WHO CoD distribution of records of neonatal deaths from the MDS+RCT dataset.

Since VA records typically do not have a medically certified cause of death based on a physical autopsy, our ground truth labels come from the consensus of the coding physicians. Table 2.5 shows the agreement between the original two codes assigned by the coding physicians. Although these disagreements were eventually resolved through reconciliation or adjudication, we are interested in how difficult it is for physicians to code each category. The rows correspond to the CoD category of the final assigned code, and the columns indicate the CoD categories of the original assigned code from each physician.

From these confusion matrices, we can see that some categories are more difficult to code than others. For example, adult category 15 (Road traffic incidents) has very high initial physician agreement, while category 1 (Acute respiratory infections) has very low agreement. We expect machine learning classifiers to also perform better on the classes that are easier to code for human physicians, since these are usually the result of clearer narratives and more specific symptoms.

PHMRC dataset

The Population Health Metrics Research Consortium (PHMRC) gold standard verbal autopsy dataset includes VA records from India, Philippines, Mexico, and Tanzania (Murray et al., 2011a). Unlike the MDS dataset, the PHMRC dataset was collected from hospital deaths with a verified CoD. While this provides a useful ground truth for CoD coding, hospital deaths typically have more specific information available than community deaths, meaning that the narratives in the PHMRC dataset are not necessarily reflective of the kinds of narratives that we might expect from deaths that occurred outside of health facilities. In addition, the distribution of deaths that occur in health facilities can be very different from the distribution of deaths that occur elsewhere (Gomes et al., 2017).

However, not all cases in the PHMRC dataset have useful narratives. When excluding narratives that were empty or provided no information (such as "Respondent had nothing to add"), we discovered only 7,743 records with useful narratives out of 11,979. Even after this process, there are some narratives that have very little symptom information. In fact, some are merely a review of the quality of service in the hospital, which does not help us predict CoD. Table 2.6 shows two example narratives from the PHMRC dataset. However, the PHMRC dataset is one of the only freely available coded VA datasets, which has been used in other automated CoD coding work (McCormick et al., 2016; Miasnikof et al., 2015; Serina et al., 2016; Chowdhury et al., 2019). The dataset is available online [5].

The dataset uses a set of 34 CoD categories for adult deaths, 21 for child deaths, and 11 for neonatal deaths (Serina et al., 2015), shown in Table 2.7. We will refer to these as the PHMRC CoD categories.

2.4.3 Hindi VA narratives

In the MDS dataset, there are many narratives that were originally written in Hindi, as well as many in other Indian languages. Since translation may produce errors and lost information, and is also a costly and time-consuming step in the data pipeline, we prefer to work with the original narratives. To this end, 500 Hindi narratives were transcribed from scans of the handwritten narratives using the ITRANS transliteration scheme[6] (since transcribers typically use English keyboards), and then converted into Devanagari script using the ITRANS mapping. These records also have CoD codes and English key phrases.

[5] http://ghdx.healthdata.org/record/ihme-data/population-health-metrics-research-consortium-gold-standard-verbal-autopsy-data-2005-2011 as of May 2020.

[6] https://www.aczoom.com/itrans/

Cat	*n*	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	*329*	0.502	0.067	0.006	0.067	0	0	0.155	0.003	0.049	0.012	0.006	0.012	0.003	0.027	0	0	0	0.091
2	*710*	0.051	0.749	0.001	0.034	0.010	0.001	0.079	0.004	0.013	0.003	0.003	0.001	0.003	0.011	0.001	0	0.001	0.034
3	*425*	0.012	0.005	0.694	0.108	0	0	0.019	0.009	0.009	0.009	0.012	0.002	0.007	0.054	0	0	0	0.059
4	*923*	0.016	0.011	0.020	0.665	0.002	0.004	0.007	0.009	0.009	0.007	0.005	0.010	0.043	0.046	0.003	0.001	0.013	0.130
5	*821*	0.002	0.002	0.001	0.022	0.883	0.006	0	0.006	0.006	0.001	0	0.004	0.006	0.017	0	0	0.018	0.024
6	*17*	0	0	0.059	0	0	0.235	0	0	0	0	0	0.059	0	0.176	0	0	0	0.471
7	*1039*	0.050	0.032	0.001	0.015	0	0	0.756	0.003	0.034	0.005	0.001	0.022	0.001	0.009	0	0	0.002	0.069
8	*1212*	0	0.012	0.005	0.008	0	0	0.005	0.882	0.005	0.003	0.002	0.002	0.012	0.036	0	0	0	0.026
9	*1687*	0.009	0.004	0.003	0.011	0	0	0.019	0.002	0.772	0.026	0.007	0.044	0.003	0.013	0	0	0.005	0.081
10	*895*	0.002	0.002	0.006	0.023	0.001	0	0.011	0.007	0.044	0.749	0.012	0.035	0.004	0.021	0	0	0.015	0.068
11	*196*	0.010	0.015	0.020	0.036	0	0	0.010	0.005	0.036	0.036	0.633	0.010	0.005	0.128	0	0	0.005	0.051
12	*314*	0.019	0.029	0.010	0.029	0	0	0.096	0.006	0.134	0.057	0.022	0.420	0.003	0.029	0	0	0.010	0.137
13	*405*	0	0.010	0.010	0.057	0	0	0.007	0.015	0.040	0.005	0.002	0.012	0.714	0.042	0.005	0.015	0.005	0.062
14	*872*	0.007	0.003	0.008	0.042	0	0.003	0.011	0.022	0.028	0.015	0.019	0.010	0.021	0.698	0.001	0.009	0.015	0.086
15	*2320*	0	0.001	0.002	0.003	0	0	0.002	0	0.003	0.006	0	0.002	0.003	0.009	0.901	0.001	0.051	0.015
16	*451*	0	0	0.004	0.004	0.002	0	0.002	0	0.007	0.002	0	0	0.011	0.040	0.002	0.860	0.049	0.016
17	*718*	0.004	0.001	0.001	0.015	0.004	0	0.008	0.006	0.019	0.025	0.001	0.004	0.010	0.015	0.014	0.022	0.801	0.047
18	*914*	0.008	0.008	0.008	0.044	0.008	0.001	0.018	0.009	0.028	0.016	0.005	0.013	0.007	0.037	0.001	0.001	0.011	0.778

(a) Physician confusion matrix for adult deaths (MDS categories)

Cat	*n*	1	2	3	4	5	6	7	8	9	10
1	*255*	0.718	0.063	0.016	0.090	0.024	0.043	0	0.016	0	0.031
2	*194*	0.046	0.789	0.031	0.072	0.005	0.041	0	0.005	0	0.010
3	*49*	0.041	0.041	0.673	0.163	0	0.041	0	0	0	0.041
4	*452*	0.139	0.104	0.062	0.520	0.013	0.086	0.004	0.029	0	0.042
5	*53*	0.113	0.057	0	0.075	0.491	0.170	0.019	0.019	0	0.057
6	*522*	0.027	0.011	0.008	0.086	0.019	0.745	0.019	0.025	0	0.059
7	*330*	0	0.003	0	0.009	0	0.009	0.967	0	0	0.012
8	*18*	0	0	0	0.111	0	0.333	0	0.222	0	0.333
10	*89*	0.045	0.022	0.034	0.124	0.011	0.079	0.011	0.101	0	0.573

(b) Physician confusion matrix for child deaths (MDS categories)

Cat	*n*	1	2	3	4	5	6
1	*233*	0.674	0.064	0.073	0.017	0	0.172
2	*176*	0.114	0.739	0.045	0	0	0.102
3	*110*	0.164	0.136	0.409	0.045	0	0.245
4	*43*	0.139	0	0.047	0.721	0	0.093
5	*0*	0	0	0	0	0	0
6	*60*	0.033	0.050	0	0	0	0.917

(c) Physician confusion matrix for neonatal deaths (MDS)

Table 2.5: Confusion matrices for initial physician CoD coding on the VA dataset. Rows are the correct CoD categories and columns are the predicted categories. n is the number of records belonging to that category in the test set. Darker shading indicates higher accuracy. Note: A small number of codes which were not found in the ICD-10 to category mapping were assigned to the "Ill-defined" class. 65 adult records and 11 child records were excluded because they were missing intermediate codes.

Narrative	Physician-certified CoD category
deceased had jaundice a month ago which was being treated by herbal medicine. 6-7 days ago had fever and was admitted to hospital. two days ago had stomach ache. had become very week, could not speak or walk properly. next day , was admitted to hospital where he expired one day later at 2 pm. was also given oxygen	Other Infectious Diseases
the medical attention was very good. i will not complain about the hospital, but about the patients that we take there. sometimes they are taken there with many diseases. i am thankful that they charged me little.call after 7 pm, after 15 days. tel [PHONE] . [PERSON] says the certificate is with her brother and she does not think he will lend it to her, as her brother did not write a will and he wants to get the house.	Stroke

Table 2.6: Two example narratives from the PHMRC dataset (adult deaths).

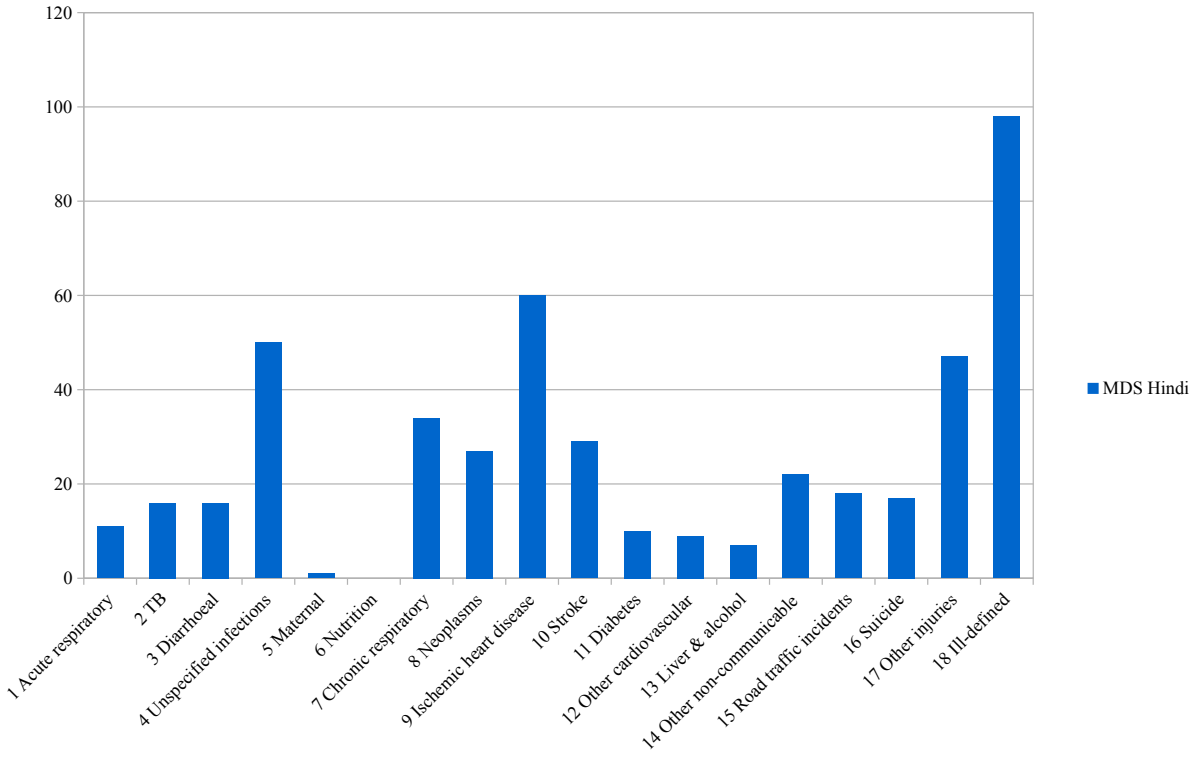

Figure 2.9: CoD distribution of records of adult deaths from the MDS dataset of 500 Hindi narratives.

Adult Category	**Child Category**
AIDS	AIDS
Asthma	Bite of venomous animal
Bite of venomous animal	Diarrhea/Dysentery
Breast cancer	Drowning
Cervical cancer	Encephalitis
Cirrhosis	Falls
Colorectal cancer	Fires
COPD	Hemorrhagic fever
Diabetes	Malaria
Diarrhea/Dysentery	Measles
Drowning	Meningitis
Epilepsy	Other cancers
Esophageal cancer	Other cardiovascular diseases
Falls	Other defined causes of child deaths
Fires	Other digestive diseases
Homicide	Other infectious diseases
Acute myocardial infarction	Pneumonia
Leukemia/Lymphomas	Poisonings
Lung cancer	Road traffic
Malaria	Sepsis
Maternal	Violent death
Other cardiovascular diseases	
Other infectious diseases	
Other injuries	
Other non-communicable diseases	
Pneumonia	
Poisonings	
Prostate cancer	
Renal failure	
Road traffic	
Stomach cancer	
Stroke	
Suicide	
TB	
Neonate Category	
Birth asphyxia	
Congenital malformation	
Meningitis/Sepsis	
Sepsis with local bacterial infection	
Preterm delivery	
Preterm Delivery (without RDS) and birth asphyxia	
Preterm Delivery (with or without RDS) and sepsis	
Preterm Delivery (without RDS) and sepsis and birth asphyxia	
Preterm Delivery with Respiratory Distress Syndrome	
Preterm Delivery without Respiratory Distress Syndrome	
Stillbirth	

Table 2.7: Cause of death categories used in the PHMRC dataset.

Narrative in Hindi	Translation
उनकी उम्र 60 साल थी, वह पुरुष था। उसे बुखार और चक्कर आना और पूरे शरीर में दर्द था। वहां कोई और बीमारी नहीं थी। उसे 3 दिनों से बुखार था, और यह कम था और यह ऊपर और नीचे की तरह था और बुखार मृत्यु तक चली गई है। और अपनी पत्नी के अनुसार, उसका पूरा शरीर दर्द हो रहा था और उसके पास भी 3 दिनों के बाद से चक्कर आना था। और वह गांव में दुकान से दर्द के लिए दवाएं ला रहा था। मेडिकल इन्फो उन्होंने गांव में स्टोर से दवाएं लीं। 1 और वह केवल गांव में दुकान से दर्द के लिए दवा ले रहा था, और किसी भी अस्पताल में भर्ती नहीं हुआ था और उसकी मृत्यु घर पर ही हुई ि या	His age was 60 years, he was male. He had fever and giddiness and whole body pain. No other illness was there. He had fever since 3 days, and it was low and it was like up and down and the fever has lasted till death. And according to his wife, his whole body was paining and he had giddiness too that was since 3 days. And he was bringing medications for pain from the store in the village. Medical Info He took medicines from the store in the village worth Rs. 1 and he was taking medicines for pain from the store in the village only, and was not admitted in any hospital and his death has occurred at home itself.

Table 2.8: A narrative in Hindi from the MDS dataset (adult deaths), "Ill-defined" category.

Although there are other records in languages that use the Devanagari script (such as Marathi and Punjabi) that have been transcribed with Latin characters, no official transliteration scheme was used, and thus the transliterations are not consistent across narratives, or even within the same narrative. This makes it difficult to automatically convert these narratives back to digital Devanagari text. This is an area for future work, either in converting the script or using the Latin transcriptions directly.

Table 2.8 shows one of the narratives from the MDS dataset in the original Hindi. Hindi has many linguistic differences from English. For example, Hindi has a gender-neutral third person pronoun, whereas English documents typically use "he" or "she". This can result in the narratives translated from Hindi having incorrect pronouns. Also, English narratives might provide implicit gender information that a Hindi narrative would not provide. However, in order to conduct error analysis, it is ideal to have a native speaker who can look at the narratives along with the model output.

2.5 Clinical dialogues

Some material based on previously published work (Jeblee et al., 2019b).

For many clinical encounters, much of the information in the clinical note comes from a verbal discussion between the physician and the patient. Typically this conversation is not captured, and the physician must rely on their memory and notes taken during the visit to write up a clinical note later. However, given a transcript of the conversation, we can extract such clinical information automatically, and use temporal ordering models to determine chronology.

These documents are quite different from verbal autopsies and clinical notes because they consist of conversational utterances from multiple speakers. Consequently, the density of clinically relevant information is much lower, since the conversations often include small talk and other irrelevant discussions. In addition, because the physician is talking to a patient they will typically use more informal terms rather than the medical terms and abbreviations that are used in clinical notes. The topics of discussion can also be less focused, since a patient may have multiple health issues beyond the primary reason for

their visit.

2.5.1 Clinical dialogue dataset

In order to examine how information extraction and temporal ordering models perform on different types of medical data, we use a dataset of 800 clinical conversation transcripts purchased from Verilogue[7]. Each record includes the audio of the conversation, a manual transcription with timestamps and speaker labels, a document creation time, some demographic information about the patient, and a primary diagnosis (as free text). The dataset has 7 main diagnosis categories, as shown in Table 2.9. The "Other" category includes about 20 diagnoses with very low instance counts.

Table 2.10 shows a constructed example of a clinical dialogue transcript. As in most doctor-patient conversations, the dialogue includes relevant medical information such as the patient's primary condition (*diabetes*), symptoms (*numbness in toes*), and medications (*Metformin*). However, it also includes irrelevant casual conversation about a Toronto sports team. It is important in this case for any automated system not to misinterpret the time information or semantic expressions about sports as medical information.

Diagnosis class	**Num of records**
Attention Deficit Hyperactivity Disorder	100
Depression	100
Chronic obstructive pulmonary disease (COPD)	101
Influenza	100
Osteoporosis	87
Type II diabetes	86
Other	226

Table 2.9: Diagnosis categories in the Verilogue dataset.

The dataset is currently being annotated for event and time information by trained medical annotators. So far, we have 475 conversations with event and time annotations, 165 of which have the temporal order of relevant events annotated. Note that in contrast to the VA and THYME datasets, where every annotated event also has an annotated temporal order (also referred to as a rank), in the Verilogue dataset only the events that are included in the final note are ranked, which is a very small percentage of all events mentioned in the text.

2.5.2 Annotation of clinical dialogues

For the clinical dialogue dataset, a medically-trained annotator identified relevant entities in the text transcripts of the conversations, including a number of attributes. The annotated entity types include: *TIMEX3*, *sign/symptom*, *reason for visit*, *medication*, *anatomical location*, *investigation/therapy*, *referral*, *diagnosis*, *quantity*, and *quality*.

Time expressions are annotated with the "type" attribute only. All other entities have the following optional attributes: "type", "modality", and "pertinence". "Pertinence" specifies the disease or condition for which a given symptom or medication is relevant. For example, Adderall is a medication which is

[7] https://www.verilogue.com/solutions/artificial-intelligence-machine-learning/

Speaker	Utterance
Doctor	It's a shame how good the Blue Jays were a couple of seasons ago compared to now.
Patient	Yeah, I'm still not sure we should have got rid of Alex Anthopoulos.
Doctor	Yeah, that was the turning point, eh? Anyways, you're here to review your diabetes right?
Patient	That's right.
Doctor	How's the numbness in your toes?
Patient	The same. I'm used to it by now, and I'm grateful it's not getting worse.
Doctor	Okay, that's good. Let's keep you on the same dose of Metformin for now then we'll check your a1c again in three months, and then I'll see you back here after that.
Patient	That makes sense to me.

Table 2.10: Constructed example of a clinical dialogue with relevant and non-relevant information. Primary diagnosis: diabetes.

pertinent to ADD. Modality describes whether the extracted entity was actually experienced or not, and at what point in time. For instance, a medication could be current, past, prescribed, negative, or conditional ("If your headache gets worse, you can take Advil"). These attributes are important for distinguishing experienced symptoms and current medications from past events and negative or hypothetical events. In these cases, the context of the conversation, as well as time information, is crucial to recording the patient's information accurately.

We use the following modality categories: *actual, conditional, current, future, negative, option, past, possible, prescribed, referred,* or *none*. The pertinence categories are the main disease categories: *ADHD, COPD, depression, influenza, other,* or *none*.

A total of 475 conversations were annotated by a single physician, and inter-annotator agreement was calculated using DKPro Statistics [8] on 30 conversations which were annotated by two physicians. The agreement across all entity types is 0.53 Krippendorff's alpha (Krippendorff, 2004) and 0.80 F_1 (partial match).

Overall, the text quality of the transcripts is higher than the quality of the VA narratives, but there are still occasional misspellings or inaudible sections. Personal information was manually masked out by the transcribers and replaced with tokens such as "[PATIENT NAME]" or "[PLACE]".

2.6 Characteristics of datasets of medical narratives

In order to build appropriate models for these datasets, we need to examine the characteristics of each type of medical narrative. Table 2.11 shows some statistics of the datasets, including document length and the number of annotated events and timexes. We are also interested in how many events have an associated time phrase. Note that the absence of a specific timex associated with the event does not mean we have no temporal information about that event – many events have implied chronology or event–event relations, but these are not captured concretely by the annotations. However, event–event relations can

[8] https://dkpro.github.io/dkpro-statistics

often be inferred from context, and the listwise ranking annotations provide this information.

Dataset	**Documents**	**Annotated documents**	**Avg words per doc**	**Authors/speakers per doc**
Verbal autopsy (English)	14,468	691	83	1
Verbal autopsy (Hindi)	500	0	54	1
Clinical notes (THYME)	417	417	1,171	1+
Conversations (Verilogue)	800	475	1,614	2+
Dataset	**Total events**	**Events per doc**	**Total timexes**	**Events w/ timexes**
Verbal autopsy (English)	691	83	1,966	33.86%
Verbal autopsy (Hindi)	500	0	–	–
Clinical notes (THYME)	66,121	159	8,748	100%
Conversations (Verilogue)	10,386	63	4,646	1.45%

Table 2.11: Statistics of the 3 medical narrative datasets.

We can see that the density of events and timexes varies greatly between the datasets. In particular, the conversational dataset has very long documents (1614 words per conversation), but a comparatively low number of annotated events (63), and only a very small fraction of those events (1.45%) have associated time information. This makes sense given that these are informal dialogues, with a lot of conversational filler.

Clinical notes, on the other hand, are written by medical professionals with very limited time, and thus are very concise. Although clinical conversations are also restricted by time, they contain some amount of conversational pleasantries and irrelevant information which is omitted when the clinician writes up the clinical note. This leads to a much higher density of annotated events and time information in clinical notes. Verbal autopsies could be considered an intermediate level of formality, since they tend to be less concise than clinical notes, but still have a focused task of describing the events leading up to death. This results in the narratives including some irrelevant background information, but much less non-medical discussion than clinical conversations. While VA narratives can vary in length from a single sentence to several paragraphs, they are by far the shortest documents (average of 83 words per document vs. over 1,000 words in the THYME and Verilogue datasets).

In addition, VA narratives have a single author (the surveyor), while clinical notes can have multiple authors (different medical staff), and conversations include two or more speakers alternating. In the conversational data we have speaker labels for each utterance, while in the clinical notes it is not always clear who wrote each section of the note.

These stylistic differences are important because they affect what kinds of pre-processing needs to be done, as well as what kinds of tools and transfer learning we can use. For example, the VA dataset contains paragraphs of full sentences, although there are frequent grammatical inconsistencies and spelling errors. In contrast, clinical notes often contain sentence fragments and are more structured (either by time or by type of information). Consequently, models and resources (such as parsers) that work well for full sentences may not work well on sentence fragments and ungrammatical text. In addition, models trained on clinical text will learn to associate formal clinical terms with the cause of death, and if these models are applied directly to other types of documents such as verbal autopsies, they might not recognize symptoms or other entities that are described in a more informal manner.

Although all three types of documents contain natural language discussing medical concepts, the structure and style of these documents varies widely between datasets. However, given a similar task, we may be able to leverage these different types of medical narratives to increase the amount of data available for training machine learning models.

2.7 Conclusion

Medical narratives can vary greatly in structure, style, and content. We have seen three different types of medical narratives: verbal autopsies, clinical notes, and clinical conversations. In the past, most NLP work has focused on formal medical documents such as clinical notes, and it is not clear that such models can be easily applied to other types of medical documents.

In the following chapters, we conduct experiments using these different datasets and analyze how the performance differs depending on the type of data. We will also investigate transfer learning methods to see if we can leverage different types of medical narratives to improve training of machine learning models, particularly for smaller datasets.

Chapter 3

Event and time phrase extraction for medical narratives

3.1 Introduction

In order to focus on the most relevant information in a text document, we first need to identify medically relevant events, as well as time information, which can be used in downstream tasks such as temporal ordering. In this chapter we discuss methods for automatically identifying events and time expressions (timexes) in text.

Section 3.2 will discuss timex extraction methods, Section 3.3 will discuss event extraction methods, and Section 3.4 will present results of experiments using various timex/event extraction methods on the three medical narrative datasets. We also discuss some experiments with unsupervised event clustering, which can be used for event normalization.

3.2 Related work on identifying time expressions in medical narratives

First we will discuss time extraction methods for medical text, including rule-based, CRF-based, and neural information extraction models. Since there has been extensive work on temporal relation and event extraction, we will focus primarily on work in the clinical/health domain.

Many timex extraction models are developed for joint timex and event extraction, which will be discussed in Section 3.3. However, there are a few tools available that focus only on extracting time phrases.

One of the most popular rule-based time taggers is HeidelTime (Strötgen and Gertz, 2013), a publicly available toolkit for domain-independent temporal tagging and normalization. HeidelTime uses regular expressions along with pattern resources to identify time expressions in text. The pattern resources can be updated for the target language or domain. It also classifies the extracted temporal entities by type (DATE, TIME, DURATION, or SET). Jindal and Roth (2013) used HeidelTime along with rules for extracting admission and discharge dates from the i2b2 2012 dataset. Dates were parsed with the

JodaTime library[1]. Their system achieved .79 F_1 score on time phrase identification.

SUTime (Chang and Manning, 2012) also uses regular expressions to identify and normalize time expressions, and achieves higher recall but lower precision than HeidelTime. Viani et al. (2020) used an adapted version of SUTime to extract time phrases from mental health clincal notes from a London electronic health record (EHR) system. The time expressions were then normalized using post-processing rules, with an extra focus on normalizing duration values.

Apache cTakes (Savova et al., 2010)[2] is a publicly available information extraction system for clinical text that has been used in multiple hospitals. The system contains many different modules, including tokenization, POS tagging, sentence boundary detection, shallow parsing, and named entity recognition (NER). For time extraction, cTakes uses support vector machine (SVM) and conditional random field (CRF) sequence tagging models (Miller et al., 2015).

More recent machine learning-based information extraction systems often use a neural model for identifying all entities of interest, including both events and time expressions. These models are typically some kind of bi-directional long short term memory network (bi-LSTM) combined with a CRF prediction layer (such models will be discussed in Section 3.3.2). Although these models can also learn to identify time phrases, rule-based time taggers tend to have much higher precision, especially on medical text.

3.3 Related work on identifying events in medical narratives

The goal of event extraction methods is to identify spans of the text that refer to events. Sometimes these models also identify event attributes such as event type, aspect, and modality or negation.

Another level of complexity that is introduced by natural text is multiple references to the same event, often using different phrases. To address this, many models add a co-reference resolution step, linking each entity mention to an underlying entity concept, in order to prevent redundancy. However, this introduces an extra step in the model, which can cause compound errors. Although co-reference resolution can be helpful, and is often necessary in long documents with many references to the same event, we do not include it in this work for two reasons: the verbal autopsy narratives (our primary dataset) do not have a large number of duplicate entity mentions, and because the pipeline for performing temporal ordering and classification already has many stages and parameters, and adding co-reference resolution would only increase the complexity. Therefore we leave this as an avenue for future work.

Event extraction can be performed using rule-based or lexicon-based methods, or using a sequence tagging model. We review both in the following sections.

Lexicon-based tagging methods have been popular in the clinical NLP domain due to the large variety of medical terms that might need to be identified as events, along with the small size of most available medical datasets, although more recently neural models have been demonstrated to have comparable or superior performance (Wu et al., 2017). Such methods offer very high precision but low recall, and often do not generalize well to new datasets. They also require some manual curation of the lexical resources used for tagging, and in order to match entities that are similar but not exactly the same as the reference items, the search function may also require some regular expressions to match variations of the target terms.

Alternatively, automatically identifying event phrases in text can be framed as a supervised or un-

[1] https://www.joda.org/joda-time/
[2] https://ctakes.apache.org/index.html

supervised sequence labeling task, as it is similar to named entity recognition (NER). The goal of such models is to label spans of text (typically sequences of whole words) as being part of specific entities. Some sequence taggers also identify attributes of the identified entities.

Supervised sequence tagging is the task of learning to label each element in a sequence (such as each word in a sentence) from a set of labeled training data. A variety of models can be used for this task, such as hidden Markov models (HMMs) and recurrent neural networks (RNNs) (Graves, 2012). Items in the sequence are typically labeled with a BIO (begin-inside-outside) tagging scheme, where *O* represents unlabeled items, *B* represents the first token in a labeled entity, and *I* represents a token inside a labeled entity (Ramshaw and Marcus, 1995). This allows for the tagging of entities that consist of multiple words. The *B* and *I* tags can also include an entity type, such as *B-event* or *I-timex*.

To address the problems of time and event extraction in the clinical domain, there have been several shared tasks and challenges. Some of these challenges also included a temporal ordering or temporal relation extraction task, which will be discussed in Chapter 4.

The i2b2 2012 challenge (Sun et al., 2013a) included 3 tracks: timex and event extraction, temporal relation extraction, and end-to-end (timex/event and relation extraction). The time phrase extraction task also included normalizing the timexes to the ISO standard [3]. Participants were provided with 310 discharge summaries annotated according to the TimeML standard.

The Clinical TempEval Shared Tasks in 2015, 2016, and 2017 (Bethard et al., 2015, 2016, 2017) provided datasets and challenges for time and event extraction and normalization from clinical notes from the THYME dataset. Clinical TempEval 2017 added a domain adaptation task, where participants trained systems on colon cancer notes and tested on brain cancer notes. Because of the availability of annotated data and organized challenges, the TempEval datasets have been widely used for developing time and event extraction systems for the clinical domain.

3.3.1 Lexicon and rule-based methods

Many NLP and information extraction systems have been built for the medical domain, including MedLee (Friedman et al., 1994), which uses a semantic parser to extract and normalize medical entities, and ConText (Harkema et al., 2009), which identifies negation and temporal information using an extension of NegEx. cTakes contains lexicon-based NER modules for recognizing medical events, including mapping to UMLS concept IDs, and negation detection (which can be performed with either NegEx or the polarity module of ClearTK (Bethard, 2013)). MetaMap[4] is a UMLS metathesaurus which can be used to recognize clinical entities in text and map them to UMLS identifiers.

Reeves et al. (2013) developed Med-TTK, a modification of The Temporal Awareness and Reasoning Systems for Question Interpretation (TARSQI) Toolkit (TTK) (Verhagen et al., 2005), which identifies temporal relationships between events and produces TimeML output. Since TTK was originally developed using newspaper text, additional rules were added to Med-TTK to increase the performance on identifying and classifying time phrases in medical documents from .14 recall (TTK) to .86 recall (Med-TTK).

Xu et al. (2010) introduced MedEx, a system for identifying medication names and attributes in clinical notes and discharge summaries. MedEx uses regular expressions to identify medication information, followed by a semantic tagger and a Chart parser with a context-free grammar to categorize the infor-

[3] https://www.iso.org/iso-8601-date-and-time-format.html
[4] https://metamap.nlm.nih.gov/

mation, which can include the drug name, dose, route, frequency, and duration. The system achieved over .90 F_1 score for drug names, strength, route, and frequency, .88 F_1 for dose, and .74 for duration. However, the system was tested on a small dataset (50 discharge summaries and 25 outpatient clinical notes), and some lexical items and disambiguation rules were added manually based on the training set, which indicates that the system might not be generalizable to new datasets or other types of clinical narratives.

While lexicon-based methods may be a good option for specific domains with a lack of training data, they are difficult to transfer to new datasets and domains.

There are also some embedding-based methods such as MedCAT (Kraljevic et al., 2019), which uses word2vec embeddings to learn concept embeddings which are then used to identify concepts in the text via embedding similarity.

Event extraction methods can also use linguistic features along with rule-based models. Jindal and Roth (2013) leveraged the assumption that events that are mentioned close to each other are more likely to have similar attributes, since the events are likely to be related. They applied an event extraction model to the 2012 i2b2 dataset, achieving .87 F_1 score on identifying event spans, and .71 on identifying spans and attributes. The span identification model consisted of a shallow parser to identify constituents, which were then filtered by a set of rules. Each event's type, modality, and polarity were classified with SVM models, using lexical features along with clinical descriptors from Medical Subject Headings (MeSH)[5] and Snomed-CT (Rogers and Bodenreider, 2008). They performed inference using an integer quadratic program with constraints on the attribute values based on the proximity of events to one another.

ClearTK-TimeML (Bethard, 2013) used simple morpho-syntactic features such as stems, POS tags, sub-tokens, and the previous and next 3 tokens, using CRF, SVM, and logistic regression classifiers. They also classified time phrases into one of four time types: DATE, TIME, DURATION or SET. However, Styler et al. (2014) found that ClearTK-TimeML performed poorly on the narratives in the THYME corpus, with an F_1 score of 0.497 on identifying temporal phrases and 0.204 on identifying temporal relations (compared to 0.770 and 0.266 respectively on TempEval 2013 data).

Velupillai et al. (2015) used a ClearTK pipeline with features from cTakes and SVM classifiers to perform time and event extraction on the Clinical TempEval 2015 dataset, and achieved some of the best results in Clinical TempEval 2015.

Other methods such as logistic regression have been used. For example, the KUL system (Kolomiyets and Moens, 2013) for Clinical TempEval 2017, although the performance was lower than rule-based and CRF-based methods.

Along with widely used toolkits such as cTakes, ClearTK, MedTTK, there are a few commercial systems for clinical information extraction, such as Amazon Comprehend Medical[6]; however, details of the models are not available.

3.3.2 CRF and neural methods

A conditional random field (CRF) model is a special case of a Markov random field, where the graph models the conditional probability of a sequence of variables Y (the labels) given input variables X, where the labels depend only on neighbouring nodes. The chain-like graph structure of CRFs makes them well

[5] https://www.nlm.nih.gov/mesh/meshhome.html
[6] https://aws.amazon.com/comprehend/medical/

suited to sequence labeling problems. For example, MedTime (Lin et al., 2013) uses a CRF for event extraction along with an SVM for attribute classification, and HeidelTime for time extraction.

CRF models can also be combined with neural models. NCRF++ (Yang and Zhang, 2018) is a toolkit for creating neural sequence labeling models that use a CRF layer for inference. The models, which are implemented in PyTorch, can use a CNN or RNN layer for learning features at both the character- and word-level. The word-level layers can use pre-trained word embeddings. The features extracted from these neural layers are then used as input to a softmax or CRF inference layer with Viterbi decoding. This type of architecture allows the model to leverage the advantages of word embeddings and neural networks, while also including the power of the CRF's sequence decoding.

Yet Another Sequence Tagger (YASeT) (Tourille et al., 2018) is another sequence tagging toolkit that uses a similar bi-LSTM-CRF architecture implemented in TensorFlow (Abadi et al., 2015). It supports character and word embeddings in word2vec format and the parameters are customizable.

Recent work has shown that using neural sequence models along with large language model embeddings such as BERT or ELMo can provide good performance on a variety of clinical concept extraction tasks. Jauregi Unanue et al. (2017) used a bi-LSTM-CRF model with GLoVe word embeddings trained on MIMIC-II, character embeddings, and morphological features to identify clinical concepts and drug names in the i2b2 dataset, as well as DrugBank and MedLine (Herrero-Zazo et al., 2013). The bi-LSTM-CRF model performed better than an LSTM or CRF alone, achieving an F_1 score of .8335 on i2b2, .8838 on DrugBank, and .6066 on MedLine. The model also performed better than previous models on the same datasets.

Similarly, after training an ELMo model, Zhu et al. (2018) performed clinical concept extraction using a bi-LSTM-CRF model, which was trained on i2b2 dataset.

Si et al. (2019) compared traditional word embeddings (word2vec, fastText, and GloVe) trained on MIMIC-III against ELMo, BERT, and BioBERT for clinical concept extraction on the i2b2 2010 and 2012 clinical note datasets, and on clinical reports with disease concepts from SemEval 2014 and 2015. The best results (which became the new state-of-the-art) were obtained by starting with the BERT-Large model, continuing training on MIMIC-III, and then adding the fine-tuning layer (a bi-LSTM) to the model. This model achieved F_1 scores between 0.80 and 0.90 on the 4 datasets, outperforming all other embedding methods.

Zhao et al. (2018) trained three versions of a bi-directional RNN for drug name recognition and classification, using the following types of word embeddings as input: (1) fixed, pre-trained embeddings treated as constants, (2) variable word embeddings initialized with pre-trained embeddings but treated as learnable parameters updated by the model, and (3) randomly initialized word embeddings which are then updated by the model. The results showed that all types of RNN perform well, but the versions that considered word embeddings as learnable parameters produced better results.

While many methods perform event attribute classification as a second step after the event spans have been identified, several methods have attempted to jointly identify event spans and attributes using neural models. Bhatia et al. (2019) trained a model to jointly identify entities in medical text and their negation status. Using the i2b2 2010 dataset along with proprietary medical data, they trained a hierarchical encoder and conditional softmax decoder shared between the two tasks. The conditional model outperformed NegEx and previous NER models on both datasets, achieving .855 F_1 on NER and .905 on negation.

Du et al. (2019) used 3,000 annotated clinical conversations, along with 90k unannotated conver-

Model	Precision	Recall	F_1	CMSFA
Narrative only	.714	.705	.702	.914
Symptoms only	.727	.727	.723	**.938**
Narrative + symptoms	**.760**	**.750**	**.750**	.930

Table 3.1: Classification results with extracted symptoms.

sations, to train a model to identify symptom mentions and their status (experienced or not). Their span-attribute model consisted of a bi-LSTM sequence-to-sequence (Seq2Seq) model to identify symptoms, followed by a layer that computes a contextual representation and a distribution over possible symptom names and statuses. They trained the model using a multi-task learning setup with a joint loss function for symptom spans and status. The best model achieved .64 F_1 score for symptom and status extraction, using pre-training on the unlabeled data.

The results from Zhao et al. (2018) and Si et al. (2019) show that fine-tuning the embedding representation on in-domain data results in the best performance. Bi-LSTM models consistently performed well on the extraction task, and can be extended to identify event attributes in addition to event spans.

3.4 Experiments

3.4.1 Experiments with lexicon-based tagging of symptoms in verbal autopsy narratives

Joint work with Yoona Park

Given the popularity of lexicon-based methods for event tagging in medical narratives, as part of our early experiments we applied a lexicon-based symptom extraction model to the VA narrative dataset. We extract symptom terms from the text using lists of symptoms from BioPortal[7] and the Consumer Health Vocabulary (CHV)[8], as well as the physician key phrases from the training dataset.

We then compare the performance of our cause-of-death classification models using only the narrative, only the extracted symptom phrases, or both. The text is represented using word2vec embeddings (Mikolov et al., 2013), and the classifier is a convolutional neural network (CNN), with a kernel size of 5. This is an earlier version of the classification model that will be described in more detail in Chapter 5. The results are shown in Table 3.1.

The combination of both full narrative and extracted symptoms provides the best performance. This suggests that extracting important phrases such as symptoms can improve classification, but there also may be important context information in the body of the narrative that we do not want to discard.

3.4.2 Evaluation of sequence tagging models for event and time phrase extraction in different types of medical text

We train two models for event and time extraction on the verbal autopsy, THYME, and Verilogue datasets: a traditional CRF model (using scikit-learn (Pedregosa et al., 2011)), and a neural CRF model

[7] https://bioportal.bioontology.org/
[8] https://www.nlm.nih.gov/research/umls/sourcereleasedocs/current/CHV/index.html

Model	Exact Match			Overlap		
	Precision	**Recall**	**F_1**	**Precision**	**Recall**	**F_1**
EVENT						
CRF (pubmed vec + attributes)	**0.538**	**0.502**	**0.519**	0.928	**0.866**	**0.896**
NCRF++ (ELMo emb)	0.515	0.424	0.465	**0.944**	0.777	0.852
cTakes (rule-based)	0.218	0.188	0.202	0.735	0.636	0.682
TIMEX3						
CRF (pubmed vec + attributes)	**0.657**	**0.621**	**0.638**	0.852	0.807	**0.829**
NCRF++ (ELMo emb)	0.617	0.597	0.607	0.843	**0.815**	0.828
cTakes (rule-based)	0.162	0.025	0.043	**0.973**	0.148	0.257
HeidelTime (rule-based)	0.450	0.280	0.345	0.954	0.593	0.731

Table 3.2: Event and TIMEX3 identification results on 100 VA records from MDS, calculated with the anafora evaluation script used by Clinical TempEval 2017. Our models and best scores in bold.

Model	Exact Match			Overlap		
	Precision	**Recall**	**F_1**	**Precision**	**Recall**	**F_1**
EVENT						
CRF (pubmed vec + attributes)	**0.887**	0.875	0.881	**0.887**	0.875	0.881
NCRF++ (pubmed vec + num)	0.883	**0.882**	**0.883**	**0.887**	**0.886**	**0.887**
cTakes (rule-based)	0.437	0.335	0.380	0.596	0.457	0.518
TIMEX3						
CRF (pubmed vec + attributes)	**0.810**	**0.781**	**0.795**	0.810	0.781	0.795
NCRF++ (pubmed vec + num)	0.662	0.689	0.676	**0.813**	**0.848**	**0.830**
cTakes (rule-based)	0.423	0.240	0.306	0.501	0.286	0.364
HeidelTime (rule-based)	0.456	0.593	0.516	0.543	0.708	0.614

Table 3.3: Event and TIMEX3 identification results on the THYME test set, calculated with the anafora evaluation script used by Clinical TempEval 2017. Our models and best scores in bold.

using NCRF++. For the CRF model we use the publicly available word2vec model trained on PubMed data[9], and for the NCRF++ model we use ELMo embeddings also trained on PubMed. For comparison we apply a rule-based system (cTakes) to each dataset, and for time extraction we also run HeidelTime.

Because the Verilogue dataset is currently too small to train a good model using the annotated data, we use a lexicon-based tagging model for event extraction using medical ontologies from BioPortal.

We evaluate the event and time extraction models using precision, recall, and F_1 score, both of exact span matches to the reference entities, and overlap with the reference entities. Table 3.2 shows event/time extraction results on the verbal autopsy data, and Table 3.3 shows the results on the THYME dataset using various models.

Table 3.4 shows the results of event and time tagging on the Verilogue clinical dialogue dataset. In this case, the event score is a weighted average of the different types of entities that are identified by the tagger: *anatomical location*, *diagnosis*, *investigation/therapy*, *medication*, *referral*, and *sign/symptom*. The TIMEX3 scores are reported separately. Words or phrases in the utterance text that are at least 3 characters long and match a term in the reference lexicon are assigned the corresponding tag. For time phrase tagging we use HeidelTime. Overlapping tags are not allowed.

The NCRF++ tagger is the same model as described above, except that the labels are the specific

[9] https://bio.nlplab.org/

Model	Exact Match			Overlap		
	Precision	**Recall**	**F_1**	**Precision**	**Recall**	**F_1**
EVENT						
Lexicon (no training)	0	0	0	**.721**	**.922**	**.807**
NCRF++ (ELMo emb)	**.188**	**.022**	**.039**	.348	.044	.077
TIMEX3						
NCRF++ (ELMo emb)	.009	.003	.005	.844	.253	.389
HeidelTime	**.897**	**.593**	**.714**	**.976**	**.637**	**.771**

Table 3.4: Event and TIMEX3 identification results on the Verilogue test set (50 conversations), trained on 425 annotated conversations. Our models and best scores in bold.

entity types (*medication*, *diagnosis*, etc.), instead of just EVENT.

We find that overall the trained CRF model performs the best, although the NCRF++ model is sometimes competitive and performs better on the THYME dataset in terms of the overlap metrics. In all cases, the trained models perform better than the rule-based cTakes pipeline. For the Verilogue dataset, the lexicon-based model performs the best for event extraction, and HeidelTime performs the best for timex extraction; therefore we use these two models for the event/time extraction step for the Verilogue data. For the other datasets we use the CRF models.

3.4.3 Unsupervised event phrase clustering

The material in this section is based on previously published work (Jeblee et al., 2018).

Several issues can affect our ability to extract useful event phrases from the text. We might not have sufficient annotated data or lexicons to train a model for supervised tagging. In addition, the extracted terms often have a lot of variation in phrasing, as there are many different ways of describing the same symptoms. For example, "abdominal pain" and "pain in the abdomen" describe the same symptom.

In order to reduce variability and standardize the event phrases, we perform unsupervised clustering of physician-labeled key phrases from the MDS dataset. Each record includes a list of key terms from each coding physician. These are often phrases directly from the narrative text, but they could also be a physician's interpretation of the descriptions in the narrative.

We use the k-means algorithm with Euclidean distance (using scikit-learn) to group the key phrases from the MDS training data into 100 clusters. Each key phrase is represented as the average of the word2vec embeddings of each word in the phrase.

For new, uncoded records, we will have only the narrative and therefore will need to predict the key phrase clusters. For evaluation, because the clustering is unsupervised and we have no gold standard mapping of key phrases in the test data to clusters, we assign each test key phrase to a cluster using a k-nearest neighbor classifier ($k = 5$). We treat these clusters as the "true" labels for evaluating the model.

In order for these clusters to be useful to physicians, we need a text label for each in order for it to be clear what each cluster represents. We could simply take the most frequent key phrase in each cluster as the label, but many key phrases are variations of the the same idea, or have extra details in them, so the most frequent phrase might not be the most representative. Therefore, to get a text label that is representative of the cluster, we choose the key phrase that is closest to the center of the cluster in

vector space. However, there are some key phrases which are much longer than average. Since the vector representation of each phrase is the average of the word embeddings, a phrase with many words is more likely to be closer to the center. Also, we want to favor shorter labels that are general enough to describe all the members of the cluster. Therefore we introduce a length penalty: the score used for selecting the label phrase is the distance of the phrase embedding from the center of the cluster multiplied by the number of words in the phrase. This gives us cluster labels that are usually one or two words.

Some of the key phrase clusters learned by the model are shown in Table 3.5. We can see that some clusters are clearly capturing specific symptoms, such as "cough" and "chest pain". However, the clustering doesn't always capture the type of similarity we're interested in, such as the "breathing difficulty" cluster, which captures phrases containing "difficulty", although these often represent different symptoms.

Label	**Key phrases in cluster**
cough	cough, cough with sputum, cough with phlegm, had sputum cough, …
rigours	fear, sudden chest pain one day and died in short while, h/o headache, epileptic, …
h/o chest pain	sudden chest pain, occasional chest pain, sudden pain in middle of chest, …
breathing difficulty	difficulty in eating, difficulty in urination, …

Table 3.5: Examples of key phrase clusters with generated labels ("h/o" means "history of").

The main difficulties of the unsupervised clustering model are choosing the parameters and evaluating the clusters. It is often not clear a priori how many clusters should be generated to best represent the data. Without labeled data, is also difficult to evaluate the resulting clusters to determine which model and parameters are the best.

To estimate the quality of an unsupervised clustering, we use the Calinski-Harabasz Index (CHI) (Caliński and Harabasz, 1974), which measures the ratio of the between-cluster variance to the within-cluster variance, so a higher number indicates that the clusters are more internally compact and separated from each other. The 100-cluster model has a CHI of 1713, and we found that using more clusters produced a lower CHI and more instances of multiple clusters that described the same symptom. Using too few clusters results in too many different concepts being grouped together.

Given a dataset of manually grouped keywords, we could conduct a supervised evaluation of the clustering by comparing to the human-clustered data. Such data could also be used for supervised or semi-supervised training of clustering models. However, in the absence of labeled data, unsupervised clustering can be helpful for term normalization.

3.5 Conclusion

Event and time extraction methods can have reasonably high accuracy, especially when using in-domain resources. Although models in the medical domain can benefit from comprehensive medical lexicons, neural models can also be trained from annotated data, which provides more flexibility and adaptability. Identifying these event and time phrases accurately is crucial for downstream tasks such as temporal ordering and classification. It also helps to identify important parts of the text to reduce noise from irrelevant information.

We have shown that we can train a fairly simple sequence-based model to identify events and times with over .80 F_1 score, and that the trained models outperform rule-based solutions such as cTakes. A good sequence tagger model can learn from in-domain examples to identify phrases of interest, and this is a much faster option if the appropriate training data is available. Despite the extensive clinical knowledge encoded in cTakes, our out-of-the-box model performs better on the target dataset. We use the models described in this chapter as the event/timex extraction step prior to the temporal ordering and classification models described in the following chapters.

Chapter 4

Temporal ordering of events in medical narratives

4.1 Introduction

In healthcare, extracting information from unstructured text can provide insights for medical professionals, and can also be useful for automating certain tasks. For example, identifying symptoms and other medically relevant events in a document can aid in disease or cause-of-death classification.

However, context for these events is crucial, especially temporal context. Temporal information such as order, duration, and co-occurrence of symptoms can be an important factor in diagnosis, but many current approaches ignore this information. When temporal information is used, it is typically encoded as pairwise relations between events, or between events and time phrases, with defined sets of relations, such as BEFORE / AFTER / OVERLAP or Allen's 13 interval relations (Allen and Ferguson, 1994). However, not all pairwise relations are defined, and this requires annotating and classifying n^2 pairs of events, many of which have no defined relation. Also, the text may be too vague to determine the boundaries of every event.

In this work, we focus on grouping and ordering event phrases from medical narratives in a listwise fashion, according to their time of occurrence. We investigate two models: a linear ordering (a simple ordering of one event after another) and a grouped ordering, which puts events in chronological order while allowing simultaneous events to be grouped together. For our purposes, "simultaneous" means that two events have roughly the same start time, although they may end at different times. For each document, the input to a model is a set of events, some of which have corresponding time phrases, and the output is a rank value for each event indicating its position in the timeline. We evaluate these models on different types of medical narratives, including verbal autopsy reports, clinical notes, and clinical conversations.

4.1.1 Temporal relations in medical text

Datasets and annotation for temporal relations

The most commonly used annotated datasets for temporal relation extraction are the TimeBank corpus (Pustejovsky et al., 2006) and the AQUAINT corpus (Graff, 2002), which are both datasets of news

articles annotated with TimeML. For the medical domain, the THYME corpus (Styler et al., 2014) and i2b2 corpora (Sun et al., 2013a) are the most widely used for clinical temporal information extraction.

In Section 2.2 we discussed the TimeML annotation schema, which is used for the THYME dataset, and the Simple TimeMl schema used for the VA dataset.

In Section 3.3 we also discussed several challenges and shared tasks that had time and event extraction tasks. Many of these challenges also included a temporal ordering task. The 2007 TempEval challenge (Verhagen et al., 2007) provided annotated TimeBank for three temporal relation identification sub-tasks: within-sentence time–event relations, DCT–event relations, and event–event relations between the main events in adjacent sentences. Systems were scored based on pairwise comparisons within each task. The 2012 i2b2 NLP Challenge (Sun et al., 2013b) included three temporal relation tasks: event–time relation extraction, event–event relation extraction, and end-to-end extraction. The Clinical TempEval Shared Task in 2015 (Bethard et al., 2015), 2016 (Bethard et al., 2016), and 2017 (Bethard et al., 2017) focused on temporal relation extraction for clinical text, including event–time and event–DCT relations.

4.1.2 Pairwise classification vs. listwise ordering

The material in this section is based on previously published work (Jeblee and Hirst, 2018).

Temporal relation extraction is typically framed as a pairwise classification problem: generate all pairs of events in a document, and then determine what type of temporal relation exists between each pair, if any. The major problem with this approach is that the vast majority of event pairs have no relationship, or the relation between them is unknown. This results in an unbalanced classification problem, and there is no guarantee that the predicted pairwise relations are consistent with one another. Because of the sparsity of annotated long-distance relations, many pairwise classification models have been limited to events mentioned within the same sentence or within some small window of the text. It is often difficult for humans to analyze the relations from an entire document quickly, especially when they are inconsistent.

In addition, for pairwise models we must decide whether to generate the inverse relation for every event pair, or to flip some relations to reduce the number of classes (e.g., transform AFTER relations into BEFORE relations). In contrast, a document-level list inherently captures pairwise relations between all events in the document, regardless of whether or not they appear in the same sentence. Thus, we choose to represent the events as a temporally ordered list (hereafter referred to as a listwise ordering) instead of as temporal relations between pairs of events or time expressions (pairwise ordering).

However, since pairwise relations often capture relationships that are more complex than just BEFORE, AFTER, and OVERLAP, we add time information to the events in the reference list when available. This information includes event start, end, and overlap times, based on the annotated relationships to time phrases in the text. For this work, we sort the list by event start time, but in principle we could sort by end time or examine event overlaps. All time information can be either exact, relative (before or after a certain time), or unknown.

Since the primary goal of this work is to produce timelines that are easy to interpret and also useful for doing downstream classification, a simple linear ordering (with grouping) is better suited to this task than pairwise ordering. Especially when working with data where detailed temporal information is not always available, it can be useful to take a step back and try to capture a global picture of chronology, rather than more fine-grained and error prone relations. We choose models with less specificity in favor

of usefulness to both downstream models and human users.

4.1.3 Our contributions

In this work, we train several models to group events from medical narratives, and order those groups chronologically. We compare the performance of these models on both formal and informal medical narratives (in the form of clinical notes, verbal autopsy narratives, and clinical conversations), including several cross-dataset experiments.

Our contributions are as follows:

- This work adapts and improves the Set2Seq model of Vinyals et al. (2016) and Logeswaran et al. (2018) to make it suitable for grouped listwise ordering. The novel mechanisms include a mask that forces the model to choose each item exactly once, a grouping threshold that allows items to be assigned the same rank, and Gaussian smoothing of the target probabilities to reduce the loss for placing items closer to the correct timestep (rank). To our knowledge, this is the first application of a Set2Seq model for temporal ordering.
- We demonstrate utility by applying it to the task of temporal ordering of events in both formal and informal medical narratives.

4.2 Related Work

Temporal ordering is a specific task, but it is rooted in several common problems such as relation extraction and ranking. We can think of temporal ordering as finding temporal relations between individual events, or as a ranking problem where the criterion for ranking is time of occurrence.

In this section, we review relevant past work, including temporal relation extraction (focused on the medical domain), container relation extraction, models for handling set-based inputs, models for learning to rank (including sentence ordering), and finally, previous work on our target task: listwise temporal ordering.

4.2.1 Temporal relation extraction

Leeuwenberg and Moens (2019) presented a survey of temporal ordering methods, starting with Allen's relations. They emphasize the incomplete nature of temporal information in most documents – therefore systems must be able to handle underspecification of temporal information. They also note that computing the transitive closure of graphs of pairwise temporal relations can be computationally intractable. This is another reason why predicting listwise relations is a more efficient way of modeling global temporal information.

Mani et al. (2003) trained a statistical classifier to learn time anchor relation rules for events in news articles, and used the predicated relations to generate a partial ordering of events, achieving .754 F-score. Mani et al. (2006) trained a maximum entropy classifier to link times and events in news text using human-annotated TimeML event features. They also computed the temporal link closure to include inferred relations in training, which improved TLINK classification accuracy.

Chambers and Jurafsky (2008) learned narrative event chains from news text using an SVM classifier (BEFORE / OTHER) and evaluated the ordering using a coherence score, which is the sum of the correct

pairwise relations, weighted by confidence. Their model achieved 75.2% ordering accuracy according to this score.

Some work has focused on examining a graph of pairwise relations, where each event or time is a node, and the temporal relations are the links between nodes.

Chambers et al. (2014) introduced CAEVO, a cascading event ordering architecture, which uses a sieve-based approach for temporal event ordering on the TimeBank-Dense corpus. The model uses multiple rule-based and machine learning classifiers to add labels to the temporal graph. The system outperformed previous models from the TempEval-3 Challenge (Uzzaman et al., 2013).

Ning et al. (2017) generated a directed temporal graph representing TLINKs in the TempEval-3 datasets using SVM and perceptron models. The system was evaluated using temporal awareness precision and recall (UzZaman and Allen, 2011), which compares a predicted temporal graph to a reference graph, including transitive closure.

Denis and Muller (2011) used integer linear programming (ILP) to perform inference on a graph of relations represented as temporal endpoints instead of intervals, using the full Allen relations instead of just BEFORE / AFTER (as Bramsen et al. (2006) did).

Kolomiyets et al. (2012) introduced a graph-based dependency parser for representing a partial ordering of events in children's stories as a dependency tree. While this method still relies on links between pairs of events, it produces an overall ordering for the whole document, which is often not possible with purely pairwise classification methods due to link conflicts.

Zhou and Hripcsak (2007) reviewed temporal reasoning work in the medical domain, including temporal reasoning theory, applications to the medical domain, and issues such as temporal granularity, uncertainty, medical grammar, and implicit temporal information.

Nikfarjam et al. (2013) developed a hybrid model for TLINK prediction, using parse dependencies of simplified sentences to generate a temporal graph in combination with an SVM classifier. The model was evaluated on the i2b2 NLP corpus of 310 discharge summaries and achieved a precision of .76 (the highest of systems in the i2b2 TLINK track) and recall of .54.

Leeuwenberg and Moens (2017) used ILP with a structured perceptron to predict event–time relations as well as event–DCT relations. They evaluated the model on the THYME corpus with temporal closure and found that learning document-level relations with global features yielded significantly better performance.

One of the main issues with pairwise models is that they do not guarantee consistent relations. Bramsen et al. (2006) attempted to address this problem by disallowing cycles in the ordering graph. They presented a method for temporal segmentation and ordering using graph constraints over pairwise relations between temporal segments. They used a binary pairwise classifier followed by global inference to find the optimal ordering graph that satisfies the constraints. The ILP inference achieved the best results at 84.3% accuracy on a dataset of medical case summaries.

Derczynski (2016) investigated the effects of the temporal relation type schema on the performance of temporal relation extraction models. He showed that while a simple set of relations such as BEFORE / AFTER / OVERLAP is faster to annotate and easier for models to learn, it is not as expressive as a larger set of interval or semi-interval relations. On the other hand, we hypothesize that more complex relations might be too specific for documents like VAs where time references are often vague or approximate, especially when referring to durations. As the specificity and expressiveness of a relation set increases, so does the difficulty in annotating and learning those relations, and so it is important to select the right

level of relation complexity.

Lee et al. (2018) used an SVM classifier to identify direct temporal relations (within-sentence relations between times and events that have limited syntactic distance), using features such as event attributes, POS tags, dependency parses, and semantic role labels. The system was evaluated on the i2b2 2012 corpus and achieved .6377 F_1 score. Direct temporal relations were proposed because they represent 89% of all within-sentence event–time relations, however this only handles a subset of relations, and only within the same sentence, and thus suffers from the same drawbacks as other pairwise and within-sentence methods.

Lin et al. (2018) used an RNN model with word embeddings using self-training (the model is trained on labeled data, then applied to unlabeled data and retrained on its own output) for predicting CONTAINS relations in the THYME dataset. The model used word embeddings learned on the MIMIC-III dataset (Johnson et al., 2016) and surpassed the previous state-of-the-art.

Lin et al. (2019) used a BERT-based model to extract CONTAINS relations within and across sentences in the THYME corpus. They used gold-standard event and time annotations, and looked for candidate pairs within a window of 60 tokens. Among the different BERT models (BERT (Devlin et al., 2018), BioBERT (Lee et al., 2019a), and a BERT model trained on MIMIC-III), the BioBERT model performed the best in terms of F_1 score (.684 for colon cancer notes and .565 for brain cancer notes). Although cross-sentence accuracy was lower than within-sentence accuracy, most previous models were limited to within-sentence relations only. However, this work still fails to capture long-distance relations.

Najafabadipour et al. (2020) used rule-based models to link events and TIMEXes in Spanish clinical notes, including section and document creation times, and then generate a medical timeline by resolving event co-references. The system achieved .89 precision, but recall was not reported. Although this model is primarily rule-based, it is one of the few models for temporal ordering in languages other than English.

4.2.2 Container relation classification

Miller et al. (2013) created a system that performed binary classification of pairs of manually annotated temporal expressions and events from the THYME corpus as having the CONTAINS relation or not, using SVMs with tree kernels and bag-of-words features along with semantic features such as the event modality and the type of time expression. The system was evaluated on a set of 78 clinical notes, achieving a maximum F_1 score of 0.737.

Tourille et al. (2017b) used a linear SVM model to identify narrative container relations (CONTAINS, IS-CONTAINED, or NO-RELATION) for medical events in clinical notes. They extracted lexical features from the THYME and MERLOT (Campillos et al., 2018) corpora using Apache cTAKES[1], an NLP information extraction toolkit. The MERLOT corpus contains clinical notes in French with UMLS concepts and temporal relations annotated. The system was also tested with word embedding features from word2vec (Mikolov et al., 2013), but these had a negative impact on performance. The system achieved 86.8% accuracy on the DCT relation classification task for English, and 0.751 F_1 score on identifying CONTAINS relations.

Tourille et al. (2017a) used a bidirectional LSTM model with word and character embeddings to classify narrative container relations in the THYME corpus. They created separate models for classifying intra-sentence relations and inter-sentence relations (up to three sentences apart). Their model achieved

[1] http://ctakes.apache.org/

.605 F_1 score with only character and word embeddings, and .613 with the addition of features from the gold-standard annotations.

Tourille (2018) used a feature-based approach to classify narrative container relations in the THYME corpus and the MERLoT corpus, a dataset of clinical notes in French (Campillos et al., 2018). On the THYME dataset their best model achieved an F_1 score of 0.538 on classifying container relations using lexical features.

Galvan et al. (2018) applied a tree-based LSTM model with dependency information to the Clinical TempEval 2016 dataset, for temporal relation extraction including container relations, and achieved state-of-the-art performance. The text was represented with word2vec embeddings trained on PubMed. However, the model was limited to relations within the same sentence. They discovered that the system had lower performance on OVERLAP relations, and a high number of false negatives, which resulted in fairly good precision but low recall.

Dligach et al. (2017) aimed to move away from a large amount of engineered features by using only text with simple event and time phrase tags as input to a CNN and an LSTM model for container relation extraction. While the model achieved an F_1 score of .700 for event–time relations, it achieved only .515 on event–event relations.

However, all of these methods suffer from the problems of pairwise classification, particularly the large number of negative instances. Most also only identified relations within the same sentence, or within a fixed window of text, and therefore cannot capture longer-range relations.

4.2.3 Set input methods for neural networks

Lee et al. (2019b) proposed the Set Transformer architecture, which uses multi-head attention and pooling to approximate a permutation invariant function over set inputs. The encoder layer consists of induced set attention blocks (ISAB), which compute multi-head attention over the input set and a set of learned inducing points. The results are then used to compute attention over the input set again, which helps to reduce the runtime complexity of the set attention block from $O(n^2)$ to $O(nm)$, where m is the number of inducing points. The decoder network then aggregates the features using pooling by multihead attention (PMA).

Vinyals et al. (2016) introduced the set-to-sequence (Set2Seq) model, a modification of a sequence-to-sequence (Seq2Seq) model for inputs that are sets instead of sequences. That model included three steps: read (create an embedding of the input set), process (run the encoder RNN with attention over the memory block), and write (the decoder reads in the hidden state and outputs items from the set one-by-one, using a pointer network (Vinyals et al., 2015)). They demonstrated that the order of input to a traditional Seq2Seq model affects the output, and proposed modifications for treating the input as a set. Logeswaran et al. (2018) used a Set2Seq model for sentence ordering and coherence modeling on sentences from academic paper abstracts, and achieved better accuracy than traditional RNN and Seq2Seq models. In our work, we adapt the Set2Seq model for the grouped temporal ordering task.

Some other set input methods for neural network models include Deep Sets (Zaheer et al., 2017) and SWARM (Vollgraf, 2019). Zaheer et al. (2017) presented a definition of permutation invariant and permutation equivariant functions. They proposed neural network models called Deep Sets that are permutation invariant for set input, namely by summing the input representations and then applying a non-linear transformation. They tested the Deep Sets models on a text concept set retrieval task and achieved better results than latent Dirichlet allocation (LDA) and nearest neighbor models when given

sufficient training data.

Vollgraf (2019) defined a set-equivariant function as any function which outputs the same individual-level results for a set of data regardless of the order in which the items are presented. They introduced a SWARM cell – a modified LSTM cell that includes a set-level input. In a SWARM layer, the weights of the cell are shared among all cells, and the output is pooled over all entities. They compared the SWARM performance to similar architectures using SetTransformer, SetLinear, or LSTM layers for an amortized clustering task, and found that the SWARM layers performed the best. However, their evaluation was performed on numerical and image data, and we found in our experiments that the SetTransformer layer performed better than the SWARM layer for temporal ordering.

4.2.4 Ranking methods for ordering

Alternatively, we can use a ranking model to find a relative ordering of events, especially when absolute time information is not available. This type of ordering is common in information retrieval models, where the goal is usually to rank items by relevance, often based on a search query.

Cao et al. (2007) introduced ListNet, a listwise ranking model for information retrieval, using a neural network with a listwise loss function. They introduced two models: permutation probability and top-k probability, and showed that the listwise approach performed better than pairwise ranking methods. For the clinical domain, Jeblee and Hirst (2018) used ListNet to order events in the THYME dataset of clinical notes. The reference listwise orderings were generated from human-annotated pairwise temporal relations.

Ranking models can use a variety of loss functions, depending on the task. ListMLE ranking loss was introduced by Xia et al. (2008), who also investigated algorithms for learning to rank. ListMLE maximizes the sum of the likelihood function over all training examples. They found that compared to previously used loss functions such as cross-entropy and cosine loss, likelihood loss performed better on sample ranking tasks. This is the loss function we will use for our work. ListMLE maximizes the sum of the likelihood function over all training examples:

$$\sum_{i=1}^{m} logP(y(i)|x(i);g)$$

Xia et al. found that neither cosine nor cross-entropy losses are sound for ranking tasks, but ListMLE is. ListMLE also outperformed ListNet and RankCosine on both synthetic and real-world data.

Kumar et al. (2019) performed experiments with pointwise ranking loss (MSE), pairwise margin ranking loss (which examines the scores of pairs of consecutive sentences), and several listwise ranking loss functions. Because for temporal ordering the accuracy of the entire list is important, not just the top of the list, we opt to use ListMLE as implemented by Vollgraf (2019)[2].

Ranking models can also be used to find the correct order of sentences. Unlike information retrieval tasks, it is equally important to order all of the sentences correctly, not just the first few. This is the same goal we have in temporal ordering tasks.

Kumar et al. (2019) applied listwise ranking methods to the task of ordering sentences from a paragraph, using various academic paper abstracts and a BERT-based transformer model (Devlin et al., 2018). They used BERT to encode each sentence, followed by a paragraph-level Transformer layer. In-

[2] https://github.com/zalandoresearch/SWARM

stead of a pointer network for the final output layer, they used a plain feed-forward layer that outputs a score for each sentence and then sorts the sentences by score. Their model surpassed the state-of-the-art for most datasets, and the best performance was achieved with the ListMLE ranking loss. We use a similar model, but with ELMo embeddings (Peters et al., 2018), for temporal ordering.

A few ranking models have been applied to medical event data. Raghavan et al. (2012) used a ranking model to induce a partial temporal ordering of events in medical narratives. The model was trained on event chains, where each event is represented by features that include event polarity, UMLS semantic category, the type of narrative, and the position in the narrative. Each event was assigned to one of six coarse time bins (relative to admission time), which were learned with a conditional random field (CRF) and then used as a feature for the ranking model, along with the event's time span. The system learned Allen's temporal relations between pairs of events using SVM-rank (Joachims, 2006), which produced a ranking of events relative to the patient's admission date, learned by minimizing the fraction of swapped pairs. Raghavan et al. compared their ranking model to a pairwise classification of events into temporal relations with an SVM, and found that the ranking model achieved higher accuracy (82.16% vs. 71.33%). In contrast, they also tested both models on TimeBank and found that unlike on medical narratives, classification performed better than ranking (0.639 vs. 0.544) for news text. This shows that ranking models are perhaps better suited to temporal ordering for medical narratives than pairwise classification. Also, the ranking model implicitly learns the transitive relationships between events and produces a consistent overall ordering.

4.2.5 Listwise temporal ordering

While most previous work on temporal ordering, especially in the clinical domain, has focused only on pairwise temporal relations, some recent work has addressed listwise ordering.

In order to bypass the problems with pairwise classification, some work focused on listwise ordering. Leeuwenberg and Moens (2018) used a bi-directional long short-term memory (LSTM) network to predict listwise timelines by predicting a start value and duration value for each event, from which a listwise timeline can be derived. They trained the model on a dataset of annotated news articles, using a variety of loss functions, including a relative ranking loss function. The events were represented with GloVe word embeddings (Pennington et al., 2014) and part-of-speech (POS) tag embeddings. However, this requires events for which an interval can be predicted, which is not always the case with clinical text. In our work we similarly aim to order events by start time, but we have not yet predicted durations, as this information is often not determinable from less formal narratives such as verbal autopsies.

In addition, their method used loss functions and evaluation metrics that use pairwise relations to construct a relative timeline ordering. In contrast to Leeuwenberg and Moens, we abstract the timeline even further by training and evaluating directly on the global ordering. Although we provide an evaluation metric based on pairwise orderings after the fact, we train and evaluate the timeline on the global ranking values, which are simpler to annotate and allow us to use a more straightforward loss function. Since we use different annotations and evaluation, our results are not directly comparable.

For the clinical domain, Jeblee and Hirst (2018) used ListNet to order events in the THYME dataset of clinical notes. The reference listwise orderings were generated from human-annotated pairwise temporal relations. We build upon this work and extend it to annotated verbal autopsy records.

Although most methods have focused on events from a single document, events can be pooled from multiple notes pertaining to the same patient. For example, Raghavan et al. (2014) used a weighted

finite state transducer model to combine temporal event sequences from multiple medical narratives into a single timeline.

4.3 Data

We apply several temporal ordering models to all three of our medical datasets: verbal autopsies (MDS), clinical notes (THYME), and clinical dialogues (Verilogue).

For the THYME dataset, we use the provided train/dev/test split with gold-standard annotations, including temporal relations to the document creation time (DCT). The training set has 208 records, and the test set has 106. We use English language narratives from the Million Death Study (Westly, 2013; Jha, 2014; Gomes et al., 2017), and we have manually annotated 600 records of adult deaths for training and 100 for testing, using a simplified version of TimeML.

Events in the THYME dataset are almost always just a single word. This is somewhat of a limitation as, in some cases, a longer phrase might better capture the details of the event, such as *pain in the left side of the chest* instead of just *pain*. However, to our knowledge, all of the currently available clinical datasets with temporal relation annotations have events annotated as single words.

The THYME dataset does not have listwise ordering annotations, but it does have pairwise temporal relations [3], so we must convert the pairwise annotations to grouped listwise orderings. We use the graph of pairwise relations to generate a listwise ordering for each record, following the procedure of Jeblee and Hirst (2018). This results in an integer rank value for each event.

In the VA annotations, events are often phrases of multiple words, and many time phrases are relative to the time or date of death, whereas the THYME dataset typically has more absolute dates, and events are only one word.

In the THYME test set, there are an average of 5.231 events per group (i.e., that have the same rank), whereas in the VA test set the average is 1.447 events per rank. Although we can approximate the ranking by outputting a single event at each rank, especially for the THYME dataset, it makes more sense for the model to rank groups of events. Moreover, the THYME events frequently have an associated time phrase, whereas many events in the VA dataset do not, and temporal ordering must be inferred from context, a task that is often difficult even for humans.

The Verilogue dataset is somewhat different from the other two because not every event is ranked. The goal of the conversational dataset annotation is to extract important information that will go into the generated clinical note, and therefore many events that are annotated in the conversation are not included in the final note, and only a subset of the included events have associated time information that allows them to be ranked. The ranked events have an average of 2.478 events per rank, but only 1.45% of the events have associated time information.

The Verilogue dataset also includes a number of different entity types (discussed in section 2.5). For temporal ordering we consider all entity types except TIMEX3 as events.

[3] As far as we know, there are no publicly available clinical datasets with listwise ordering annotations.

4.4 Embedding events and time phrases

4.4.1 Embedding event phrases

Once events and time expressions have been identified, we must choose a way to represent them in numerical values in order to use them as input to neural network models.

Extensive work has been done on representing words and phrases as distributed representations, known as word embeddings (such as word2vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014), fasttext (Bojanowski et al., 2017), and ELMo (Peters et al., 2018)). For representing the event phrase, we can use any existing word embedding model, or train our own. In the following work, we use ELMo (Peters et al., 2018) with the available weights trained from PubMed [4] to represent each word as a 1024-dimensional embedding.

We then use a denoising autoencoder based on the model described in (Oshri and Nishith, 2015) to generate a single fixed-length vector representation of each event phrase. This model is an encoder-decoder architecture, similar to ones that are typically used for machine translation, except that the decoder reconstructs the original input text. The encoder operates over the word embeddings of the input sequence, which in this case is the event phrase plus some previous and next words for context (for these experiments we use the 5 previous words and the next 5 words, unless otherwise specified). The model is trained with cross-entropy loss of each output token from the decoder, and then only the encoder layer is used for the rest of the temporal ordering task.

We found that updating the weights of the autoencoder layer during the training of the temporal ordering model improved the overall temporal ordering performance.

4.4.2 Embedding time phrases

Time phrases are somewhat more difficult to represent because they have an inherent value that might not be captured by a contextual word embedding model. While some time expressions, such as dates and times, can be represented with an ISO time value, there are many vague and relative time expressions that have no specific numerical value.

Jiang et al. (2019) generated numeral embeddings from a set of prototype numerals using skip-gram word embeddings. These embeddings can be used in conjunction with the original word embedding model, and can improve downstream tasks.

Cai et al. (2018) included temporal scope information in medical concept embeddings from EMR data. They used a modification of the CBOW algorithm using nearby clinical concepts as context, along with a time-aware attention to learn a soft (in the form of non-uniform attention over time periods) temporal scope for each concept. These embeddings outperformed traditional word2vec and GloVe embeddings on clustering and nearest neighbor tasks.

A few recent works have attempted to learn embeddings specifically for timexes. Lin et al. (2017) represented time expressions from THYME clinical notes as single vectors for temporal relation classification (focused on CONTAINS relations) using a CNN model. They found the best results for event–time relation classification when including the time types in the time representation. However, they only examined pairwise within-sentence relations.

Goyal and Durrett (2019) trained a timex embedding model by randomly generating pairs of temporal

[4] PubMed weights for ELMo available at https://allennlp.org/elmo.

Dataset	Train	Test	P	R	F_1
VA time pairs	9908	1054	.738	.731	.733
THYME time pairs	10,000	1000	.899	.899	.899

Table 4.1: Timex pair classification results using the timex embedding model. We limit the THYME pairs to 11,000 total to reduce training time.

expressions and training an LSTM model to classify their order (BEFORE, AFTER, or SIMULTANEOUS). In a temporal ordering task, the timex embeddings produced a slight improvement in ordering accuracy over the plain ELMo embeddings.

We train a similar model using pairs of time phrases extracted from the listwise temporal ordering annotations on the VA dataset. Figure 4.1 shows the architecture of the time embedding model from Goyal and Durrett (2019) that we adapt for use in our temporal ordering models.

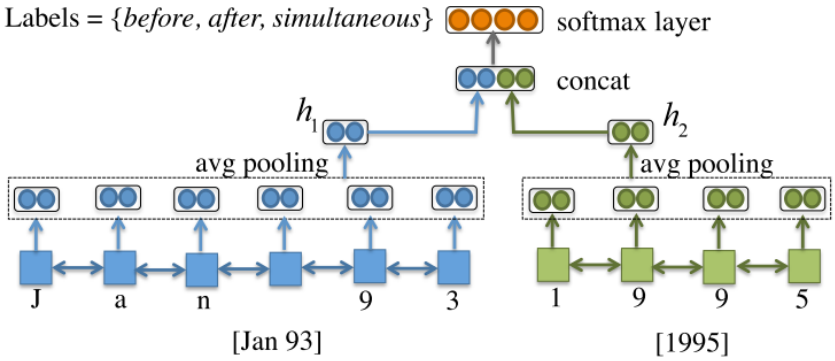

Figure 4.1: Time embedding model from Goyal and Durrett (2019).

Each time phrase is represented as a matrix of ELMo embeddings (using the PubMed model), which is passed through a GRU layer to create a single encoding. This encoding is then concatenated with a flag that represents the type of time phrase (DATE, TIME, DATETIME, DURATION, SET, UNK). The representations of the two time phrases are concatenated together and used as input to the linear classification layer, which predicts the order of the two time phrases.

We add the type representation because a phrase such as "two days" could mean very different things depending on whether it is a date (e.g., 2 days ago) or a duration (e.g., fever for 2 days). In the first case, this is a relative date that may help us determine the order of two events. However, if it is a duration it does not tell us anything about when the event actually occurred. If the model misinterprets this as a date, it could result in an incorrect ordering of events.

We evaluate the TIMEX embedding model on the timex pair classification task (BEFORE, AFTER, SIMULTANEOUS to verify that the model is capturing some sense of temporal value; see Table 4.1.

4.5 Listwise temporal ordering models

The code for the following models, as well as the evaluation metrics, is available at `https://github.com/sjeblee/chrononet`.

We can frame listwise ordering as a ranking task, with a few key differences. In traditional ranking, such as information retrieval, the top of the list is the most important, as the goal is usually to return the best search result. Thus, metrics such as Precision@K and normalized discounted cumulative gain (nDCG) are typically used for optimization and evaluation, and documents are typically ranked one after the other, without ties. However, in event ordering, especially in the medical context, we care equally about the ranking of all events in the timeline, since many events overlap (such as symptoms that co-occur with each other).

Therefore, we must make a few modifications to the model and evaluation methods. We present two models for temporal ordering: linear and grouped. We also present a set of metrics for evaluation.

4.5.1 Linear ordering model

For the first model, we use a gated recurrent unit (GRU) network (Cho et al., 2014), which predicts a rank value between 0 and 1 for each event (rank values scaled by the number of events in the document). Each event in the document is represented as the concatenation of the event encoding generated by the previously described autoencoder model and the time embedding. Each document is then represented by the sequence of event encodings, in the order they appear in the text. The second GRU component operates over the whole document, where each timestep is an input event. At each timestep, the GRU outputs a vector, which is then passed through a feed-forward layer which outputs the rank value. We train the model using ListMLE loss (Xia et al., 2008), which we found to have better performance than L2 loss. We use GRU layers because they have fewer parameters than long short-term memory (LSTM) networks.

We also experiment with using a SetTransformer (ST) layer instead of the first GRU layer, in order to make the model input order-invariant. We use the ST implementation of Vollgraf (2019)[5], using induced set attention blocks. See Figure 4.2 for a diagram of the model architecture.

This model assigns a separate rank value to each event, which results in a linear ordering. However, the human-labeled data has groups of many events at the same rank. In theory, the linear model could output the same rank value for two different events but, in practice, this rarely happens, potentially because one of the major limitations of this model is that the prediction of a rank value for an event is not conditioned on the rank values of other events, even though the events are dependent, since human annotators rank events relative to other events.

In addition, the linear ordering model is biased by the input order of events, because the GRU is conditioned on the order the events appearing in the text, which may be misleading. Even though many events are indeed mentioned in chronological order, there are also many instances where they are mentioned out of order. Ideally, the model would be conditioned on the whole narrative when making a decision about the temporal ordering of a specific event. This is the main motivation for creating a grouped ordering model with order-invariant input.

4.5.2 Grouped ordering: Set-to-sequence (Set2Seq) model

In order to produce a ranking more similar to the human-assigned group ranking, we introduce a grouped ordering model using an extension of the set-to-sequence (Set2Seq) ordering model proposed by Logeswaran et al. (2018).

[5] https://github.com/zalandoresearch/SWARM

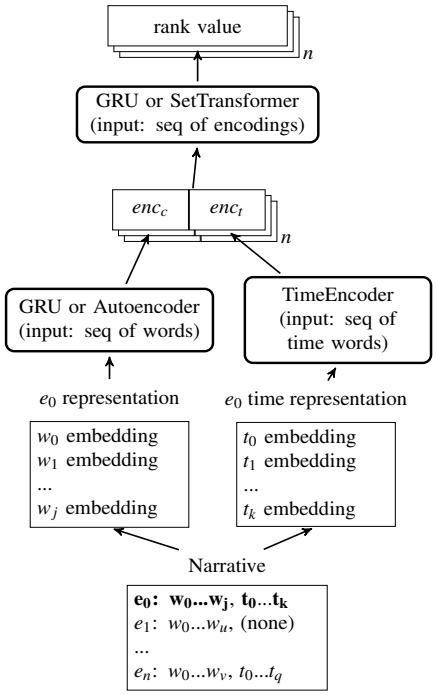

Figure 4.2: Linear ordering model architecture. enc_c is the event context encoding, enc_t is the time phrase encoding, and n is the number of events in the document.

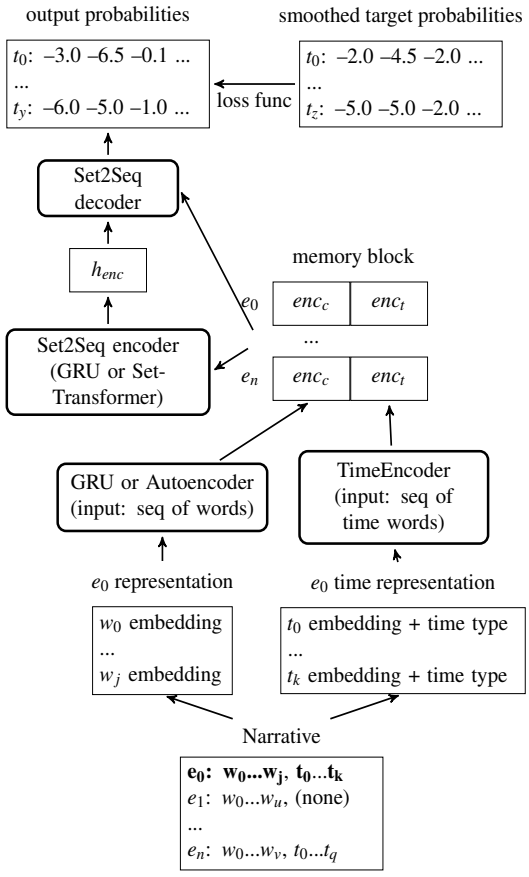

Figure 4.3: Grouped set-to-sequence model architecture. Each encoder is a GRU layer, and the event/context and time encoders are bidirectional.

The structure of a Set2Seq network is similar to that of a Seq2Seq model, which encodes an input sequence using a recurrent neural network (RNN) (the encoder), and then outputs a sequence using a second RNN (the decoder). Because both layers are RNNs, the order of the input sequence influences the output. The Set2Seq model, however, is invariant to input order — the input is treated as a set instead of a sequence. Instead of reading one input at each timestep, the encoder runs for a predefined number of "read cycles" where, at each cycle, it calculates a weighted sum of the entire input sequence based on an attention function. In the input sequence, or memory block, each row corresponds to the encoding of one input item. This architecture is typically called a "memory network" (Weston et al., 2015). In this case the input items are the same event and time encodings we used for the linear model.

The Set2Seq encoder attention and output are calculated as follows (from Logeswaran et al. (2018)):

$$
\begin{aligned}
e^{t,i}_{\text{enc}} &= f\left(s_i, h^t_{\text{enc}}\right), i \in \{1, \ldots, n\} \\
a^t_{\text{enc}} &= \text{Softmax}\left(e^t_{\text{enc}}\right) \\
s^t_{\text{att}} &= \sum_{i=1}^{n} a^{t,i}_{\text{enc}} s_i \\
h^{t+1}_{\text{enc}}, c^{t+1}_{\text{enc}} &= \text{LSTM}\left(h^t_{\text{enc}}, c^t_{\text{enc}}, s^t_{\text{att}}\right)
\end{aligned}
$$

In our model, f is a bilinear function whose parameters are learned from training. Instead of LSTM layers, we use GRU layers because they have fewer parameters. Since we have only a small number of training examples, we want to keep the number of model parameters small as well.

The decoder is initialized with the final state of the encoder and, at each timestep, the input is the encoding of the previous output. The decoder calculates a similar attention function, which is treated as the probabilities for each item in the memory block. The index of the highest probability is selected as the output for that timestep. For the loss function we use Kullback-Leibler divergence (Kullback and Leibler, 1951). Figure 4.3 shows the full architecture of the Set2Seq network. See Vinyals et al. 2016 for more details of the original Set2Seq model.

The benefit of the Set2Seq model is that it does not depend on the input order of the items, since attention is calculated over the whole memory block at each read cycle. The linear ordering model reads the events in the order they are presented in the text, which may bias the model against correctly ordering events that are not mentioned in chronological order. Because the Set2Seq model estimates probabilities for all events at each timestep, it is better suited to predicting simultaneous events.

Modifications to the Set2Seq model

We make several modifications to this model: a mask vector, a grouping probability margin, and target probability smoothing.

Mask: First, we introduce a mask vector that is applied to the attention matrix in the decoder. When the decoder selects an item to output, we set the mask value corresponding to the index of that item to 0 (in practice we use log probabilities, so the mask value is set to $-\infty$). Since we treat the attention matrix as output probabilities, and select the highest probability item at each timestep, this prevents the model from choosing the same item twice. In the typical Set2Seq model, the chosen item's encoding is then fed as input to the decoder for the next timestep. If multiple items were chosen, we use the average of the encodings of all chosen items as the input to the decoder for the next timestep.

Group probability margin: Second, we introduce a probability margin m for choosing the output

item. Instead of choosing only the item with the highest probability, the decoder outputs all items with probability $p > (p_{max} - m)$, where p_{max} is the highest probability in the decoder attention matrix. We set the mask values for all of the chosen items to 0. For our experiments, m was set to 0.001 for the THYME dataset, and 0.05 for the VA dataset, based on performance on the development sets. Unlike in traditional Seq2Seq models, where the decoder must decide when to stop, we know exactly how many items are in the set, so the decoder stops when it runs out of items to choose. The model does not, however, know how many timesteps (i.e., groups of events) there should be; this is determined by the probability margin parameter.

Target probability smoothing: The loss function is the KL divergence between the target probabilities and the predicted probabilities for each timestep. Since the entire ordering is important, we recognize that misplacing an event by one timestep is not as severe an error as misplacing it by many timesteps. Therefore, in order to penalize the model less for getting closer to the correct timestep, we introduce a Gaussian smoothing of the binary target probabilities over the timesteps of the target distribution, with $\sigma = 1$. This gives each event some probability in the timesteps before and after the correct timestep, which results in a greater loss for events that are predicted farther from the correct timestep.

Lastly, we also test a version of the Set2Seq architecture where we replace the original encoder with a SetTransformer layer. We report results for both versions. All models are implemented in PyTorch 1.0 (Paszke et al., 2017).

4.6 Metrics for evaluating listwise temporal ordering

Traditional ranking metrics do not measure the accuracy of the entire ordering; rather, they focus on the accuracy of the top results. But for temporal ordering we care equally about the ranking accuracy of the entire list. We also do not want to treat the rank value for each event independently, because ultimately what matters is the relative order, not the absolute position in the list. We evaluate the models according to several metrics:

- Mean squared error (MSE) of the predicted vs. true rank values. This is a metric that measures how accurate the absolute position of the event is in the overall list.
- Pairwise ordering accuracy (POA), as used by Jeblee and Hirst (2018), with $\varepsilon = 0.01$.[6] Note that POA only tells us whether the events are ordered correctly relative to each other; it does not tell us anything about the distance between those two events.
- Kendall's τ rank correlation coefficient. Lapata (2006) demonstrated that τ has high agreement with human annotations for ordering tasks. We use the τ_b version, which accounts for tied rankings. Since τ is a correlation coefficient, scores near 0 mean no correlation, 1 means perfect correlation, and -1 means perfect negative correlation.
- Events per rank (EPR) is the average number of events assigned the same rank value, for the purpose of making sure the model is creating appropriately sized groups of events.
- Gold pair recall (GPR): For the THYME dataset, we have human-annotated relations between pairs of events, which we have reduced to BEFORE and OVERLAP labels. However, since not all pairs of

[6] Two rank values with a difference $\leq \varepsilon$ are considered to have the same rank.

events are annotated, we cannot calculate precision, only recall. We calculate the GPR separately for BEFORE and OVERLAP relations, GPR_B and GPR_O respectively. While POA examines every pair of events in the list, GPR is calculated only on event pairs that were annotated as having a defined relation.

4.7 Evaluation of temporal ordering models for medical narratives

For our experiments we use the parameters shown in table 4.2. On the VA data, the models take about 6 hours to train and test on a GPU, and for the THYME dataset it takes about 13 hours.

Model	Dataset	Parameter	Value
all	all	event encoding size	64
all	all	time encoding size	32
all	all	hidden size	96
Linear	all	epochs	20
S2S	all	epochs	15
S2S	all	read cycles	25
S2S	THYME	group margin	0.001
S2S	VA	group margin	0.05
ST	all	layers	1
ST	all	attention heads	4
ST	all	inducing points	16

Table 4.2: Parameters used for temporal ordering experiments.

Hyperparameters were chosen using `hyperopt` (Bergstra et al., 2013) on the development datasets over 100 runs. For the VA dataset we used a set of 50 records from the training set as the dev set. For THYME we used the standard development set.

We evaluate these two models and compare them to two baselines: a random ordering of the events (each event is randomly assigned a number between 0 and 1) and "text order" (the order that the events appear in the text of the narrative).

In a real-world scenario, we would need to first identify the events and TIMEXes, given raw text data. To test this scenario, we train a conditional random field (CRF) sequence tagger on the training data, and use it to identify event and time phrases in the test set, which we then use as input to the best temporal ordering model for each dataset. The CRF model has a precision, recall, and F_1 score of .80 on the VA data and .95 on the THYME data. As expected, we see a slight decrease in temporal ordering accuracy when using predicted tags, although this difference is very small.

We also apply the models trained on each dataset (VA and THYME) to the other dataset. We finally train the two models (Linear and Set2Seq) on a combination of the two training sets, and then evaluate on each test set, in order to examine the effects of formal vs. informal medical narratives. The combined model is trained on a total of 606 narratives.

It is important to note that, in converting the pairwise annotations to listwise annotations for the THYME dataset, we were forced to ignore many OVERLAP relations. This is because we are ordering events by start time, so an event could start before another one, but still overlap with it later. Since the linear model predicts one event per rank, it does not capture any OVERLAP relations; thus, the GPR_O

Model	Train data	Entities	MSE	POA	Tau	EPR
Reference list	none	gold	.000	1.000	1.000	1.447
Random order	none	gold	.245	.384	−.018	1.000
Text order	none	gold	.222	.493	−.014	1.000
Linear (GRU)	VA (text order)	gold	**.123**	.708	.705	1.000
Linear (GRU)	VA+TH (text order)	gold	.658	.730	**.777**	1.000
Linear (GRU)	VA (shuffled order)	gold	.788	.609	.502	1.000
Linear (ST)	VA (text order)	gold	.240	**.733**	.330	1.085
Linear (ST)	VA+TH (text order)	gold	.239	.703	.274	1.010
Linear (ST)	VA (shuffled order)	gold	.240	.616	.208	**1.434**
Linear (GRU)	VA (text order)	CRF	.642	.708	.726	1.000
Set2Seq (attn)	VA (text order)	gold	.176	.523	.313	**1.461**
Set2Seq (attn)	VA+TH (text order)	gold	.182	.527	**.371**	2.104
Set2Seq (attn)	VA (shuffled order)	gold	.173	**.546**	.367	1.623
Set2Seq (ST)	VA (text order)	gold	.180	.528	.312	1.867
Set2Seq (ST)	VA+TH (text order)	gold	.178	.489	.352	2.430
Set2Seq (ST)	VA (shuffled order)	gold	**.169**	.516	.320	1.735
Set2Seq (attn)	VA (shuffled order)	CRF	.163	.517	.397	2.430

Table 4.3: Listwise ordering on the VA test set (100 records). ST: SetTransformer version.

Model	Train data	Entities	MSE	POA	Tau	EPR
Reference list	none	gold	.000	1.000	1.000	1.447
Linear (GRU)	VA (text order)	gold	.913	**.699**	**.754**	1.000
Set2Seq (attn)	VA (text order)	gold	**.155**	.551	.396	**1.265**

Table 4.4: Listwise ordering model evaluation on the VA dataset with 10-fold cross-validation.

score is 0.

Since the annotated portion of the dataset is much smaller than the full dataset, we run the best performing sequence tagger from Chapter 3 (CRF), and use the predicted events and time phrases to train and evaluate the temporal ordering model, as this is what we would need to do in a real-world scenario with new data.

4.7.1 Results of temporal ordering models

Table 4.3 shows the results of the two ordering models on the VA test set. We also run the best linear and grouped models with 10-fold cross-validation; results shown in Table 4.4.

Table 4.5 shows the results of the two ordering models on the THYME test set, as well as the previous ListNet model. Table 4.6 shows the results of the best linear and grouped models with 10-fold cross-validation.

For the Verilogue dataset, we include only events that have annotated rank values, which is a small percentage of all events. Due to the small size of the dataset, we also apply the model trained on VA data to the Verilogue test set. The results are shown in Table 4.7.

We also conduct some experiments to determine the best parameter settings for the verbal autopsy

Model	Train data	Entities	MSE	POA	Tau	EPR	GPR_B	GPR_O
Reference list	none	gold	.000	1.000	1.000	5.231	.844	.246
Random order	none	gold	.171	.363	.002	1.000	.482	0
Text order	none	gold	.172	.494	−.012	1.000	.481	0
ListNet	TH	gold	.072	.517	–	–	.420	.254
Linear (GRU)	TH (text order)	gold	.256	.550	**.474**	1.000	.543	0
Linear (GRU)	VA+TH (text order)	gold	.356	.551	.458	1.000	.510	0
Linear (GRU)	TH (shuffled order)	gold	**.076**	.550	.471	1.000	**.554**	0
Linear (ST)	TH (text order)	gold	.368	.741	.259	1.091	.476	.002
Linear (ST)	VA+TH (text order)	gold	.368	**.744**	.267	**1.136**	.496	**.007**
Linear (ST)	TH (shuffled order)	gold	.365	.743	.358	1.001	.510	0
Linear (ST)	TH (text order)	CRF	.368	.741	.259	1.097	.475	0
Set2Seq (attn)	TH (text order)	gold	.218	.291	−.192	2.280	.431	.071
Set2Seq (attn)	VA+TH (text order)	gold	**.102**	**.449**	.237	11.039	.335	.342
Set2Seq (attn)	TH (shuffled order)	gold	.396	.285	−.214	**2.459**	**.432**	.084
Set2Seq (ST)	TH (text order)	gold	.179	.435	**.293**	23.585	.239	.605
Set2Seq (ST)	VA+TH (text order)	gold	.126	.354	.010	19.698	.317	.438
Set2Seq (ST)	TH (shuffled order)	gold	.197	.387	.240	30.220	.179	**.761**
Set2Seq (ST)	TH (text order)	CRF	.197	.387	.240	30.221	.177	.761

Table 4.5: Listwise ordering on the THYME test set (106 records). TH: trained on THYME; VA: trained on verbal autopsy; MSE: mean squared error; POA: list pairwise ordering accuracy; Tau: Kendall's τ with tied ranks; EPR: average events per rank; GPR_B: gold-standard pairwise relation recall of BEFORE relations; GPR_O: GPR of OVERLAP. ListNet results from (Jeblee and Hirst, 2018).

Model	Train data	Entities	MSE	POA	Tau	EPR	GPR_B	GPR_O
Reference list	none	gold	.000	1.000	1.000	5.231	.844	.246
Linear (GRU)	TH (text order)	gold	.195	**.541**	**.436**	1.000	**.500**	.000
Set2Seq (attn)	TH (text order)	gold	**.194**	.376	−.005	**5.367**	.321	**.301**

Table 4.6: Listwise ordering model evaluation on the THYME dataset with 10-fold cross-validation.

Model	Train data	Entities	MSE	POA	Tau	EPR
Reference list	none	gold	.000	1.000	1.000	2.478
Random order	none	gold	.167	.342	.151	1.000
Text order	none	gold	.228	.453	.393	1.000
Linear (GRU)	Verilogue (text order)	gold	**.186**	.460	.348	1.000
Linear (GRU)	VA (text order)	gold	.281	**.468**	**.423**	1.000
Set2Seq (attn)	Verilogue (text order)	gold	.342	.397	−.009	**2.280**
Set2Seq (attn)	VA (text order)	gold	.296	.309	−.058	1.839

Table 4.7: Listwise ordering on the annotated Verilogue test set (15 records).

Model	Feature	Value	MSE	POA	Tau	EPR
Reference list	none	–	.000	1.000	1.000	1.447
Random order	none	–	.245	.384	−.018	1.000
Text order	none	–	.222	.493	−.014	1.000
Linear (GRU)	context size	**5**	**.376**	.691	**.696**	1.000
Linear (GRU)	context size	10	.602	**.692**	.689	1.000
Linear (GRU)	context size	15	.690	.684	.662	1.000
Set2Seq (attn)	context size	**5**	.196	.482	.239	**1.436**
Set2Seq (attn)	context size	10	.189	.510	.298	1.623
Set2Seq (attn)	context size	15	**.175**	**.585**	**.412**	1.793
Linear (GRU)	event encoding size	32	**.295**	.684	.670	1.000
Linear (GRU)	event encoding size	**64**	.376	.691	.696	1.000
Linear (GRU)	event encoding size	128	1.096	**.695**	**.697**	1.000
Set2Seq (attn)	event encoding size	32	**.156**	**.527**	**.374**	1.833
Set2Seq (attn)	event encoding size	**64**	.196	.482	.239	**1.436**
Set2Seq (attn)	event encoding size	128	.252	.411	.113	2.250
Linear (GRU)	time encoding size	16	.591	.705	.717	1.000
Linear (GRU)	time encoding size	**32**	**.376**	.691	.696	1.000
Linear (GRU)	time encoding size	64	1.384	**.736**	**.792**	1.000
Set2Seq (attn)	time encoding size	16	.191	.505	.307	1.600
Set2Seq (attn)	time encoding size	**32**	.196	.482	.239	**1.436**
Set2Seq (attn)	time encoding size	64	**.172**	**.561**	**.387**	1.584
Linear (GRU)	no time type		.412	.688	.677	1.000
Linear (GRU)	**with time type**		**.376**	**.691**	**.696**	1.000
Set2Seq (attn)	no time type		**.157**	**.567**	**.415**	1.464
Set2Seq (attn)	**with time type**		.196	.482	.239	**1.436**
Linear (GRU)	**no flags**		.376	.691	.696	1.000
Linear (GRU)	position flag		**.341**	**.696**	**.705**	1.000
Linear (GRU)	polarity flag		.592	.669	.648	1.000
Set2Seq (attn)	**no flags**		.196	.482	.239	**1.436**
Set2Seq (attn)	position flag		**.145**	**.578**	**.449**	1.286
Linear (GRU)	polarity flag		.191	.510	.307	1.617

Table 4.8: Feature ablation study on the VA dataset. Unless otherwise specified the parameters are as follows (default feature values in bold): text input order, 15 epochs, context size = 5, time encoding size = 32 (including time type), event encoding size = 64, event autoencoder epochs = 30, dropout = 0.1. Best results for each model and parameter combination in bold.

Narratives from MDS
Since *a year*, the deceased used to **bleed from the mouth every time he coughed**. After **taking medicines** prescribed by the Doctor, she **stopped bleeding every time she coughed**. She also had **asthma** since *eight years*. *Two days before her death*, she had had an **Asthma attack** and she was **rushed to XXX Hospital**. When her **condition did not improve** even after **keeping in the Hospital** for *three days*, she was **brought back home**. She **passed away** *a day after being brought home*.
2 days before she had **severe pain in abdomen**, **burning sensation in stomach**, no **fever** nor **vomiting**, no any **discharged summary** nor any other scripts found.

Table 4.9: Two verbal autopsy narratives with human-annotated events (boldface) and time phrases (italics). Their CoD categories are "Chronic respiratory disease" and "Ischemic heart disease".

Event	**TIMEX**	**Correct**	**Linear**	**ST**	**S2S (attn)**	**S2S (ST)**
asthma	eight years	1	4	3	5	4
bleed from the mouth every time he coughed	a year	2	1	4	2	4
taking medicines	(none)	3	2	1	1	3
stopped bleeding every time she coughed	(none)	4	3	6	1	2
Asthma attack	Two days before her death	5	5	2	4	2
rushed to XXX Hospital	(none)	6	6	10	1	2
keeping in the Hospital	three days	7	8	5	7	5
condition did not improve	(none)	8	7	7	8	1
brought back home	(none)	9	9	8	6	5
passed away	a day after being brought home	10	10	9	3	6
severe pain in abdomen	2 days before	1	1	3	1	1
burning sensation in stomach	2 days before	1	2	2	2	1
fever (neg)	(none)	1	3	1	2	2
vomiting (neg)	(none)	1	4	5	2	2
discharged summary (neg)	(none)	1	5	4	3	2

Table 4.10: Example records from Table 4.9 with correct and predicted ranks from the text order input models. S2S: Set2Seq, ST: SetTransformer

data. Results are shown in Table 4.8. Overall, adding a parameter for the position of the event in the narrative improves the results for both models. While the Set2Seq model benefits from a longer embedding context and a larger time encoding, the linear model performs better with a larger event encoding and smaller time encoding.

4.7.2 Discussion and analysis

For all single-dataset experiments except the Linear (TH) model, using the pre-trained timex embeddings instead of a simple GRU layer improves the temporal ordering performance. This shows us that the embedding model is indeed capturing useful time value information from these phrases.

Although the reference annotations have events grouped together (5.23 EPR for THYME, and 1.45 for VA), the linear ordering models performs better on POA and Kendall's τ. However, the linear

ordering model is only predicting a numerical value for each event, whereas the group model must predict a probability distribution across all events at each timestep, without knowing in advance how many timesteps there should be (i.e., how many groups of events). This makes the grouped ranking a much harder problem.

While the linear ordering models perform better in terms of ordering metrics, the Set2Seq models are robust to changes in the input order due to calculating attention over the whole input set. While the linear models experience a statistically significant drop in Tau and POA (p-value < 0.01) when the input order is randomized, the difference in performance for the Set2Seq models is not statistically significant (p-value$= 0.22$)[7].

We note that on the THYME dataset, the Set2Seq (ST) model tends to group too many events together (19 to 30, whereas the reference is 5.321 events per rank). However, this could be remedied with further tuning of the grouping threshold.

Combining both training sets produces better performance in terms of Kendall's τ for the VA test set for the Linear (GRU) model and both Set2Seq models. For the THYME dataset, training on the combined data achieves a better τ for the Linear (ST) and Set2Seq (attn) models.

Table 4.10 shows the ranking predictions from the linear and Set2Seq models on the two example narratives, compared to the human-annotated ranks. In the first example output of the linear model, only the event "asthma" is out of place, but the rest of the ordering is correct. The linear model simply predicts the events in the order they appear in the text, which is the best we can expect from the linear model, but the Set2Seq models are able to group some of the events together.

The performance on the Verilogue data improves when we train the model on the VA dataset, likely because the VA dataset is much larger than the Verilogue training set. We pre-train on VA narratives rather than THYME because both the VA and Verilogue datasets contain informal medical language, whereas the THYME narratives are more formal.

While the overall ordering accuracy is better using the linear model, and this may be a good approximation for the VA dataset (which has fewer events per rank), the THYME dataset has an average of more than 5 events per rank, so the Set2Seq model is more appropriate for this type of grouped temporal ordering task.

However, there are some limitations that currently affect the performance of the model. At this point, the biggest limitation is the amount of annotated training data available. Because the model has many parameters and the input embeddings are quite large (1024 dimensions), we believe that more training data will improve the model. However, we can use predicted events and TIMEXes when human-annotated ones are not available, and still achieve similar temporal ordering performance.

4.8 Conclusion

We presented two models for listwise ordering of events in medical text using time embeddings: linear and Set2Seq. Both models outperform the baseline of assuming that events are mentioned in chronological order. This appears to be the first use of a grouped Set2Seq model, and the first application of a Set2Seq model to the task of temporal ordering.

In the next chapter, we use these predicted timeline orderings as input to cause-of-death classification models (Jeblee et al., 2019a) to see whether they will improve classification accuracy over typical word

[7]Two-sided t-test calculated on the VA results using `scipy.stats.ttest_rel`

embedding models. This type of ordering model can easily be applied to temporal ordering in other domains, as well as other listwise ordering tasks.

Chapter 5

Disease classification models

5.1 Introduction

The end goal of our work on temporal information extraction and classification is to correctly diagnose a patient with a disease or cause of death using the text of a medical narrative. Once we have extracted relevant medical events and time phrases, and determined chronology, the last step is to use that information as input to an automated classifier.

Automated classification for health records is an important task because it can save valuable time for both healthcare professionals and patients. While some classification tasks involve learning a mapping from natural language to a specific category, as is the case with medical coding, other tasks require inference and domain expertise.

For example, some ICD-10 coding tasks involve assigning the correct code from a text description of the disease or CoD. Although this can be a difficult task due to the variability of language and large number of possible labels, CoD classification from VA narratives is even more difficult in the sense that the CoD is usually not present in the text. Rather, the model must infer the diagnosis from descriptions of symptoms. Both automated classification tasks can be very useful to medical professionals, but CoD diagnosis is an inherently harder problem, even for human coders.

The goal of our automated diagnosis models is not to replace human physicians, but rather to decrease their workload by providing a first-pass diagnosis, and/or coding the "easier" examples (CoDs with high physician agreement and model confidence) and deferring the harder ones to human physicians. Automated diagnoses can be provided to physicians, who are responsible for making final judgements and managing patient care.

In this chapter, we first examine machine-readable representations of the extracted timelines, then we present supervised classification models, and lastly we present the results of those models on different types of medical narratives (verbal autopsies and clinical conversations) [1], including some transfer learning experiments.

[1] We do not include experiments on the THYME dataset because it only contains two different types of cancer diagnoses, this does not make for a very interesting classification problem. However, these methods could be used for a larger dataset of clinical notes with a greater variety of diagnoses, such as the MIMIC dataset. However this would also require some annotation which is not currently available.

5.1.1 Related work on medical text classification

Text classification is an old problem for NLP, and since there has been extensive work on text classification models, we will focus here specifically on disease and cause-of-death classification models.

In order to apply deep learning for electronic health records (EHRs), Miotto et al. (2016) developed Deep Patient, an EHR representation model using a stack of denoising autoencoders, which was trained on a dataset of EHRs from about 700,000 patients. They then used the generated representations to perform topic modeling and disease classification and found that the Deep Patient representation provided better classification results than traditional feature extraction methods.

Although large transformer-based models such as BERT have been successfully applied to many NLP tasks, the publicly available models are typically trained on general domain data such as Wikipedia, and therefore do not always perform well on domain-specific data. However, such models can either be trained from scratch or fine-tuned on in-domain data.

Alsentzer et al. (2019) fine-tuned BERT and BioBERT models on the MIMIC-III dataset, demonstrating that, as expected, the fine-tuned model performed better on clinical NLP tasks.

XLNet (Yang et al., 2019) is another transformer-based language model that has been shown to outperform BERT on some NLP tasks. Huang et al. (2019) adapted XLNet to the clinical domain by training on clinical notes from MIMIC-III and fine-tuning the model on temporally sequenced notes using a bi-LSTM layer. ClinicalXLNet outperformed BERT, ClinicalBERT, and XLNet on clinical prediction tasks using the MIMIC-III dataset.

Mascio et al. (2020) tested numerous combinations of word representation methods, pre-processing, and text classification models for electronic health records. The bi-LSTM classification model with embeddings custom trained for the target task performed the best, although pre-trained BERT and BioBERT methods were close in terms of performance. They found that tokenization and pre-processing methods had a relatively small impact on performance, suggesting that word representations should be trained on in-domain data when possible, but if there is not enough data available for training a custom model, a large pre-trained model such as BERT is a good alternative.

While these results are promising for clinical notes, due to the differences in text style, these models might not perform well on VA data, and there is currently not enough publicly available VA data to train these models.

5.1.2 Related work on Verbal autopsy classification and automated coding

Some material in this subsection is based on previously published work (Jeblee et al., 2019a).

Several expert-driven and machine learning methods have been used for automatically categorizing VAs by CoD, at both the individual and the population level (Boulle et al., 2001; Byass et al., 2012; McCormick et al., 2016; James et al., 2011; Miasnikof et al., 2015; Danso et al., 2013a,b). Expert-driven methods use knowledge curated by domain experts (in this case, medical professionals) to generate resources such as medical dictionaries that can be used for automated classification. Many of these methods use response data from questionnaires such as the World Health Organization (WHO) 2016 Verbal Autopsy Instrument (Nichols et al., 2018), which is a standardized VA questionnaire with detailed questions about the subject's symptoms and medical history.

VA classification methods are often evaluated at both the individual and population level. For

individual record performance, many methods report recall (sensitivity), F_1 score (the harmonic mean of precision and recall), or chance-corrected correspondence (CCC). For population-level performance the predominant metric is cause-specific mortality fraction (CSMF) accuracy. Since CSMF accuracy can be high even with random guessing, some have advocated for chance-corrected CSMF accuracy (Flaxman et al., 2015), which re-scales CSMF accuracy so that random assignment results in a score of 0 (whereas random assignment can result in a regular CSMF accuracy score of around .6).

Boulle et al. (2001) were among the first to use neural networks for VA CoD classification in 2001. They used a small set of structured questionnaire data with a neural network and achieved a sensitivity of .453 for individual classification into 16 CoD categories. However, to our knowledge, none of the automated VA coding methods currently in use have applied neural networks despite their recent popularity.

The King-Lu method (King and Lu, 2008) uses the conditional probability distributions of symptoms to estimate the CoD distribution of a dataset over 13 categories. It does not provide a CoD for individual records. Desai et al. (2014) reported a CSMF accuracy of .96 using the King-Lu method on the Indian Million Death Study dataset (Aleksandrowicz et al., 2014).

InterVA-4, a popular automated VA coding method developed by Byass et al. (2012), uses a predetermined list of symptoms and risk factors extracted from a structured questionnaire. Records are assigned one of 62 CoD categories from the WHO 2012 VA Instrument (World Health Organization, 2012). The category is assigned according to conditional probabilities for each symptom given a CoD, as assigned by medical experts, as well as the probabilities of the CoDs themselves. Miasnikof et al. (2015) reported a sensitivity of .43 and CSMF accuracy of .71 for InterVA-4 on data from the Million Death Study (Aleksandrowicz et al., 2014). InterVA-5 (Byass et al., 2019) was updated to use the 2016 WHO VA instrument, and is open-source. It was evaluated on the PHMRC dataset, with a concordance correlation of .86 for adult deaths.

InSilicoVA, described by McCormick et al. (2016), is a statistical tool that uses a hierarchical Bayesian framework to estimate the CoD for individual records as well as the population distribution. They reported a mean sensitivity of .341 across 34 CoD categories for individual records, and .85 CSMF accuracy.

The Tariff Method, presented by James et al. (2011) and Serina et al. (2015), uses a sum of weighted scores (tariffs) to determine the most probable CoD. The score for each of the possible CoDs is the weighted sum of different tariffs, which are each calculated from the value of a certain indicator (usually a symptom or risk factor). Most of these indicators are taken from the structured questionnaire, although there are also tariffs that represent the presence of some frequent narrative words (50 or more occurrences in the training data). James et al. (2011) reported .505 CCC and .770 CSMF accuracy for adult records from the PHMRC dataset, using 53 CoD categories.

Miasnikof et al. (2015) used a naïve Bayes classifier to assign CoD categories. They evaluated their classifier on several different datasets, including the PHMRC dataset and the Million Death Study dataset (Aleksandrowicz et al., 2014; Gomes et al., 2017), which we will use in this paper (see section 2.11), with 16 CoD categories. They obtained results that surpassed those of the Tariff Method and InterVA-4, including a sensitivity of .57 and CSMF accuracy of .88. However, their model used only data from the structured questionnaire.

Murtaza et al. (2018) used a one vs. all naïve Bayes classifier to assign CoD to VA records from MDS, PHMRC, as well as datasets from South Africa and Bangladesh. They found that the results

more closely resembled physician coding than other leading VA methods including Tariff, InterVA-4, and InSilicoVA.

Danso et al. (2013a) used word frequency counts and tf·idf scores (the frequency of a term divided by the frequency of documents in which it occurs) from VA narratives as features with a support vector machine (SVM) classifier, achieving a maximum F_1 score of .419. They also used a naïve Bayes classifier and a random forest classifier, which achieved F_1 scores of .373 and .149 respectively. They did not report population level metrics.

Danso et al. (2013b) used a variety of linguistic features such as part-of-speech tags, noun phrases, and word pairs from 6,407 VA narratives of infant deaths from Ghana, and classified the records into 16 CoD categories, achieving a sensitivity of .406 using only the narrative-based features and .616 using a combination of narrative and structured questionnaire features. They noted that they achieved better performance with the linguistic features than with only word occurrence features, though their dataset was small and the part-of-speech tagger was not trained on medical data, and thus is likely to produce incorrect part-of-speech information.

In a study of neonatal deaths using the VA 2012 VA survey form, Aggarwal et al. (2013) found that VA diagnostic accuracy was lower for congenital anomalies and perinatal asphyxia than for other neonatal causes. The study was done on 313 neonatal deaths in northern India.

Mujtaba et al. (2017) created a VA classification model using features ranked by domain experts as input to several supervised classifiers, including naive Bayes, SVM, k-nearest neighbor, decision tree, and random forest. The dataset consisted of 2200 VAs of accident-related deaths in Malaysia. They found that expert-selected features performed around .1 F_1 better than automatically selected features. However, this approach requires a domain expert to select features, which is time consuming and not necessarily adaptable to new datasets. It is also possible that newer context-based language model features such as BERT or ELMo embeddings could produce similar performance without the manual effort.

Blanco et al. (2020) investigated combining structured and narrative data together with a neural model for automated coding of VAs. The structured data was represented by low-dimensional categorical embeddings, which were concatenated with word embeddings learned by the model. The classification model was a bi-directional GRU with attention and max and average-pooling layers. They evaluated the models on the PHMRC dataset and found that the models performed the better with the combined data than with the narrative or structured data alone.

Serina et al. (2016) investigated the consistency and reliability of VA questionnaires by conducting a second survey of deaths from the PHMRC dataset. They found that the reliability of question answers was low (.447 mean kappa); however, they evaluated only the questionnaire items, not the text narrative. They concluded that respondents may report different aspects of the same illness in different interviews, but the information provided is still sufficient to make a diagnosis using an automated coding method such as the Tariff Method.

The most widely used automated VA coding methods (InterVA, the Tariff Method, and InSilicoVA) are still mainly dependent on the questionnaire data, as well as the specific data input format. At best they only capture individual words from a pre-defined set if they appear in the narrative. Our goal is to harness the temporal and contextual information from the narrative that these methods have missed. To our knowledge no other VA classification methods have made use of explicit time information from the narrative.

5.2 Representing timelines for downstream models

We use two types of features for the classification models: the matrix of word embeddings as generated by ELMo, and a matrix representing the timeline of events in the predicted temporal order. In the timeline matrix, each row represents an event. The embedding vector is the concatenation of the event encoding and the associated timex encoding, as described in Section 4.4 (see also the "representation" blocks in Figures 4.2 and 4.3).

For experiments where we use the human annotated events and/or correct ordering, we use the same autoencoder and time encoder models to generate the representation of the event timeline. Thus, while the "gold" events and ordering are correct in terms of text span and temporal ordering, as features they are still dependent upon the representation generated by the autoencoder and time encoder models.

5.3 Classification models

Classification models map each input item to an output class, usually via the extraction of some important features. In our case, the input is the text of the medical narrative and the output is the CoD or disease category. The features used will either be the word embeddings of the entire narrative, or the timeline representation described in the previous section.

5.3.1 Baseline classification models

Some material in this subsection is based on previously published work (Jeblee et al., 2019a).

For our first CoD classification experiments, we used several basic machine learning models, including naïve Bayes, random forest, and SVM. The features that we use are word frequency counts from the narrative and one feature that indicates whether the record is of an adult, child, or neonatal death. We compute the ANOVA F-value[2] for each feature, which calculates the ratio of the variance between the means of the feature values for each of the CoD categories to the variance within each class. If the means are significantly different between CoD categories and the variance within categories is small, then the feature is likely to be discriminative. We keep only the features with the highest F-values, reducing the space from over 4000 to several hundred features, depending on the model (the actual number is chosen by hyper-optimization).

All baseline models except the neural network are created in Python with scikit-learn. Each classifier is optimized[3] for 100 runs for model parameters and the number of features, using a small subset of the MDS data. The models are optimized separately so we are comparing the best version of each model. The naïve Bayes classifier, which assigns a CoD category to a record using the independent conditional probabilities for each feature, uses the best 200 features (as chosen by ANOVA). The random forest model, which uses a combination of learned decision trees to classify new data points, uses the best 414 features and 26 trees.

Support vector machines (SVMs) are commonly used models that learn to classify data by maximizing the margin between categories in the training data, using a kernel function that maps the input features

[2] We use scikit-learn's SelectKBest module with the f_classif function.
[3] For optimization we use the hyperopt Python library (Bergstra et al., 2013).

to higher dimensional space. Our SVM model is an aggregate of one-vs-rest SVMs with linear kernel functions, using 378 features.

The baseline neural network model we use is a feed-forward network with one hidden layer (297 nodes, chosen by optimization) created with Keras (Chollet, 2015), using Theano (Theano Development Team, 2016) as the backend. It uses 398 features and rectified linear units (ReLUs) as the activation function (the function that computes the output of an artificial neuron in the network given input values and learned weights).

For the adult and child datasets, each training set is augmented with all the data from the other two datasets. In general, we found that the classifiers perform better with extra training data, especially for the smaller child dataset. For neonatal records, the models are trained only with neonatal data because these records use a different set of CoD categories.

We also use several neural network classifier models. The convolutional neural network (CNN) model uses convolutional filters over the embedding matrix, followed by max-pooling, to extract feature vectors. Although this architecture is commonly used for image classification, it has been shown to perform well for text classification tasks (Kim, 2014). To adapt it to this task, we use filters that are the full width of the embeddings, and kernel sizes of 1 to 5, which capture features from sequences of 1 to 5 words.

The gated recurrent unit (GRU) network (Cho et al., 2014) is a type of RNN that can capture longer-range dependencies, similar to an LSTM model but with fewer parameters. For many text classification tasks GRU models have been shown to have similar performance to LSTMs. The GRU classification model has a similar architecture to the CNN, except that the convolutional and pooling layers are replaced by a GRU layer with a hidden size of 128.

5.3.2 End-to-end models

In a real-world scenario, we will typically receive plain text with no annotations. Therefore, we must conduct all three steps (event/time extraction, temporal ordering, and classification) in an automated manner.

In order to incorporate the temporal information extracted from the temporal ordering models, we also use an end-to-end model that includes the temporal ordering task as well as the classification task, and trains the model for both tasks together. The temporal ordering component produces a timeline representation as described in Section 5.2, which is then used as input to the classification module, which can be the GRU or CNN model described in Section 5.3.1. In addition to the timeline features, we can add a parallel CNN module over the word embeddings of the narrative. The resulting features are concatenated with the features from the timeline before being passed to the classification layer. The model is trained with backpropagation over the entire network. We also present results using human-labeled events and time phrases, as well as event/time phrases predicted by the CRF sequence tagger.

We will compare the results using the end-to-end system (automated event/time extraction, temporal ordering, and disease classification) to results using human-annotated events/times, and also human-annotated temporal ordering (i.e. the upper bound of how well the classifier can do with correct temporal ordering). For the sequence tagger and temporal ordering models, we use the model that achieved the best results for each dataset, as demonstrated in Chapters 3 and 4 respectively. For most datasets this is the CRF sequence tagger and the linear (GRU) ordering model. We also present some preliminary results with the Set2Seq model for comparison.

5.4 Evaluation of disease classification models on different types of medical text

5.4.1 Results on verbal autopsies

Table 5.1 shows the baseline classification results on the VA dataset using only the text of the narrative (no event or time information).

Table 5.2 shows the results of CoD classification on only the annotated VA records, using the different sequence taggers and temporal ordering models. Here, we can compare the end-to-end results using predicted event/time phrases and predicted temporal ordering to the human-annotated event/time phrases and temporal ordering. The random classification model assigns CoD categories to the test instances randomly, with probabilities according to the distribution of the training data.

Adult (18 categories)	**Precision**	**Recall**	**F_1**	**CSMFA**	**CCCSMFA**
SVM (word counts)	.721	.717	.718	**.963**	**.896**
Neural network (word counts)	.749	**.749**	**.747**	.961	.891
GRU (word2vec)	.736	.740	.732	.938	.826
CNN (word2vec)	**.755**	**.749**	.745	.929	.801

Table 5.1: CoD classification results without symptom information: mean scores from 10-fold cross-validation on the MDS+RCT adult dataset (14,313 records). CSMFA: cause-specific mortality fraction (CSMF) accuracy, CCCSMFA: chance-corrected CSMFA.

Event /time tagger	**Ordering model**	**Classification model**	**Classification features**	**Precision**	**Recall**	**F_1**	**CSMFA**	**CCCSMFA**
None	None	Random	None	.023	.043	.030	.743	.280
None	None	CNN	ELMo word emb (1024 dim)	.574	**.623**	.582	.794	.423
None	None	GRU	ELMo word emb (1024 dim)	.453	.450	.433	.786	.401
Gold*	Linear	CNN	timeline (192)	.124	.260	.163	.469	−.487
Gold*	Linear	CNN	timeline + words	.579	.610	.580	.837	.543
Gold*	Set2Seq	CNN	timeline (96)	.265	.300	.234	.551	−.258
Gold*	Linear	GRU (e2e)	timeline (192)	.447	.390	.393	.786	.401
Gold*	Linear	CNN (e2e)	timeline (192) + words	.587	.610	.583	.786	.401
Gold*	Set2Seq	CNN (e2e)	timeline (96) + words	.631	.610	.601	.857	.599
CRF	Linear	CNN	timeline + words	.570	.580	.553	.786	.401
CRF	Linear	CNN (e2e)	timeline + words	**.617**	.590	.564	.755	.314
CRF	Set2Seq	CNN (e2e)	timeline (96) + words	.612	.610	**.592**	**.826**	**.513**

Table 5.2: CoD classification results on the annotated VA dataset (700 records of adult deaths, 600 train, 100 test) using words and ordered events. Gold* indicates that we used the human-annotated event/time phrases or temporal ordering as input. In the "Classification model" column, "e2e" indicates that the temporal ordering and classification model was trained end-to-end (the event/time tagger was still run separately as a first step).

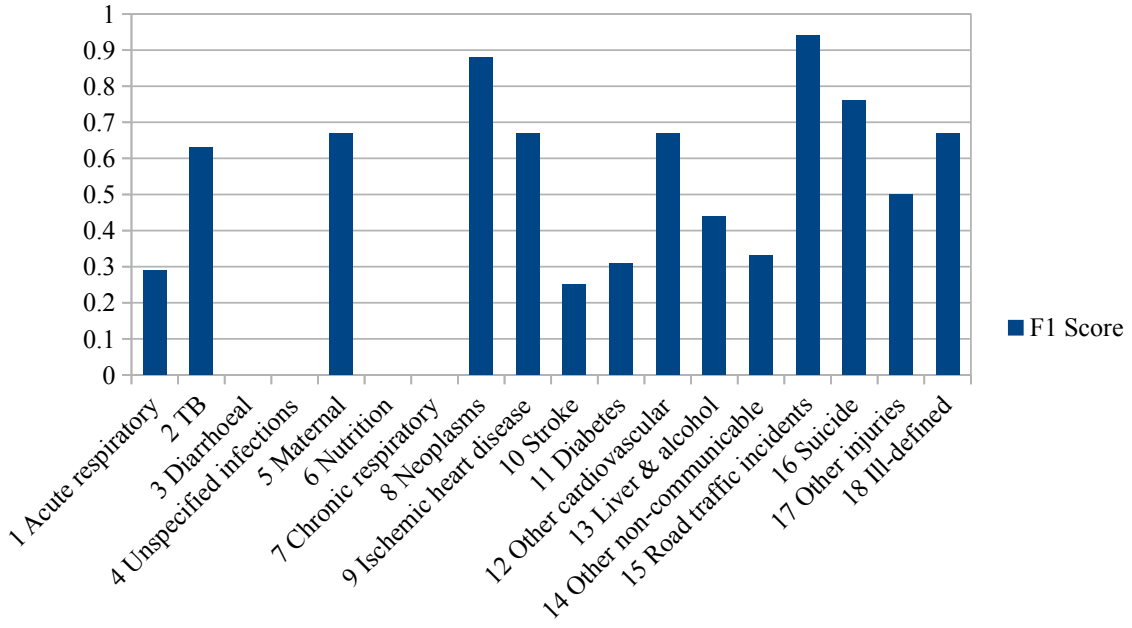

Figure 5.1: Classification F_1 scores of the best model (CRF tagger, Set2Seq temporal ordering, end-to-end trained CNN) by CoD category.

Model	Precision	Recall	F_1	CSMFA F_1	improvement
Adult (18 categories)					
Word2Vec (CNN)	.759	.755	.751	.933	.000
Character-word Emb. Concatenation (CNN)	.716	.699	.699	.912	−.052
ELMo (CNN)	.724	.735	.725	.934	−.026
ELMo (CNN) with features	**.779**	**.769**	**.768**	.963	**.017**
BERT (GRU)	.749	.750	.747	.963	−.004
BERT (GRU) with features	.752	.751	.750	**.964**	−.001
Child (11 categories)					
Word2Vec (CNN)	.713	.707	.697	.902	.000
Character-word Emb. Concatenation (CNN)	**.740**	.718	.712	.890	.015
ELMo (CNN)	.728	.720	.711	.905	.014
ELMo (CNN) with features	.732	.734	.723	.924	.026
BERT (GRU)	.731	.736	.726	.934	.029
BERT (GRU) with features	.734	**.738**	**.731**	**.940**	**.034**
Neonate (5 categories)					
Word2Vec (CNN)	.515	.556	.515	.795	.000
Character-word Emb. Concatenation (CNN)	.562	.585	.556	.819	.041
ELMo (CNN)	.473	.480	.451	.751	−.064
ELMo (CNN) with features	**.618**	**.603**	**.589**	.831	**.033**
BERT (GRU)	.517	.541	.520	.860	−.036
BERT (GRU) with features	.563	.562	.551	**.888**	−.005

Table 5.3: Results from 10-fold cross-validation for each model and each age group in the MDS dataset. F_1 improvement is relative to the first baseline model (Word2Vec (CNN)).

However, the annotated dataset is quite small for training such complex neural network models, but we have limited annotated data. In order to use more data, we trained the event/time tagger on the subset of records for which annotations were available for each fold, and then applied the tagger to both the training and test set. We then use the predicted event/time tags for the rest of the pipeline. Although this may introduce errors from the sequence tagger, this more accurately represents a real-world scenario where we will be given raw text data.

Table 5.3 shows the results of CoD classification with the text of the narrative and some non-narrative features (language and region), using ELMo and BERT models (Yan et al., 2019). For this preliminary work, we tested our baseline model with word2vec embeddings against a CNN model using a combination of word and character embeddings[4] We also tested two popular language models: ELMo and BERT. Lastly we performed an ANOVA to determine which non-textual features from the MDS data were the most informative for classification, and found the top two to be "language" (the language of the original narrative) and "region" (the state of India where the death occurred). These two features were added to the ELMo and BERT models, and generally improved classification performance.

5.4.2 Results on non-English verbal autopsies

Some of the material in this subsection is based on previous work (Kumar et al., 2021).

Since the MDS contains many narratives that were originally recorded in languages other than English, we apply similar classification models to a set of 500 VA narratives in Hindi.

Hindi pre-processing and feature selection

Initially, the Hindi narratives were transcribed from scans of handwritten forms to digital documents in Latin characters. We then transliterated the narratives into Devanagari using an automated Python script. Some of the words were subsequently corrected manually due to incorrect output from the script. Stop-words, which include common articles, conjunctions, and pronouns, such as "a" and "the", were also removed from the narratives.

We tested a variety of models with two different types of features: word frequency counts and word embeddings. The word frequency features were generated using the same process used for the English data. For the CNN, LSTM, and GRU models, we used the following freely available word embeddings for Hindi, which cover approximately 90% of the words in the original Hindi narratives:

- *FastText* (Bojanowski et al., 2017), which acts as an extension of conventional word2vec embeddings by constructing vectorizations of words through addition of their n-gram embeddings, allowing for non-zero vectorizations of out-of-vocabulary words. FastText requires training strictly on Hindi data in order to achieve testing capability on Hindi.
- *LASER* (Schwenk and Douze, 2017), a cross-linguistic word embedding model that allows for training in one language or domain and evaluation on a different set of linguistic data through relational multi-lingual translations.

The input to the neural network models is text data in the form of sentence sequences, pre-processed with tokenization and vectorization of the tokens in order to create $(S \times E)$-sized input frames, where S

[4]These models are not directly comparable to the results in Tables 5.2 and **??** because the models used different parameters.

is the standardized sentence length, taken as the mean sentence length from the aggregate data, and E is the embedding size. No lemmatization was performed on the Hindi data due to the lack of publicly available Hindi lemmatizers.

Classifiers for Hindi data

For the baseline models, we use the same baseline classifiers as used for English: naïve Bayes, SVM, and a feed-forward neural network. We also use several neural network models, including variations of LSTM, GRU, and CNN architectures, developed using PyTorch. The models we use are as follows:

- Temporal convolutional network (TCN) models, including CNNText, SeqCNN1D, and SeqCNN2D. These models convolve 1D and 2D kernels over each sequence of word embeddings in order to compose together spatio-temporal relationships between words across small and large distances by adjusting the stride. These models are based on the CNN sentence classification models from Kim (2014).
- RNN model architectures, consisting of traditional LSTM and GRU sequential networks. The LSTM and GRU units act as "memory cells" to store and forget relevant long-term dependencies across sentence data, being able to hypothetically capture nuances of relevant information from context.

The CNNText and SeqCNN variants possess the same general structure. Each sentence is represented as a vector of N words from a vocabulary ordered in the sequence they appear in the sentence, and then is converted into an input representation of size $(N \times M)$ where each row n_i is the M-dimensional embedding of the respective word in the sentence. The input is then passed through distinct sets of r convolution filters, which independently produce r different outputs, which are concatenated and fed through a fully-connected layer before being classified. The primary difference between the two models lies in the number of convolution layers and the presence of batch normalization. Furthermore, SeqCNN variants (1D and 2D) differ in the kernel size, with the 2D variant including convolutions between sentences in the same block of text data, not simply within sentences.

The vanilla LSTM and GRU models function as baselines in relation to the CNNText and SeqCNN models. Both model architectures were divided between model variants that trained on LASER or on FastText, in order to investigate the effect and utility of cross-linguistic training with regards to the hyperspecificity of Hindi medical data. Within the LASER-based models, we have English-trained LASER models and Hindi-trained LASER models.

For CNNText, the principal and experimentally significant hyperparameter was the dropout rate, for which we tested three values: 0.0, 0.2, 0.4. The CNNText kernel size was found to have little effect in variation of metrics with regards to the specificity of its architecture. Kernel size, however, was the primary hyperparameter with regards to the SeqCNN-1D models, with the 1D models offering a $(1 \times k)$ horizontal sequence length kernel while the 2D models use a two-dimensional $(k \times k)$ window across and between text sequences.

Using a small subset of our data, we performed parameter optimization using the hyperopt library (Bergstra et al., 2013) for 100 runs for each type of classifier, in order to determine the best model parameters. Hyperoptimized parameters included the number of epochs, batch size, and number of hidden units. We split the dataset into 70% train, 20% test, and 10% validation. Finally, we compared the performance of the optimized version of each model.

Model	Precision	Recall	F_1	CSMFA
Naïve Bayes	.313	.424	.337	.603
Random forest	.417	.492	.425	.655
SVM	.421	.361	.365	.804
CNN	**.657**	**.616**	**.618**	**.884**

Table 5.4: Results of non-neural models on classifying translated+original Hindi narratives.

Model	Data	Conv Kernel Size	Test Acc.	F_1	CSMFA
SeqCNN2D-64	FT	64	.310	.178	.388
SeqCNN2D-128	FT	128	**.330**	.152	**.392**
SeqCNN2D-256	FT	256	.310	.178	.388
SeqCNN2D-64	LASER Eng-trained	64	.190	.018	.210
SeqCNN2D-128	LASER Eng-trained	128	.260	**.182**	.324
SeqCNN2D-256	LASER Eng-trained	256	.170	.106	.267
SeqCNN1D-64	LASER Eng-trained	64	.080	.002	.101
SeqCNN1D-128	LASER Eng-trained	128	.150	.019	.174
SeqCNN1D-256	LASER Eng-trained	256	.120	.017	.163

Table 5.5: Results of the neural SeqCNN models on the Hindi dataset.

Results on Hindi data

Table 5.4 shows the results of the baseline classifiers on a test set of 100 Hindi narratives. The training set is the remaining 400 Hindi narratives plus the English narratives translated into Hindi.

Table 5.5 shows the results of our classification models with various embeddings and parameters, evaluated at the individual level (accuracy F_1) and the population level (CSMFA).

The basic CNN model performs the best on the Hindi data. While the scores are not quite as high as for the English data, we note that these models depend on translated data, which may have errors, and on embeddings which were trained on general domain data rather than medical data. At the time of writing we were unable to find a suitable publicly-available corpus of medical Hindi.

5.4.3 Results on clinical dialogues

For the clinical dialogue dataset, we classify conversations as one of 7 primary diagnoses, as shown in Table 2.9. The input to the classification model is the text of all the utterances in the conversation, concatenated together.

As with the VA data, the random classifier assigns a diagnosis category randomly according to the training distribution. The baseline CNN model is a classifier using only the word embeddings of the entire conversation. We compare this model to the timeline-based classification models. The results are shown in Table 5.6.

5.5 Discussion and analysis

For the VA dataset, combining the word embedding and timeline features provided better performance than the timeline features alone, and the end-to-end models performed better than the models without

Event/time tagger	Ordering model	Classification model	Classification features	Precision	Recall	F_1	CSMFA
None	None	Random	None	.472	.200	.279	.500
None	None	CNN	ELMo word emb (reduce to 128 dim)	**1.000**	.733	.846	.750
gold	Linear	CNN (e2e)	timeline (192) + words	.941	**.867**	**.891**	.812
gold	S2S	CNN (e2e)	timeline (192) + words	**1.000**	.800	.887	**.813**

Table 5.6: CoD classification results on the Verilogue dataset using words and ordered events. Each of 10 folds is trained on 150 records and tested on 15 records of adult deaths.

end-to-end training. Although the linear model had better temporal ordering performance, the classification performance was better using the output of the Set2Seq model, in terms of F_1 score (individual level performance) and CSMFA (population level performance). This indicates that the grouped event ordering is useful for downstream tasks even though the linear models scored higher on ordering metrics.

The results from Yan et al. (2019) suggest that adding some of the non-narrative features to the narrative-based model can improve performance. In fact the best results might be obtained by an ensemble of several models, or a combination of non-narrative and narrative features. Although we tested several other non-narrative features in the classification models, most did not make a large difference in performance, likely because they capture information that is already present in the narrative.

These VA classification results are not directly comparable with other automated VA methods for several reasons. First, most current methods use primarily non-narrative data (usually from the WHO form, and the MDS uses a different form), and because no other evaluations have been performed on this particular dataset. Many evaluations use the PHMRC dataset due to its availability; however, it has many uninformative narratives, and it surveys only hospital-based deaths. As discussed in Chapter 1, the distribution of CoDs outside of health facilities can be very different.

On a different subset of MDS data, Miasnikof et al. (2015) reported a maximum recall (sensitivity) of .57 using the naïve Bayes classifier, with lower performance by the Tarrif method and InterVA-4, whereas our baseline neural network method achieved .749 recall.

For the Verilogue dataset, we observe that both timeline-based models outperform the baseline classifier. Although this is a fairly small dataset we are still able to get reasonably high classification accuracy. However, these are just preliminary results for the Verilogue dataset. The transcripts provide some interesting data that is not currently used by the model, such as speaker labels. While VAs typically have a single author, and clinical notes don't always specify who the author of a particular section is, it might matter who the speaker is in clinical conversations when it comes to diagnosis. The type of information mentioned by the doctor might be very different from what is said by the patient. In addition, modality can play a crucial role. For example, if a physician mentions a symptom, they might be asking the patient if they've experienced that symptom, and the patient might say no. In that case, that symptom is not relevant to the diagnosis (or perhaps the absence of that symptom could be relevant to the diagnosis). There is also conversational structure in the dialogue transcripts that could provide additional information to a classifier, and this structure has not yet been harnessed by the existing models.

Although large transformer-based models such as BERT work well for many tasks, it is unlikely that the current datasets are large enough to take full advantage of such architectures. However, a model pre-trained on medical data might provide better performance, and this is an avenue for future work.

The current models are highly dependent upon the event and time embeddings that are learned from the training data. It is possible that the representation could be improved with more temporal training data and some hyperoptimization. In addition, all of these models would benefit from larger datasets of in-domain data, especially for non-English narratives.

5.6 Conclusion

We have explored several different classification models for different types of medical narratives. Disease and CoD classification can use different types of features, including word embeddings, character embeddings, features from non-narrative data, and temporally ordered event embeddings. We obtain the best results by combining basic word embedding features with the timeline representation generated by temporal ordering models.

While here we have only classified VA records into broad CoD categories, more specific categories could be predicted with sufficient training data. These models are a useful first step, but more development is needed in order for these models to be maximally useful in a clinical setting. Ideally for verbal autopsies the model would predict individual ICD-10 codes instead of broad categories. It would also be helpful for the model to report a confidence score along with each prediction in order to flag low-confidence output for review by a human.

We have demonstrated that information beyond just words and context can be useful for automated classification. Especially since time information can be very important in a clinical context, adding such information to classification models can make them more accurate. In the future, such models could incorporate further temporal information such as duration and event overlap, which could improve the temporal representation and thus further improve classification accuracy.

Chapter 6

Explainability for disease classifiers

6.1 Introduction

Now that we have examined models for temporal ordering and classification of medical narratives, we want to investigate the performance of these models in order to understand *why* the model gives the output it does for a specific input example. In this chapter, we aim to provide automated explanations for the classifier decisions based on the extracted temporal ordering, as well as for the temporal ordering itself.

Machine learning models can provide useful data analysis and automated prediction tools for the medical domain. However, it is critical for the users of such clinical systems to know whether the output is reliable or not. A system's ability to provide a rationale for its output can be an important factor for real-world adoption, especially in the clinical domain. Since a clinician's decisions directly affect their patients' health, they need to know that they can trust the output of any machine learning models that they use in practice. The goal of automated rationale (or explanation) generation is to provide evidence to a clinician that both the automated predictions and the explanations for those predictions are reliable.

Although there is not a single agreed-upon definition of **explainability** or **interpretability** in a machine learning context, Serrano and Smith (2019) state that "in order for a model to be interpretable, it must not only suggest explanations that make sense to people, but also ensure that those explanations accurately represent the true reasons for the model's decision". We adopt this definition of explainability — namely that the explanations we provide must be human-interpretable and also faithfully represent how the input influenced the classifier's output.

Explainability tools can take several different approaches. Some methods aim to investigate the inner workings of the network to determine which parts of the network are responsible for certain decisions. This is sometimes called "network dissection", or saliency-based methods. An alternative is to look at the input itself and assign a weighting to the input based on which features the model focused on the most. Models that employ an attention mechanism (Bahdanau et al., 2015) can display a visual representation of the weights learned by the model for a specific example so that the user can see whether they roughly correspond to the most important parts of the input. At a higher level, "perceptive interpretability" models aim to provide a human-understandable rationale such as a word or phrase or a visual highlighting of the data (Tjoa and Guan, 2019). Some tools look for rule-based patterns in how independent features map to the output class.

Evaluation of such explainability methods is still an open problem. Model explanations are often provided as qualitative examples rather than as part of a quantitative evaluation. As a first step toward evaluating explanations for CoD classification, we make the following contributions:

- We apply a gradient-based attribution method (integrated gradients) to a CoD classification model to identify which parts of the text are the most influential for diagnosing CoD.
- We evaluate the gradient-based attributions by comparing the selected rationale text to physician-generated keyword phrases for verbal autopsy records using cosine similarity.

6.2 Related Work

Explainability tools come in many different forms, from "network dissection" methods, which focus on understanding a model's inner workings, to "perceptive interpretability" methods, which focus on providing human-understandable rationales (Tjoa and Guan, 2019).

Sundararajan et al. (2017) introduced integrated gradients for extracting attribution vectors for neural networks. Integrated gradients are defined as "the path integral of the gradients along the straightline path from the baseline x' to the input x." The integral is calculated via a Riemman approximation with m steps. They argue that integrated gradients are a more suitable attribution method than previous work such as DeepLift (Shrikumar et al., 2017) and Layer-wise relevance propagation (LRP) (Bach et al., 2015) because they satisfy several axioms such as sensitivity, implementation invariance, and symmetry preservation. Because of these desirable properties, we opt to use integrated gradients for our work.

On the perceptive interpretability side, Ribeiro et al. (2016) presented Local Interpretable Model-agnostic Explanations (LIME), a method which aims to provide a locally interpretable approximation of a decision function for explaining the output of a classifier. For text classification they used bag-of-words features as the interpretable representation. They evaluated the faithfulness of the explanations by comparing to interpretable models such as decision trees and sparse logistic regression classifiers, and found that the explanations achieved a recall of .90 or above for important features.

Lei et al. (2016) performed unsupervised rationale extraction by using a generator model to identify spans of text as candidate rationales, and an encoder model to encode them and predict labels from the candidates. The goal is to identify sequences of the original text that are sufficient to produce a similar prediction as the whole input.

AllenNLP Interpret (Wallace et al., 2019) is a freely available toolkit for applying interpretation methods to NLP models and developing new interpretation methods. The toolkit provides two main types of interpretations: gradient-based saliency maps and adversarial input modification. The gradient-based methods can be used to determine which parts of the input had the greatest impact on the model's loss calculation. The input reduction method aims to reduce the input text to the smallest number of words that still produces the same output. This can be helpful for determining which parts of the text are the main cause of a model's prediction.

Another commonly used strategy for interpreting models is to visualize the attention mechanism weights from a model or to apply adversarial attacks (Wallace et al., 2019). Attention visualizations have been popular for models in the medical domain because they are straightforward to implement. Mullenbach et al. (2018) used an attentional convolution neural network (CNN) to predict ICD-9 (International Classification of Diseases, version 9) diagnosis codes in the MIMIC-II and MIMIC-III clinical

note datasets (Pollard and Johnson, 2016). They extracted the most important 4-gram from the text according to the attention weights, along with the previous and next 5 words for context, and provided these as explanations. They also selected explanations by computing the idf-weighted cosine similarity between the stemmed explanations and stemmed ICD-9 description for the target code, and choosing the explanation with the highest similarity score. In a human evaluation, 88% of the explanations generated by each method were selected by a physician as "informative" or "highly informative".

However, there has been some debate as to whether examining attention weights provides sufficient explanation for a model's output. Clark et al. (2019) analyzed attention weights in BERT and found that particular attention heads learn to capture different syntactic aspects of the text. For example, certain attention heads correspond to syntactic relations such as direct objects of verbs and objects of prepositions with $> 75\%$ accuracy. However, Jain and Wallace (2019) found that the ability of attention weights to identify the relative importance of inputs was inconsistent, and that changes in attention weights did not result in corresponding changes in prediction. In contrast, Wiegreffe and Pinter (2020) found that properly constructed adversarial attention distributions did produce worse performance than the original attention, which suggests that the attention weights do correspond to the model's predictions. Furthermore, it was difficult to compute reliable adversarial attention distributions in the first place. However, Serrano and Smith (2019) found that ranking rationale candidates by the gradient of the decision function produced a better minimal set of explanations than ranking by attention weights.

Since faithfulness is highly important in the medical domain, we opt to use a gradient-based method (integrated gradients).

6.3 Methods

We use the integrated gradients implementation from Captum[1] to calculate the attribution values for the matrix of ELMo word embeddings, using zero vectors as the baseline for computing the integral. We take the exponential of the attribution values, and then scale them from 0 to 1 to use as weights across the embedding matrix. The exponential scaling puts more emphasis on the words that have higher attribution values as opposed to a linear scaling. We then compute the weighted average of the word embeddings for the entire document.

For each key phrase of each example, we average the ELMo embeddings of all the words in the key phrase, and then compute the cosine similarity between the key phrase embedding and the attribution-weighted text embedding. We average the cosine similarity scores over all of the key phrases for a given document, and then we report the average over the test dataset

For comparison, we also compute the cosine similarity between the key phrase embedding and the unweighted mean of the word embeddings of the text (without the attributions). We note that since many of the key phrases are copied directly from the narrative, we expect the unweighted representation to have decent similarity to the key phrases.

6.4 Results

We report the average cosine similarity between the reference key phrases and the attribution-weighted text representation, as well as the similarity of the key phrases to the unweighted text representation

[1] https://captum.ai/

CoD Category	Unweighted embeddings mean (std dev)	Attribution-weighted embeddings	Classifier F1	Records
1	.412 (.040)	**.414** (.043)	.29	4
2	**.390** (.009)	.389 (.012)	.63	3
3	.380 (.043)	**.381** (.044)	.00	5
4	.399 (.027)	**.409** (.034)	.00	4
5	.397 (.040)	**.403** (.042)	.67	8
6	–	–	.00	0
7	.398 (.061)	**.401** (.060)	.00	15
8	.397 (.080)	**.400** (.079)	.88	5
9	.412 (.052)	**.413** (.056)	.67	9
10	**.376** (.046)	.373 (.045)	.25	7
11	.381 (.017)	**.382** (.021)	.31	3
12	**.376** (.010)	.374 (.009)	.67	2
13	.343 (.019)	**.346** (.016)	.44	3
14	.391 (.031)	**.392** (.032)	.33	4
15	.485 (.063)	**.494** (.065)	.94	14
16	.485 (.097)	**.501** (.104)	.76	4
17	.437 (.061)	**.445** (.063)	.50	3
18	.384 (.022)	.384 (.023)	.67	3
Average	.419 (.069)	**.423** (.067)	.60	100

Table 6.1: Cosine similarity scores between the text representation and the physician-generated key phrases.

in Table 6.1, which also shows the F_1 scores of the CoD classifier for each class. The highest F_1 score is 0.94 for Road traffic incidents.

Figure 6.1 shows an example narrative with the words highlighted according to attribution weights, and Table 6.2 shows the cosine similarity for each key phrase for the same example.

6.5 Discussion

By re-weighting the text representation according to attribution values, we achieve a higher similarity with physician-authored keywords than the unweighted text. This gives us some indication that the model attributions are in line with the parts of the text that physicians think are the most important for CoD diagnosis.

In the example visualization, we can see that the model is focusing on important symptoms words such as “vomiting” and “motions”, as well as environmental factors such as “contaminated water”. Note that the key phrase “old age” has a very low similarity score because the patient’s age is not actually mentioned in the narrative.

This brings us to one of the limitations to using human-authored references as an evaluation — because physicians are allowed to write in the key phrases, they sometimes write information that was not in the narrative, but rather came from the demographic or questionnaire data. The key phrases could also reflect extrapolations based on clinical knowledge, which may or may not map directly to the narrative text.

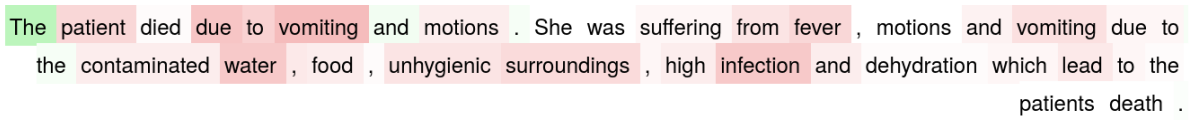

Figure 6.1: Visualization of attribution weights.

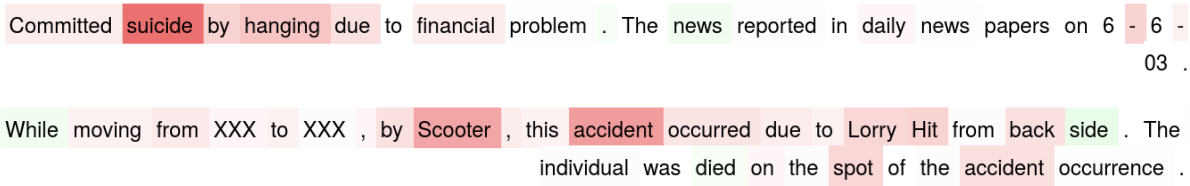

Figure 6.2: Example visualizations of attribution weights, CoDs are "Suicide" and "Road traffic accidents" respectively.

Key phrase	**Cosine similarity (attr-weighted)**	**Cosine similarity (unweighted)**
old age	0.328	0.292
diarrhea	0.361	0.358
vomiting	0.385	0.394
fever due to ingestion of contaminated water and food	0.552	0.563
dehydration	0.391	0.391
loose motion	0.393	0.383
Average	0.402	0.397

Table 6.2: Example of cosine similarity of key phrases for the first example narrative in Figure 6.1.

6.6 Conclusion

In the clinical domain, it is important to verify that the automated explanations the model produces match up with human explanations. We have provided an evaluation of integrated gradient attributions for cause-of-death classification, and shown that these explanations are similar to key phrases written by physicians.

Future work could include having physicians extract key phrases from the narrative where they are limited to only selecting portions of the text directly. We could also do a post-hoc human evaluation by having physicians rate the usefulness of the phrases highlighted by the attribution weights.

Most explainability methods have focused on classification or regression models, but in the future we hope to apply such methods to non-classification tasks such as ranking and relation extraction.

Chapter 7

Conclusion and future research directions

7.1 Overview

Although much work has been done on machine learning (ML) for clinical text, there have been two common approaches: use a pre-trained word embedding model (such as word2vec) or language model (such as BERT) to represent the text, and then apply a classifier model and hope that the embeddings capture the kind of information that will be useful for the target problem, or handcraft rules, lexicons, or linguistic features that work well for the task. While handcrafted features can produce systems with higher precision that work better for smaller datasets, and can be more interpretable, it is often time-consuming and difficult to adapt to new domains. On the other hand, models that learn purely from context or embeddings do not necessarily capture specific information such as the value of time phrases which can be helpful for classification.

In this work we have attempted a middle-ground approach, where we specifically extract time information and encode it along with event embeddings that make use of a large pre-trained language model (in this case, ELMo trained on PubMed). This allows us to leverage a large amount of contextual training data for representing events and clinical narratives, while also encoding specific time information, including time phrase type and event chronology. This global listwise ordering approach differs from previous work in that it does not rely on possibly inconsistent pairwise relations, and it is well-suited to data where the time information may be vague or missing. This temporal model provides a simple and useful way of conceptualizing time information, and could be extended to include more fine-grained temporal information such as duration.

The proposed system includes domain-specific supervised event and time extraction, listwise temporal ordering, disease or CoD classification, and an explainability model that can highlight which parts of the text had the most influence on classification.

The main contributions of this work are as follows:

- Experiments with different types of medical narratives that demonstrate that although the language in these narratives can vary widely, in the absence of available data, we can leverage a different type of medical text to improve results (for example, we can use VA data to train a model for

clinical conversations).

- A listwise temporal ordering model for events that allows for simultaneous events. Despite lower performance than the linear ordering model, the timelines generated by the grouped ordering model produced better CoD classification. While previous work has looked at relations between events, very few have explicitly encoded groupings of events in the medical domain. Moreover, we have demonstrated that such grouping information can actually improve downstream classification results, and therefore can be important for automated diagnosis. Although here we have used this model for clinical events, it could be applied to text entities in any domain.

- An embedding-based timeline representation that encodes both event and time information.

- End-to-end disease classification with joint training from raw text, incorporating temporal information and chronology.

- Interpretability mechanisms to explain disease classification based on the input text.

The application of this work to verbal autopsy (VA) survey programs such as the Million Death Study (MDS) could result in faster (possibly even near real-time) coding of VA records. Even with a human physician validating every CoD, it would cut the cost of VA coding in half. In fact, one of the main criticisms of the MDS system was the delay between the collection and coding of records (Krishnan et al., 2020). Automated coding can provide a preliminary CoD diagnosis, which could even be shared with the family of the deceased upon the conclusion of the interview.

Although this system is not yet made for deployment in a real-world setting, with the new computer-based survey collection it could process VA records as soon as they are collected. Although several automated VA coding solutions are already in production, these are typically run after the fact and require a specific data format, whereas the classification method presented here can be run on any text documents. This makes it immediately applicable to any VA survey regardless of the country of origin or which form was used.

We have also demonstrated that these narrative-based methods can work for languages other than English. The translation process is another time-consuming and expensive step (which can also introduce errors) that can be eliminated if we train the classification models on the original language narratives.

Reducing the delay and costs for coding VAs would allow CoD statistics to be available more quickly, which can be vital for public health planning. Especially in the case of new disease outbreaks, such death statistics are critical and time-sensitive for controlling outbreaks and identifying disease hotspots.

More generally, time information can provide crucial insights from medical documents. The listwise ordering methods proposed here can be applied to any type of medical text, and avoid some of the problems of traditional pairwise temporal relation extraction methods. We have demonstrated that the extracted time information and chronology can be used as input to downstream tasks such as classification. In principle, temporal information could be useful for a variety of other clinical NLP tasks, and the models presented here could easily be applied to other domains and tasks.

However, this work is merely a step forward towards using temporal information for disease classification. In the rest of this chapter, we will discuss some limitations to the current work, as well as open questions and future research directions.

7.2 Limitations

There are some limitations to this work, both technical and practical. First of all, this is not yet a production-ready system (although with some software development it could become one). Using these models in production has certain computational hardware and software requirements which are not always available in hospital or other public health systems. For example, we typically train the models on high-performance graphics processing units (GPUs), which are not always available on hospital systems. In addition, loading the embedding models sometimes requires more memory than a typical computer has. To ensure that ML models run quickly enough to be practical to use, specific computational hardware may need to be set up. As an alternative, models could be deployed to cloud computing platforms such as Amazon Web Services (AWS); however, some institutions are hesitant to send data outside of their own systems due to privacy and security concerns. Alternatively (or perhaps additionally), models that have lower memory and processor requirements could be beneficial for deploying to healthcare settings, and could also reduce both the computational costs and environmental impact.

The neural models we use for temporal ordering and classification have some inherent limitations. For example, the RNN architectures such as GRU and LSTM layers, while able to capture context via memory cells that store information from previous items in a sequence, are in practice only able to retain recent information. This means that they are useful for capturing relationships between entities that are relatively close to each other in the text, but less useful for capturing long-range dependencies.

The grouped (Set2Seq) ordering model, while it is able to capture event overlap, must learn to group events and order them at the same time. The model architecture is also more complex, which could be a contributing factor in the lower performance of the Set2Seq model compared to the linear ordering model. An alternative could be to train a model to first group events together correctly, and then train a second model to order the groups. This requires a two-phase approach, but could be done with a simpler model architecture.

With the modern availability of high-power computing resources and large datasets, there is a lot of interest in leveraging the power of ML to improve healthcare. However, concerns have been raised over data privacy issues with regards to making medical datasets available for use in ML models. Medical data can contain highly sensitive information, and there are many regulations surrounding how it can be stored and shared. In addition, ML models learn and store information from data, and it is possible that the models themselves contain sensitive information that could be extracted. Therefore it is unclear whether such models can be shared publicly after training, even if the dataset is kept private.

These concerns make it much more difficult to share and obtain large volumes of medical data. In fact, several studies have shown that even some "anonymized" health records can be re-identified with a few key pieces of information (Culnane et al., 2017). In fact, it is difficult to fully prevent re-identification without removing crucial information (Abdalla et al., 2020). For temporal ordering work specifically, the models rely on having correct time information available. However, some de-identification methods scramble or remove time information due to the re-identification risk. Therefore it can be difficult to find a balance of useful information and patient privacy.

In the case of verbal autopsies, it would be useful to have a large, freely-available dataset of VA records with narratives, especially if such a dataset included ground truth CoD data such as physical autopsy results. Although the PHMRC dataset includes medically certified CoDs, many of the narratives do not have useful information, and the dataset is limited in size.

Once usable data has been acquired, most current models require some level of annotation, which

requires domain expertise as well as access to the data. While in other domains, data can be cheaply annotated by students or crowd-sourcing methods such as Amazon Mechanical Turk, medical data typically requires more understanding of the clinical domain. Because of this, annotators are more expensive, and often must sign a data use agreement or go through specific training in order to be able to access the data.

These restrictions make it more difficult to apply cutting edge models quickly to health data. However, it is important that privacy not be sacrificed for the sake of speed. And while other domains might offer more data and annotations, healthcare data poses unique challenges and opportunities for natural language processing.

In addition, the lack of computing infrastructure in healthcare may indicate that more work is needed to reduce the computational requirements of ML models, so that they can more easily be applied to low-resource domains where such models could have a positive impact.

7.3 Future research directions

Temporal information extraction is still an open problem, especially in the medical domain. While we have examined some initial listwise ordering methods, and presented a small annotated dataset, there are many avenues for future research.

7.3.1 Comparing pairwise and listwise ordering

Since most previous work on temporal ordering has focused on pairwise ordering, it is difficult to compare our listwise methods directly to previous work. Although it is possible to convert a graph of pairwise relations to a listwise ordering, as we have done for the THYME dataset, the conversion is not foolproof. In addition to dealing with conflicting relations, the conversion depends on whether we want to order events by start time, end time, or some other criteria. Moreover, the pairwise metrics reported by previous work are not the most appropriate for listwise methods. Future work could include finding a better way to automate the conversion from pairwise relation annotations to listwise annotations, in order to make use of already-annotated datasets.

7.3.2 Temporal information

For this work we used only simple temporal relation sets such as BEFORE, AFTER, and OVERLAP. While listwise temporal ordering models are simpler to work with and less error-prone than pairwise models, they do not capture partial overlap of events (at least with the current setup). For events with a known duration, adding such duration information to the timeline could help to model partial overlap. In fact, every interval can be specified either by two endpoints, or by a start point and a duration. By adding this information we can still sort the events by start time, but also infer partial overlap from the duration information where it is available.

In addition, we could encode domain-specific temporal constraint knowledge when conducting temporal ordering or disease classification. For instance, we know that nearly all symptoms must occur before death, that “postoperative” means after surgery, etc. Applying some simple inference rules could help to make sure that events are ordered correctly.

As noted in Chapter 5, the neural models we use are also very dependent upon the representation of the data. Most embedding methods, including ELMo, are trained to represent linguistic context for language modeling, which does not necessarily capture the temporal value of time expressions. A timex can convey multiple pieces of information, including the time value, time type, and the exactness of the time phrase. It is unclear whether the current embedding representations accurately convey this information to the model, and explicitly encoding such information (as we did with the time type) may further improve the representation. For example, if we had explicit duration information for each interval, it could be encoded along with the timex's start point and time type.

7.3.3 Metrics for listwise temporal ordering

In Chapter 4 we presented several metrics for evaluating listwise temporal ordering models, including both absolute metrics like mean squared error (MSE), and relative metrics such as pairwise ordering accuracy (POA) and Kendall's τ. For this task, relative ordering accuracy is more important, but there could be tasks for which absolute rank values are more important.

However we do not have a single metric that captures all desirable aspects of a good listwise ordering, including the correctness of the grouping. While Kendall's τ is probably the closest to the kind of metric we want, we cannot optimize the models according to τ because it is not differentiable, and neural models require a differentiable loss function in order to perform backpropagation for training.

For the grouped temporal ordering model, current metrics do not explicitly capture whether the model is grouping the appropriate number of events together. We report events per rank (EPR) for the reference labels and predicted labels, with the goal that the predicted EPR should be similar to the reference EPR. However we do not currently have a way to train the model to minimize the difference between the EPR measures.

The EPR can also affect other metrics – in parameter tuning experiments we have observed that a higher predicted EPR value corresponds to higher values in other metrics, such as τ and POA. This could be because if more events are grouped together, there are fewer events that are explicitly mis-ordered. For grouped models it is important to have a metric that includes the group size, to avoid artificially high scores that may result from grouping too many events together.

7.3.4 Multi-document input

At present, the temporal ordering and classification models we have presented here operate only on a single document. However, in the case of clinical notes, there is usually a longer patient history available. Instead of processing only the current note, it may be helpful to include information from previous clinical notes for the same patient. In that case, temporal information could be especially useful in contextualizing past vs. present events. Fortunately, clinical notes almost always have a document creation time (DCT), which means that even in the absence of temporal information in the note itself, we can map each note onto an absolute timeline. If, for example, we wanted to create a summarization system that would generate a medical history timeline for each patient, we would want to include information from all of the patient's visits. Such summarization could save physicians valuable time when reviewing a patient's EHR.

7.3.5 Non-English languages

Although the majority of current NLP research is conducted on English data, including resources such as embedding models and pre-trained language models, many datasets and situations exist that require work in other languages. We have attempted to extend this work to Hindi, and hope to expand it to other languages in the future.

In the absence of sufficient data in the source language, it is possible that we could leverage training data in English and adapt the models to other languages. One possibility is to use cross-lingual embeddings such as LASER (Schwenk and Douze, 2017), which learn embeddings for multiple languages in the same vector space. However, the currently available LASER models are not trained on medical data.

Another possibility is to map embeddings in one language onto the embedding space of another language. Although this has been successfully done for more closely related languages like English and French, or English and German (Artetxe et al., 2018), when applying these methods to our English word2vec embeddings, we were unable to get a usable alignment to other languages. However, this may be simply a need for optimization of the embedding methods, or much larger training datasets.

7.3.6 Predicting individual ICD-10 codes

In this work, we predicted a small number of broad CoD categories. However, physicians typically assign individual ICD-10 codes to VA records. In addition, clinical notes can have a very wide variety of diagnosis or medical billing codes. Since there can be thousands of such diagnoses, predicting them using supervised classification can be difficult because of the sparsity of the labels in the dataset. A purely supervised classifier can only predict labels it has seen during training.

There are several possibilities for finer-grained disease classification. Clinical resources could be incorporated along with some rule-based logic to match documents to a diagnosis. For example, we could use the ICD-10 manual, which has descriptions of each code, or another clinical resource such as MetaMap. This approach requires more time and effort to curate the resources and rule-based logic. This approach is also more difficult for VA records which only contain descriptions of symptoms, not of the cause of death. The MDS coding manual contains some descriptions of positive and negative indicators for each major CoD, but this will also require some reasoning to match the phrases in the narrative to the corresponding descriptions in the manual.

Alternatively, with enough data we could learn label embeddings (Singh-Miller and Collins, 2009), that map discrete labels onto a continuous label space, allowing the model to predict labels not seen in the training data. But this requires sufficient data and a good label embedding method to ensure that the label representation will result in an accurate ICD-10 code.

7.3.7 Explainability methods for temporal ordering

Many explainability methods have been used as qualitative examples to help users understand a system's output. However since accuracy is paramount in the medical field, it is also important to evaluate the accuracy and quality of the provided explanations to ensure that they actually represent the decision-making process of the model. The explanation should not only be understandable to humans, but should also be consistent with the model's output. In Chapter 6 we presented a similarity metric for verifying that the attribution weights of a model are similar to human-selected key phrases. However

the field could benefit from more straightforward metrics and perhaps explicit human evaluation of model-generated explanations.

While existing toolkits allow for mapping the model weights to specific words in the input, for the temporal ordering models we would like to isolate the event vs. time representation, to see whether the model is more focused on the content of the event or the temporal information. In addition, we could examine which events in the chronology have the most influence on the CoD classifier's output.

7.4 Conclusion

Medical narratives such as verbal autopsies and clinical notes contain a wealth of information that cannot be as easily captured by structured variables. Natural language provides an opportunity for explanations and documentation that is understandable to humans, and provides a freedom and flexibility that long, complicated questionnaires do not. While some non-narrative data can be helpful for CoD diagnosis from VA records, narratives are faster to collect, and can be collected in any language using any form. Natural language processing methods and state-of-the-art machine learning allow us to use this free-text data to perform automated CoD coding and disease classification.

Furthermore, temporal information and chronology can be valuable for many types of clinical text classification tasks. In this work we have presented methods for listwise temporal ordering of events, chronology-based classification, and explainability for classifying clinical narratives. Incorporating temporal reasoning into health text processing can help to ensure that such models properly handle chronology. In addition, we can provide key phrases or highlighting of the input text in order to help human users understand the output of these automated classifiers. The interpretability and reliability of such classifiers is paramount for them to be used and trusted by medical professionals.

The current climate of large data analytics and machine learning advances provide an excellent opportunity for further work in this area, which can be used to improve healthcare processes and decision-making. With the proper application of these models, we can improve the lives of both medical professionals and patients.

Appendices

Appendix

Num	Category
01.01	Sepsis
01.02	Acute respiratory infection, including pneumonia
01.03	HIV/AIDS related death
01.04	Diarrheal diseases
01.05	Malaria
01.06	Measles
01.07	Meningitis and encephalitis
01.08	Tetanus
01.09	Pulmonary tuberculosis
01.10	Pertussis
01.11	Haemorrhagic fever
01.99	Other and unspecified infectious disease
02.01	Oral neoplasms
02.02	Digestive neoplasms
02.03	Respiratory neoplasms
02.04	Breast neoplasms
02.05	Female reproductive neoplasms
02.06	Male reproductive neoplasms
02.99	Other and unspecified neoplasms
03.01	Severe anaemia
03.02	Severe malnutrition
03.03	Diabetes mellitus
04.01	Acute cardiac disease
04.02	Stroke
04.03	Sickle cell with crisis
04.99	Other and unspecified cardiac disease
05.01	Chronic obstructive pulmonary disease (COPD)
05.02	Asthma
06.01	Acute abdomen gastrointestinal disorders
06.02	Liver cirrhosis
07.01	Renal failure
08.01	Epilepsy

Table 1: Cause-of-death categories used by the 2012 WHO VA Instrument (1/2).

Num	Category
09.01	Ectopic pregnancy
09.02	Abortion-related death
09.03	Pregnancy-induced hypertension
09.04	Obstetric haemorrhage
09.05	Obstructed labour
09.06	Pregnancy-related sepsis
09.07	Anaemia of pregnancy
09.08	Ruptured uterus
09.99	Other and unspecified maternal cause
10.01	Prematurity
10.02	Birth asphyxia
10.03	Neonatal pneumonia
10.04	Neonatal sepsis
10.05	Neonatal tetanus
10.06	Congenital malformation
10.99	Other and unspecified perinatal cause of death
11.01	Fresh stillbirth
11.02	Macerated stillbirth
12.01	Road traffic accident
12.02	Other transport accident
12.03	Accidental fall
12.04	Accidental drowning and submersion
12.05	Accidental exposure to smoke, fire and flames
12.06	Contact with venomous animals and plants
12.07	Accidental poisoning and exposure to noxious substance
12.08	Intentional self-harm
12.09	Assault
12.10	Exposure to force of nature
12.99	Other and unspecified external cause of death
98	Other and unspecified non-communicable disease
99	Cause of death unknown

Table 2: Cause-of-death categories used by the 2012 WHO VA Instrument (2/2).

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Mohamed Abdalla, Moustafa Abdalla, Frank Rudzicz, and Graeme Hirst. 2020. Using word embeddings to improve the privacy of clinical notes. *Journal of the American Medical Informatics Association*, 27(6):901–907.

Arun K. Aggarwal, Praveen Kumar, Sadbhawna Pandit, and Rajesh Kumar. 2013. Accuracy of WHO Verbal Autopsy Tool in Determining Major Causes of Neonatal Deaths in India. *PLoS ONE*, 8(1):e54865.

Lukasz Aleksandrowicz, Varun Malhotra, Rajesh Dikshit, Rajesh Kumar Prakash C Gupta, Jay Sheth, Suresh Kumar Rathi, Wilson Suraweera, Pierre Miasnikof, Raju Jotkar, Dhirendra Sinha, Shally Awasthi, Prakash Bhatia, and Prabhat Jha. 2014. Performance criteria for verbal autopsy-based systems to estimate national causes of death: Development and application to the Indian Million Death Study. *BMC Medicine*, 12:21.

James F Allen. 1984. Towards a general theory of action and time. *Artificial Intelligence*, 23(2):123–154.

James F Allen and George Ferguson. 1994. Actions and events in interval temporal logic. *Journal of Logic and Computation*, 4(5):531–579.

Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018. Unsupervised Neural Machine Translation. *International Conference on Learning Representations (ICLR) 2018*, 22(1):1–11.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE*, 10(7):e0130140.

Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*.

Andrew L Beam, Benjamin Kompa, Inbar Fried, Nathan Palmer, Xu Shi, Tianxi Cai, and Isaac S. Kohane. 2018. Clinical Concept Embeddings Learned from Massive Sources of Medical Data. *arXiv*, 1804.01486:1–27.

James Bergstra, Dan Yamins, and David D Cox. 2013. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *Proceedings of the 30th International Conference on Machine Learning (ICML 2013)*, pages 115–123.

James A Berkley, Brett S Lowe, Isaiah Mwangi, Thomas Williams, Evasius Bauni, Saleem Mwarumba, Caroline Ngetsa, Mary PE Slack, Sally Njenga, C Anthony Hart, Kathryn Maitland, Mike English,

Kevin Marsh, and J Anthony G Scott. 2005. Bacteremia among children admitted to a rural hospital in kenya. *New England Journal of Medicine*, 352(1):39–47.

Steven Bethard. 2013. ClearTK-TimeML: A minimalist approach to TempEval 2013. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 10–14.

Steven Bethard, Leon Derczynski, Guergana Savova, James Pustejovsky, and Marc Verhagen. 2015. SemEval-2015 Task 6: Clinical TempEval. In *Proceedings ofthe 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 806–814.

Steven Bethard, Guergana Savova, Wei-Te Chen, Leon Derczynski, James Pustejovsky, and Marc Verhagen. 2016. SemEval-2016 Task 12: Clinical TempEval. In *Proceedings of SemEval-2016*, pages 1052–1062.

Steven Bethard, Guergana Savova, Martha Palmer, and James Pustejovsky. 2017. SemEval-2017 task 12: Clinical TempEval. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 565–572, Vancouver, Canada. Association for Computational Linguistics.

Parminder Bhatia, Busra Celikkaya, and Mohammed Khalilia. 2019. Joint Entity Extraction and Assertion Detection for Clinical Text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 954–959.

Robert I. Binnick. 2012. *The Oxford Handbook of Tense and Aspect*. Oxford University Press, Inc.

Alberto Blanco, Alicia Perez, Arantza Casillas, and Daniel Cobos. 2020. Extracting Cause of Death from Verbal Autopsy with Deep Learning interpretable methods. 2194(June):1–13.

Willie Boag and Hassan Kané. 2017. AWE-CM Vectors: Augmenting Word Embeddings with a Clinical Metathesaurus. *arXiv*, 1712.01460.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Andrew Boulle, Daniel Chandramohan, and Peter Weller. 2001. A case study of using artificial neural networks for classifying cause of death from verbal autopsy. *International Journal of Epidemiology*, 30(3):515–520.

Philip Bramsen, Pawan Deshpande, Yoong Keok Lee, and Regina Barzilay. 2006. Finding temporal order in discharge summaries. In *AMIA Annual Symposium Proceedings, 2006*, pages 81–85.

Peter Byass, Daniel Chandramohan, Samuel Clark, Lucia D'Ambruoso, Edward Fottrell, Wendy Graham, Abraham Herbst, Abraham Hodgson, Sennen Hounton, Kathleen Kahn, Anand Krishnan, Jordana Leitao, Frank Odhiambo, Osman Sankoh, and Stephen Tollman. 2012. Strengthening standardised interpretation of verbal autopsy data: The new InterVA-4 tool. *Global Health Action*, 5:19281.

Peter Byass, Laith Hussain-Alkhateeb, Lucia D'Ambruoso, Samuel Clark, Justine Davies, Edward Fottrell, Jon Bird, Chodziwadziwa Kabudula, Stephen Tollman, Kathleen Kahn, Linus Schiöler, and Max Petzold. 2019. An integrated approach to processing WHO-2016 verbal autopsy data: The InterVA-5 model. *BMC Medicine*, 17(1):1–12.

Xiangrui Cai, Jinyang Gao, Kee Yuan Ngiam, Beng Chin Ooi, Ying Zhang, and Xiaojie Yuan. 2018. Medical Concept Embedding with Time-Aware Attention. *arXiv*, 1806.02873.

T Caliński and J Harabasz. 1974. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3:1–27.

Leonardo Campillos, Louise Deléger, Cyril Grouin, Thierry Hamon, Anne-Laure Ligozat, and Aurélie

Névéol. 2018. A French clinical corpus with comprehensive semantic annotations: Development of the Medical Entity and Relation LIMSI annOtated Text corpus (MERLOT). *Language Resources and Evaluation*, 52:571–601.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to Rank : From Pairwise Approach to Listwise Approach. *Proceedings of the 24th International Conference on Machine Learning*, pages 129–136.

Nathanael Chambers, Taylor Cassidy, Bill McDowell, and Steven Bethard. 2014. Dense Event Ordering with a Multi-Pass Architecture. *Transactions of the Association of Computational Linguistics*, 2(1):273–284.

Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised Learning of Narrative Event Chains. In *ACL-08: HLT - 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 789–797.

Angel X. Chang and Christopher D. Manning. 2012. SUTIME: A library for recognizing and normalizing time expressions. *Proceedings of the 8th International Conference on Language Resources and Evaluation, LREC 2012*, (iii):3735–3740.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Youngduck Choi, Chill Yi-I Chiu, and David Sontag. 2016. Learning Low-Dimensional Representations of Medical Concepts. In *AMIA Summits on Translational Science Proceedings*, page 41. American Medical Informatics Association.

François Chollet. 2015. Keras. https://github.com/fchollet/keras.

Hafizur Rahman Chowdhury, Abraham D. Flaxman, Jonathan C. Joseph, Riley H. Hazard, Nurul Alam, Ian Douglas Riley, and Alan D. Lopez. 2019. Robustness of the Tariff method for diagnosing verbal autopsies: Impact of additional site data on the relationship between symptom and cause. *BMC Medical Research Methodology*, 19(1):1–10.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What Does BERT Look At? An Analysis of BERT's Attention. In *BlackBoxNLP 2019*.

Chris Culnane, Benjamin I. P. Rubinstein, and Vanessa Teague. 2017. Health Data in an Open World. *arXiv*, 1712.05627.

Samuel Danso, Eric Atwell, and Owen Johnson. 2013a. A comparative study of machine learning methods for verbal autopsy text classification. *International Journal of Computer Science Issues*, 10(6).

Samuel Danso, Eric Atwell, and Owen Johnson. 2013b. Linguistic and statistically derived features for cause of death prediction from verbal autopsy text. In *Language Processing and Knowledge in the Web*, pages 47–60. Springer Berlin Heidelberg.

Pascal Denis and Philippe Muller. 2011. Predicting globally-coherent temporal structures from texts via endpoint inference and graph decomposition. *IJCAI International Joint Conference on Artificial Intelligence*, pages 1788–1793.

Leon Derczynski. 2016. Representation and learning of temporal relations. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1937–1948. The COLING 2016 Organizing Committee.

Leon Derczynski. 2017. *Automatically Ordering Events and Times in Text*, volume 677 of *Studies in Computational Intelligence*. Springer International Publishing.

Nikita Desai, Lukasz Aleksandrowicz, Pierre Miasnikof, Ying Lu, Jordana Leitao, Peter Byass, Stephen Tollman, Paul Mee, Dewan Alam, Suresh Kumar Rathi, Abhishek Singh, Rajesh Kumar, Faujdar Ram, and Prabhat Jha. 2014. Performance of four computer-coded verbal autopsy methods for cause of death assignment compared with physician coding on 24,000 deaths in low- and middle-income countries. *BMC Medicine*, 12:20.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *International Journal of Computer Science Issues*, 10(6).

Dmitriy Dligach, Timothy Miller, Chen Lin, Steven Bethard, and Guergana Savova. 2017. Neural temporal relation extraction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 746–751. Association for Computational Linguistics.

Nan Du, Kai Chen, Anjuli Kannan, Linh Tran, Yuhui Chen, and Izhak Shafran. 2019. Extracting Symptoms and their Status from Clinical Conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 915–925.

Lisa Ferro and Inderjeet Mani. 2001. TIDES temporal annotation guidelines. *ACM Transactions on Asian Language Information Processing - TALIP*.

Samuel G. Finlayson, Paea LePendu, and Nigam H. Shah. 2014. Building the graph of medicine from millions of clinical narratives. *Scientific Data*, 1(140032):1–9.

Abraham D. Flaxman, Riley Hazard, Ian Riley, Alan D. Lopez, and Christopher J.L. Murray. 2020. Born to fail: flaws in replication design produce intended results. *BMC medicine*, 18(1):73.

Abraham D. Flaxman, Jonathan C. Joseph, Christopher J.L. Murray, Ian Douglas Riley, and Alan D. Lopez. 2018. Performance of InSilicoVA for assigning causes of death to verbal autopsies: Multisite validation study using clinical diagnostic gold standards. *BMC Medicine*, 16(1):56.

Abraham D Flaxman, Peter T Serina, Bernardo Hernandez, Christopher JL Murray, Ian Riley, and Alan D Lopez. 2015. Measuring causes of death in populations: a new metric that corrects cause-specific mortality fractions for chance. *Population Health Metrics*, 13:28.

Christian Freksa. 1992. Temporal reasoning based on semi-intervals. *Artificial Intelligence*, 54(1):199 – 227.

Carol Friedman, Philip O. Alderson, John H.M. Austin, James J. Cimino, and Stephen B. Johnson. 1994. A general natural-language text processor for clinical radiology. *Journal of the American Medical Informatics Association*, 1(2):161–174.

Diana Galvan, Naoaki Okazaki, Koji Matsuda, and Kentaro Inui. 2018. Investigating the Challenges of Temporal Relation Extraction from Clinical Text. In *Proceedings ofthe 9th International Workshop on Health Text Mining and Information Analysis (LOUHI 2018)*, pages 55–64.

Mireille Gomes, Rehana Begum, Prabha Sati, Rajesh Dikshit, Prakash C Gupta, Rajesh Kumar, Jay Sheth, Asad Habib, and Prabhat Jha. 2017. Nationwide mortality studies to quantify causes of death: Relevant lessons from India's Million Death Study. *Health Affairs*, 36(11):1887–1895.

Tanya Goyal and Greg Durrett. 2019. Embedding time expressions for deep temporal ordering models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4400–4406, Florence, Italy. Association for Computational Linguistics.

David Graff. 2002. The AQUAINT Corpus of English News Text LDC2002T31. *Linguistic Data Consortium*.

Alex Graves. 2012. Supervised Sequence Labelling. *arXiv*, 1308.0850v1:5–13.

Henk Harkema, John N. Dowling, Tyler Thornblade, and Wendy W. Chapman. 2009. ConText: An algorithm for determining negation, experiencer, and temporal status from clinical reports. *Journal of Biomedical Informatics*, 42(5):839–851.

María Herrero-Zazo, Isabel Segura-Bedmar, Paloma Martínez, and Thierry Declerck. 2013. The ddi corpus: An annotated corpus with pharmacological substances and drug–drug interactions. *Journal of Biomedical Informatics*, 46(5):914 – 920.

Kexin Huang, Abhishek Singh, Sitong Chen, Edward T. Moseley, Chih-ying Deng, Naomi George, and Charlotta Lindvall. 2019. Clinical XLNet: Modeling Sequential Clinical Notes and Predicting Prolonged Mechanical Ventilation.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. *arXiv*, 1902.10186.

Spencer L James, Abraham D Flaxman, and Christopher JL Murray. 2011. Performance of the Tariff Method: Validation of a simple additive algorithm for analysis of verbal autopsies. *Population Health Metrics*, 9(1):31–47.

Iñigo Jauregi Unanue, Ehsan Zare Borzeshi, and Massimo Piccardi. 2017. Recurrent neural networks with specialized word embeddings for health-domain named-entity recognition. *Journal of Biomedical Informatics*, 76(June):102–109.

Serena Jeblee, Mireille Gomes, and Graeme Hirst. 2018. Multi-task learning for interpretable cause-of-death classification using key phrase prediction. In *Proceedings of BioNLP 2018 Workshop*, pages 1–6, Melbourne, Australia. Association for Computational Linguistics.

Serena Jeblee, Mireille Gomes, Prabhat Jha, Frank Rudzicz, and Graeme Hirst. 2019a. Automatically determining cause of death from verbal autopsy narratives. *BMC Medical Informatics*, 9(127).

Serena Jeblee and Graeme Hirst. 2018. Listwise temporal ordering of events in clinical notes. In *Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis*, pages 177–182, Brussels, Belgium. Association for Computational Linguistics.

Serena Jeblee, Faiza Khan Khattak, Noah Crampton, Muhammad Mamdani, and Frank Rudzicz. 2019b. Extracting relevant information from physician-patient dialogues for automated clinical note taking. In *Proceedings of the Tenth International Workshop on Health Text Mining and Information Analysis (LOUHI 2019)*, pages 65–74.

Prabhat Jha. 2014. Reliable direct measurement of causes of death in low- and middle-income countries. *BMC Medicine*, 12:19.

Prabhat Jha, Dinesh Kumar, Rajesh Dikshit, Atul Budukh, Rehana Begum, Prabha Sati, Patrycja Kolpak, Richard Wen, Shyamsundar J. Raithatha, Utkarsh Shah, Zehang Richard Li, Lukasz Aleksandrowicz, Prakash Shah, Kapila Piyasena, Tyler H. McCormick, Hellen Gelband, and Samuel J. Clark. 2019. Automated versus physician assignment of cause of death for verbal autopsies: Randomized trial of 9374 deaths in 117 villages in India. *BMC Medicine*, 17(1):116.

Chengyue Jiang, Zhonglin Nian, Kaihao Guo, Shanbo Chu, Yinggong Zhao, Libin Shen, Haofen Wang, and Kewei Tu. 2019. Learning Numeral Embeddings. *arXiv*, 2001.00003.

Prateek Jindal and Dan Roth. 2013. Extraction of events and temporal expressions from clinical narratives. *Journal of Biomedical Informatics*, 46(SUPPL.):S13–S19.

Thorsten Joachims. 2006. Training linear SVMs in linear time. In *Proceedings of the 12th ACM SIGKDD*

International Conference on Knowledge Discovery and Data Mining, pages 217–226.

Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. MIMIC-III, a freely accessible critical care database. *Scientific data*, 3:160035.

Faiza Khan Khattak, Serena Jeblee, Chloé Pou-Prom, Mohamed Abdalla, Christopher Meaney, and Frank Rudzicz. 2019. A survey of word embeddings for clinical text. *Journal of Biomedical Informatics: X*, 4(100057).

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Gary King and Ying Lu. 2008. Verbal autopsy methods with multiple causes of death. *Statistical Science*, 23(1):78–91.

Oleksandr Kolomiyets, Steven Bethard, and Marie Francine Moens. 2012. Extracting narrative timelines as temporal dependency structures. In *50th Annual Meeting of the Association for Computational Linguistics, (ACL 2012)*, volume 1, pages 88–97.

Oleksandr Kolomiyets and Marie Francine Moens. 2013. KUL: A data-driven approach to temporal parsing of documents. **SEM 2013 - 2nd Joint Conference on Lexical and Computational Semantics*, 2(SemEval):83–87.

Zeljko Kraljevic, Daniel Bean, Aurelie Mascio, Lukasz Roguski, Amos Folarin, Angus Roberts, Rebecca Bendayan, and Richard Dobson. 2019. MedCAT – Medical concept annotation tool. *arXiv*, 1912.10166.

Klaus Krippendorff. 2004. *Content Analysis: An Introduction to Its Methodology, Chapter 11*. Sage, Beverly Hills, CA, USA.

Anand Krishnan, Vivek Gupta, Baridalyne Nongkynrih, Rakesh Kumar, Ravneet Kaur, Sumit Malhotra, Harshal R Salve, Venkatesh Narayan, and Ayon Gupta. 2020. Mortality in India established through verbal autopsies (MINErVA): Strengthening national mortality surveillance system in India. *Journal of Global Health Research*, pages 1–22.

S. Kullback and R. A. Leibler. 1951. On information and sufficiency. *Ann. Math. Statist.*, 22(1):79–86.

Ash Kumar, Parth Parmar, Serena Jeblee, and Graeme Hirst. 2021. Cause-of-Death Classification of Verbal Autopsies in Hindi. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP) (in submission)*.

Pawan Kumar, Dhanajit Brahma, Harish Karnick, and Piyush Rai. 2019. Deep Attentive Ranking Networks for Learning to Order Sentences. *arXiv*, 2001.00056.

Mirella Lapata. 2006. Automatic evaluation of information ordering: Kendall's Tau. *Computational Linguistics*, 32(4):471–484.

Joseph R. Larsen, Margaret R. Martin, John D. Martin, Peter Kuhn, and James B. Hicks. 2020. Modeling the Onset of Symptoms of COVID-19. *Frontiers in Public Health*, 8:473.

Hee Jin Lee, Yaoyun Zhang, Min Jiang, Jun Xu, Cui Tao, and Hua Xu. 2018. Identifying direct temporal relations between time and events from clinical notes. *BMC Medical Informatics and Decision Making*, 18(49).

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019a. BioBERT: A pretrained biomedical language representation model for biomedical text mining. *International Journal of Computer Science Issues*, 10(6).

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R. Kosiorek, Seungjin Choi, and Yee Whye Teh. 2019b. Set transformer: A framework for attention-based permutation-invariant neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, pages 1–10, Long Beach, California.

Artuur Leeuwenberg and Marie-Francine Moens. 2017. Structured learning for temporal relation extraction from clinical records. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1150–1158, Valencia, Spain. Association for Computational Linguistics.

Artuur Leeuwenberg and Marie-Francine Moens. 2018. Temporal information extraction by predicting relative time-lines. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1237–1246.

Artuur Leeuwenberg and Marie-Francine Moens. 2019. A survey on temporal reasoning for temporal information extraction from text. *Journal of Artificial Intelligence Research*, 66:341–380.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117. Association for Computational Linguistics.

Omer Levy and Yoav Goldberg. 2014. Dependency-Based Word Embeddings. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308.

Zehang Richard Li, Tyler H. McCormick, and Samuel J. Clark. 2020. Non-confirming replication of "Performance of InSilicoVA for assigning causes of death to verbal autopsies: multisite validation study using clinical diagnostic gold standards," by Flaxman et al. *BMC medicine*, 18(1):69.

Chen Lin, Timothy Miller, Dmitriy Dligach, Hadi Amiri, Steven Bethard, and Guergana Savova. 2018. Self-training improves Recurrent Neural Networks performance for Temporal Relation Extraction. In *Proceedings ofthe 9th International Workshop on Health Text Mining and Information Analysis (LOUHI 2018)*, pages 165–176.

Chen Lin, Timothy Miller, Dmitriy Dligach, Steven Bethard, and Guergana Savova. 2017. Representations of time expressions for temporal relation extraction with convolutional neural networks. In *BioNLP 2017*, pages 322–327, Vancouver, Canada,. Association for Computational Linguistics.

Chen Lin, Timothy Miller, Dmitriy Dligach, Steven Bethard, and Guergana Savova. 2019. A BERT-based Universal Model for Both Within- and Cross-sentence Clinical Temporal Relation Extraction. *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, 2:65–71.

Yu Kai Lin, Hsinchun Chen, and Randall A Brown. 2013. MedTime: A temporal information extraction system for clinical narratives. *Journal of Biomedical Informatics*, 46(SUPPL.):S20–S28.

BQ Liu, R Peto, ZM Chen, J Boreham, Y Wu, J Li, TC Campbell, and J Chen. 1998. Emerging tobacco hazards in China: Retrospective proportional mortality study of one million deaths. *BMJ*, 317:1411–1422.

Lajanugen Logeswaran, Honglak Lee, and Dragomir Radev. 2018. Sentence ordering and coherence modeling using recurrent neural networks. In *Thirty-Second AAAI Conference on Artificial Intelligence*, pages 5285–5292.

Rafael Lozano, Alan D Lopez, Charles Atkinson, Mohsen Naghavi, Abraham D Flaxman, and Christopher JL Murray. 2011. Performance of physician-certified verbal autopsies: multisite validation study using clinical diagnostic gold standards. *Population Health Metrics*, 9(32).

Inderjeet Mani, Barry Schiffman, and Jianping Zhang. 2003. Inferring Temporal Ordering of Events in

News. In *NAACL-Short '03 Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 55–57.

Inderjeet Mani, Marc Verhagen, Ben Wellner, Chong Min Lee, and James Pustejovsky. 2006. Machine learning of temporal relations. In *COLING/ACL 2006 - 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 753–760.

Aurelie Mascio, Zeljko Kraljevic, Daniel Bean, Richard Dobson, Robert Stewart, Rebecca Bendayan, and Angus Roberts. 2020. Comparative analysis of text classification approaches in electronic health records. In *Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing*, pages 86–94, Online. Association for Computational Linguistics.

Tyler H McCormick, Zehang Richard Li, Clara Calvert, Amelia C Crampin, Kathleen Kahn, and Samuel Clark. 2016. Probabilistic cause-of-death assignment using verbal autopsies. *Journal of the American Statistical Association*, 111(15):1036–1049.

Eneldo Loza Mencia, Gerard de Melo, and Jinseok Nam. 2016. Medical Concept Embeddings via Labeled Background Corpora. In *Proceedings of the 10th Language Resources and Evaluation Conference*, pages 4629–4636.

Pierre Miasnikof, Vasily Giannakeas, Mireille Gomes, Lukasz Aleksandrowicz, Alexander Y Shestopaloff, Dewan Alam, Stephen Tollman, Akram Samarikhalaj, and Prabhat Jha. 2015. Naïve Bayes classifiers for verbal autopsies: Comparison to physician-based classification for 21,000 child and adult deaths. *BMC Medicine*, 13(1):286–294.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

Timothy Miller, Steven Bethard, Dmitriy Dligach, Chen Lin, and Guergana Savova. 2015. Extracting time expressions from clinical text. In *Proceedings of BioNLP 15*, pages 81–91, Beijing, China. Association for Computational Linguistics.

Timothy A Miller, Steven Bethard, Dmitriy Dligach, Sameer Pradhan, Chen Lin, and Guergana K Savova. 2013. Discovering narrative containers in clinical text. In *Proceedings of the 2013 Workshop on Biomedical Natural Language Processing (BioNLP 2013)*, pages 18–26.

Riccardo Miotto, Li Li, Brian A Kidd, and Joel T Dudley. 2016. Deep Patient: An Unsupervised Representation to Predict the Future of Patients from the Electronic Health Records. *Scientific Reports*, 6(April):1–10.

Ghulam Mujtaba, Liyana Shuib, Ram Gopal Raj, Retnagowri Rajandram, Khairunisa Shaikh, and Mohammed Ali Al-Garadi. 2017. Automatic ICD-10 multi-class classification of cause of death from plaintext autopsy reports through expert-driven feature selection. *PLOS ONE*, 12(2):e0170242.

James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. 2018. Explainable prediction of medical codes from clinical text. In *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 1101–1111.

Christopher JL Murray, Alan D Lopez, Robert Black, Said Mohd Ali Ramesh Ahuja, Abdullah Baqui, Lalit Dandona, Emily Dantzer, Vinita Das, Usha Dhingra, Arup Dutta, Wafaie Fawzi, Abraham D Flaxman, Sara Gómez, Bernardo Hernández, Rohina Joshi, Henry Kalter, Aarti Kumar, Vishwajeet Kumar, Rafael Lozano, Marilla Lucero, Saurabh Mehta, Bruce Neal, Summer Lockett Ohno,

Rajendra Prasad, Devarsetty Praveen, Zul Premji, Dolores Ramírez-Villalobos, Hazel Remolador, Ian Riley, Minerva Romero, Mwanaidi Said, Diozele Sanvictores, Sunil Sazawal, and Veronica Tallo. 2011a. Population health metrics research consortium gold standard verbal autopsy validation study: design, implementation, and development of analysis datasets. *Population Health Metrics*, 9:27.

Christopher J.L. Murray, Rafael Lozano, Abraham D. Flaxman, Alireza Vahdatpour, and Alan D. Lopez. 2011b. Robust metrics for assessing the performance of different verbal autopsy cause assignment methods in validation studies. *Population Health Metrics*, 9(1):28.

Syed Shariyar Murtaza, Patrycja Kolpak, Ayse Bener, and Prabhat Jha. 2018. Automated verbal autopsy classification: Using one-against-all ensemble method and Naïve Bayes classifier. *Gates Open Research*, 2(63).

Marjan Najafabadipour, Massimiliano Zanin, Alejandro Rodríguez-González, Maria Torrente, Beatriz Nuñez García, Juan Luis Cruz Bermudez, Mariano Provencio, and Ernestina Menasalvas. 2020. Reconstructing the Patient's Natural History from Electronic Health Records. *Artificial Intelligence in Medicine*, 105(101860).

Jinseok Nam, Eneldo Loza Mencía, and Johannes Fürnkranz. 2016. All-in Text: Learning Document, Label, and Word Representations Jointly. *Thirtieth AAAI Conference on Artificial Intelligence*, pages 1948–1954.

Erin K. Nichols, Peter Byass, Daniel Chandramohan, Samuel J. Clark, Abraham D. Flaxman, Robert Jakob, Jordana Leitao, Nicolas Maire, Chalapati Rao, Ian Riley, Philip W. Setel, and on behalf of the WHO Verbal Autopsy Working Group. 2018. The WHO 2016 verbal autopsy instrument: An international standard suitable for automated analysis by InterVA, InSilicoVA, and Tariff 2.0. *PLOS Medicine*, 15(1):1–9.

Azadeh Nikfarjam, Ehsan Emadzadeh, and Graciela Gonzalez. 2013. Towards generating a patient's timeline: Extracting temporal relationships from clinical notes. *Journal of Biomedical Informatics*, 46(0):S40–S47.

Qiang Ning, Zhili Feng, and Dan Roth. 2017. A Structured Learning Approach to Temporal Relation Extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1027–1037.

Barak Oshri and Khandwala Nishith. 2015. There and back again: Autoencoders for textual reconstruction. Stanford NLP Course.

S. Pakhomov, B. McInnes, T. Adams, Y. Liu, T Pedersen, and GB Melton. 2010. Semantic similarity and relatedness between clinical terms: An experimental study. In *Proceedings of the Annual Symposium of the American Medical Informatics Association*, pages 572–576.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. In *NIPS 2017 Autodiff Workshop*, pages 1–4.

Kevin Patel, Divya Patel, Mansi Golakiya, Pushpak Bhattacharyya, and Nilesh Birari. 2017. Adapting pre-trained word embeddings for use in medical coding. *BioNLP 2017*, pages 302–306.

Ted Pedersen, Serguei VS Pakhomov, Siddharth Patwardhan, and Christopher G Chute. 2007. Measures of semantic similarity and relatedness in the biomedical domain. *Journal of Biomedical Informatics*, 40(3):288–299.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexan-

dre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12(Oct):2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL-HLT 2018*, pages 2227–2237.

Tom J Pollard and Alistair EW Johnson. 2016. The MIMIC-III Clinical Database. `http://dx.doi.org/10.13026/C2XW26`.

James Pustejovsky, José Castaño, Robert Ingria, Roser Saurí, Robert Gaizauskas, Andrea Setzer, and Graham Katz. 2003. TimeML: Robust specification of event and temporal expressions in text. In *IWCS-5, Fifth International Workshop on Computational Semantics*, pages 1–11.

James Pustejovsky and Amber Stubbs. 2011. Increasing informativeness in temporal annotation. In *Proceedings of the 5th Linguistic Annotation Workshop*, pages 152–160.

James Pustejovsky, Marc Verhagen, Roser Sauri, Jessica Littman, Robert Gaizauskas, Graham Katz, Inderjeet Mani, Robert Knippen, and Andrea Setzer. 2006. *TimeBank 1.2 LDC2006T08*. Linguistic Data Consortium. Web download.

Preethi Raghavan, Eric Fosler-Lussier, Noémie Elhadad, and Albert M. Lai. 2014. Cross-narrative temporal ordering of medical events. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 998–1008, Baltimore, Maryland. Association for Computational Linguistics.

Preethi Raghavan, Eric Fosler-Lussier, and Albert M Lai. 2012. Learning to temporally order medical events in clinical text. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, pages 70–74.

Usha Ram, Rajesh Dikshit, and Prabhat Jha. 2016. Level of evidence of verbal autopsy–authors' reply. *The Lancet Global Health*, 4(6):e368–e9.

Lance Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora*.

Ruth M. Reeves, Ferdo R. Ong, Michael E. Matheny, Joshua C. Denny, Dominik Aronsky, Glenn T. Gobbel, Diane Montella, Theodore Speroff, and Steven H. Brown. 2013. Detecting temporal expressions in medical narratives. *International Journal of Medical Informatics*, 82(2):118–127.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144.

Jeremy Rogers and Olivier Bodenreider. 2008. SNOMED CT: Browsing the Browsers. In *KR-MED*, pages 30–36.

Guergana Savova, James Masanz, Philip Ogren, Jiaping Zheng, Sunghwan Sohn, Karin Kipper-Schuler, and Christopher Chute. 2010. Mayo Clinic Clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications. *JAMIA*, 17:507–513.

Holger Schwenk and Matthijs Douze. 2017. Learning joint multilingual sentence representations with neural machine translation. In *Proceedings of the ACL workshop on Representation Learning for*

NLP, pages 1–11.

Peter Serina, Ian Riley, Bernardo Hernandez, Abraham D Flaxman, Devarsetty Praveen, Veronica Tallo, Rohina Joshi, Diozele Sanvictores, Andrea Stewart, Meghan D Mooney, Christopher J L Murray, and Alan D Lopez. 2016. The paradox of verbal autopsy in cause of death assignment : symptom question unreliability but predictive accuracy. *Population Health Metrics*, pages 10–19.

Peter Serina, Ian Riley, Andrea Stewart, Spencer L. James, Abraham D. Flaxman, Rafael Lozano, Bernardo Hernandez, Meghan D. Mooney, Richard Luning, Robert Black, Ramesh Ahuja, Nurul Alam, Sayed Saidul Alam, Said Mohammed Ali, Charles Atkinson, Abdulla H. Baqui, Hafizur R. Chowdhury, Lalit Dandona, Rakhi Dandona, Emily Dantzer, Gary L. Darmstadt, Vinita Das, Usha Dhingra, Arup Dutta, Wafaie Fawzi, Michael Freeman, Sara Gomez, Hebe N. Gouda, Rohina Joshi, Henry D. Kalter, Aarti Kumar, Vishwajeet Kumar, Marilla Lucero, Seri Maraga, Saurabh Mehta, Bruce Neal, Summer Lockett Ohno, David Phillips, Kelsey Pierce, Rajendra Prasad, Devarsatee Praveen, Zul Premji, Dolores Ramirez-Villalobos, Patricia Rarau, Hazel Remolador, Minerva Romero, Mwanaidi Said, Diozele Sanvictores, Sunil Sazawal, Peter K. Streatfield, Veronica Tallo, Alireza Vadhatpour, Miriam Vano, Christopher J.L. Murray, and Alan D. Lopez. 2015. Improving performance of the Tariff Method for assigning causes of death to verbal autopsies. *BMC Medicine*, 13(1):291.

Sofia Serrano and Noah A. Smith. 2019. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951, Florence, Italy. Association for Computational Linguistics.

Andrea Setzer. 2001. *Temporal information in newswire articles: an annotation scheme and corpus study*. Ph.D. thesis, University of Sheffield.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning Important Features Through Propagating Activation Differences. *34th International Conference on Machine Learning, ICML 2017*, 7:4844–4866.

Yuqi Si, Jingqi Wang, Hua Xu, and Kirk Roberts. 2019. Enhancing Clinical Concept Extraction with Contextual Embedding. *JAMIA*, 26(11):1297–1304.

Natasha Singh-Miller and Michael Collins. 2009. Learning label embeddings for nearest-neighbor multiclass classification with an application to speech recognition. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 1678–1686. Curran Associates, Inc.

Irena Spasic and Goran Nenadic. 2020. Clinical text data in machine learning: Systematic review. *Journal of Medical Internet Research*, 22(3):1–19.

SRS Collaborators of the RGI-CGHR. 2014. *Prospective study of million deaths in India: Technical Document No. VIII: Health care professional's manual for assigning cause of death (COD) based on RHIME household reports*. Registrar General of India (RGI), Centre for Global health Research (CGHR), University of Toronto. `www.cghr.org/mds`.

Jannik Strötgen and Michael Gertz. 2013. Multilingual and cross-domain temporal tagging. *Language Resources and Evaluation*, 47(2):269–298.

William F Styler, IV, Steven Bethard, Sean Finan, Martha Palmer, Sameer Pradhan, Piet C de Groen, Brad Erickson, Timothy Miller, Chen Lin, Guergana Savova, and James Pustejovsky. 2014. Temporal annotation in the clinical domain. *Transactions of the Association for Computational Linguistics*, 2014(2):143–154.

Weiyi Sun, Anna Rumshisky, and Ozlem Uzuner. 2013a. Annotating temporal information in clinical narratives. *Journal of Biomedical Informatics*, 46(0).

Weiyi Sun, Anna Rumshisky, and Ozlem Uzuner. 2013b. Evaluating temporal relations in clinical text: 2012 i2b2 challenge. *Journal of the American Medical Informatics Association*, 20(5):806–813.

Weiyi Sun, Anna Rumshisky, and Ozlem Uzuner. 2013c. Temporal reasoning over clinical text: The state of the art. 20(5):814–819.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. *34th International Conference on Machine Learning, ICML 2017*, 7:5109–5118.

Theano Development Team. 2016. Theano: A Python framework for fast computation of mathematical expressions. *arXiv*, 1605.02688.

Hegler Tissot, Angus Roberts, Leon Derczynski, Genevieve Gorrell, and Marcos Didonet Del Fabro. 2015. Analysis of temporal expressions annotated in clinical notes. In *Proceedings 11th Joint ACL-ISO Workshop on Interoperable Semantic Annotation (isa-11)*, page 93.

Erico Tjoa and Cuntai Guan. 2019. A Survey on Explainable Artificial Intelligence (XAI): Towards Medical XAI.

Julien Tourille. 2018. *Extracting Clinical Event Timelines : Temporal Information Extraction and Coreference Resolution in Electronic Health Records*. Ph.D. thesis, Université Paris-Saclay.

Julien Tourille, Matthieu Doutreligne, Olivier Ferret, Aurélie Névéol, Nicolas Paris, and Xavier Tannier. 2018. Evaluation of a sequence tagging tool for biomedical texts. In *Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis*, pages 193–203, Brussels, Belgium. Association for Computational Linguistics.

Julien Tourille, Olivier Ferret, Aurélie Névéol, and Xavier Tannier. 2017a. Neural architecture for temporal relation extraction: A bi-LSTM approach for detecting narrative containers. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 224–230.

Julien Tourille, Olivier Ferret, Xavier Tannier, and Aurélie Névéol. 2017b. Temporal information extraction from clinical text. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2017)*.

Naushad UzZaman and James F. Allen. 2011. Temporal Evaluation. In *The 49th Annual Meeting of the Association for Computational Linguistics Human Language Technologies*, volume 271, pages 351–356.

Naushad Uzzaman, Hector Llorens, Leon Derczynski, Marc Verhagen, James Allen, and James Pustejovsky. 2013. SemEval-2013 Task 1 : TempEval-3: Evaluating Time Expressions, Events, and Temporal Relations. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, volume 2, pages 1–9.

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

Sumithra Velupillai, Danielle L Mowery, Samir Abdelrahman, Lee Christensen, and Wendy Chapman. 2015. BluLab: Temporal Information Extraction for the 2015 Clinical TempEval Challenge. pages 815–819. Association for Computational Linguistics (ACL).

Marc Verhagen. 2005. Temporal Closure in an Annotation Environment. *Language Resources and Evaluation*, 39(2/3):211–241.

Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Graham Katz, and James Pustejovsky. 2007. SemEval-2007 task 15: TempEval temporal relation identification. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 75–80, Prague, Czech Republic. Association for Computational Linguistics.

Marc Verhagen, Inderjeet Mani, Roser Sauri, Robert Knippen, Seok Bae Jang, Jessica Littman, Anna Rumshisky, John Phillips, and James Pustejovsky. 2005. Automating temporal annotation with TARSQI. In *Proceedings of the ACL 2005 on Interactive Poster and Demonstration Sessions*, ACLdemo '05, pages 81–84. Association for Computational Linguistics.

Natalia Viani, Joyce Kam, Lucia Yin, André Bittar, Rina Dutta, Rashmi Patel, Robert Stewart, , and Sumithra Velupillai. 2020. Temporal information extraction from mental health records to identify duration of untreated psychosis. *Journal of Biomedical Semantics*, 11(1):1–11.

Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. 2016. Order matters: Sequence to sequence for sets. In *International Conference on Learning Representations 2016*.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2692–2700. Curran Associates, Inc.

Roland Vollgraf. 2019. Learning Set-equivariant Functions with SWARM Mappings. *ArXiv e-print*, 1906.09400.

Eric Wallace, Jens Tuyls, Junlin Wang, Sanjay Subramanian, Matt Gardner, and Sameer Singh. 2019. AllenNLP Interpret: A framework for explaining predictions of NLP models. In *Empirical Methods in Natural Language Processing*.

Erica Westly. 2013. Global health: One million deaths. *Nature*, 504(7478):22–23.

Jason Weston, Sumit Chopra, and Antoine Bordes. 2015. Memory Networks. In *3rd International Conference on Learning Representations (ICLR 2015)*, pages 1–15.

Sarah Wiegreffe and Yuval Pinter. 2020. Attention is not not explanation. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, pages 11–20.

World Health Organization. 2008. *International statistical classifications of diseases and related health problems. 10th rev*, volume 1. World Health Organization, Geneva, Switzerland.

World Health Organization. 2012. *The 2012 WHO Verbal Autopsy Instrument*. World Health Organization, Geneva, Switzerland.

Yonghui Wu, Min Jiang, Jun Xu, Degui Zhi, and Hua Xu. 2017. Clinical Named Entity Recognition Using Deep Learning Models. *AMIA Annual Symposium proceedings. AMIA Symposium*, 2017:1812–1819.

Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. 2008. Listwise approach to learning to rank. (June):1192–1199.

Hua Xu, Shane P. Stenner, Son Doan, Kevin B. Johnson, Lemuel R. Waitman, and Joshua C. Denny. 2010. MedEx: A medication information extraction system for clinical narratives. *Journal of the American Medical Informatics Association*, 17(1):19–24.

Zhaodong Yan, Serena Jeblee, and Graeme Hirst. 2019. Can character embeddings improve cause-of-death classification for verbal autopsy narratives? In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 234–239, Florence, Italy. Association for Computational Linguistics.

Jie Yang and Yue Zhang. 2018. NCRF++: An open-source neural sequence labeling toolkit. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 74–79.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. *Advances in Neural Information Processing Systems*, 32.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbhakhsh, Barnabás Póczos, Ruslan Salakhutdinov, and Alexander J. Smola. 2017. Deep sets. *Advances in Neural Information Processing Systems*, 2017-Decem(ii):3392–3402.

Mengnan Zhao, Aaron J Masino, and Christopher C Yang. 2018. A framework for developing and evaluating word embeddings of drug-named entity. In *Proceedings of the BioNLP 2018 workshop*, pages 156–160.

Li Zhou and George Hripcsak. 2007. Temporal reasoning with medical data – a review with emphasis on medical natural language processing. *Journal of Biomedical Informatics*, 40(2007):183–202.

Henghui Zhu, Ioannis Ch. Paschalidis, and Amir Tahmasebi. 2018. Clinical Concept Extraction with Contextual Word Embedding. *arXiv*, 1810.10566.