Extracting information from clinical text with limited annotated data

Kevin Lybarger

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2020

Reading Committee:

Mari Ostendorf, Chair

Meliha Yetisgen, Chair

Eve Riskin

Program Authorized to Offer Degree:
Electrical & Computer Engineering

University of Washington

Abstract

Extracting information from clinical text with limited annotated data

Kevin Lybarger

Co-Chairs of the Supervisory Committee:
System Design Methodologies Professor Mari Ostendorf
Electrical & Computer Engineering

Associate Professor Meliha Yetisgen
Biomedical and Health Informatics

Electronic health record (EHR) data informs decision-making in clinical care; however, EHR data are generally underused for other purposes, including secondary use applications. The need to leverage EHR data, including clinical notes, is highlighted by the COVID-19 pandemic, as clinicians, researchers, and policymakers struggle to understand, treat, and contain a new disease. Secondary use cases for EHR data extend to many research areas related to healthcare effectiveness, epidemiology, and public health.

Clinical notes contain many types of patient information that are not well characterized through structured data in the EHR, including social determinants of health (SDOH), symptoms, and other factors relevant to clinical informatics research. These patient data are frequently represented in the clinical narrative, rather than structured data, because structured data entry tools can be time-consuming and free-text entry allows richer descriptions. This text-encoded information can benefit secondary use applications, like large retrospective studies and clinical decision-support systems; however, the key information must first be automatically extracted, creating structured representations from unstructured clinical text. Data driven information extraction models require annotated data for training and evaluation, and annotated clinical data is limited by the high cost of annotation and privacy

regulations.

This work explores the automatic extraction of SDOH and COVID-19 diagnosis, testing, and symptom information from clinical text. The exploration of SDOH and COVID-19 focus on addressing the challenges associated with the limited availability of annotated clinical text. Here, "limited" is intended to mean a relatively small data set or low resource setting. The primary contributions of this work include the introduction of neural clinical information extraction models, new annotated clinical corpora, a novel active learning framework, and a secondary use application utilizing automatically extracted data.

We present state-of-the-art neural information extraction approaches for SDOH and COVID-19 information, specifically designing the data-driven extraction architectures to achieve high performance with limited training data, by using multi-task learning and unsupervised pre-training. The extraction models generate event-based predictions that provide a detailed characterization of SDOH and COVID-19, achieving performance levels comparable to the inter-annotator agreement for several important factors. These information extraction approaches are relevant to a range of clinical data.

As part of the exploration of SDOH and COVID-19, two new annotated corpora are developed: the Social History Annotation Corpus (SHAC) and the COVID-19 Annotated Clinical Text (CACT) Corpus. These corpora include detailed, high-quality annotations that characterize SDOH and COVID-19 across multiple dimensions. SHAC is unique in its annotation detail, size, and heterogeneity, and CACT is one of the first corpora with COVID-19 related annotations. These corpora are a substantial contribution to the available resources for training and evaluating machine learning-based extraction models at the University of Washington and for the larger clinical informatics community.

In collecting SHAC, we introduced a novel active learning framework that uses a relatively simple text classification task as a proxy for a more complex event extraction task. The framework increased corpus richness and heterogeneity and improved extraction performance,

relative to random selection. The largest performance improvements are associated with prominent risk factors, like drug and tobacco use, homelessness, and living with others.

To demonstrate the utility of the automatically extracted data, this work presents a secondary use application exploring the prediction of COVID-19 infection. Incorporating automatically extracted symptom data improves COVID-19 infection prediction performance, beyond just using existing structured data.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY OF TERMS, ABBREVIATIONS, AND ACRONYMS

ALSC: Active Learning using Surrogate Classifiers

CACT: COVID-19 Annotated Clinical Text Corpus

CDS: clinical decision support

EHR: electronic health record

FFNN: feed-forward neural network

POS: part-of-speech

SHAC: Social History Annotation Corpus

ACKNOWLEDGMENTS

I would first like to thank my advisors, Mari Ostendorf and Meliha Yetisgen, who invested an incredible amount of time and effort into my education. I benefited greatly from their experience and mentorship and am very thankful for the opportunity to work with Mari and Meliha.

I also want to thank my committee members Eve Riskin, Thomas Payne, and Shan Liu. Eve was a steadying force throughout my PhD journey, providing emotional, professional, and academic support. Thomas was a catalyst for my involvement in clinical informatics, and I am appreciative for guidance and mentorship.

I am grateful to all my current and former lab mates, including Hao Cheng, Kevin Everson, Hao Fang, Ji He, Aaron Jaech, Michael Lee, Yi Luan, Roy Lu, Farah Nadeem, Sara Ng, Trang Tran, Ellen Wu, Vicky Zayats, and Sitong Zhou. Their advice, mentorship, and comradery were an invaluable part of my PhD journey.

I want to thank Brenda Larson for her guidance and the passion she shares with our entire community.

I am incredibly thankful for all the support my wife, Erica, provided over the course of my PhD. Her support was unwavering, even through late nights and weekends of research. My boys, Hudson and Kian, helped me keep things in perspective, and I appreciate all their patience and support. I also want to thank my parents for emphasizing education and curiosity when I was growing up, as my PhD is the result of my love of learning.

DEDICATION

To my boys, Hudson and Kian,

The noted astronomer, Carl Sagan, said, "Somewhere, something incredible is waiting to be known." Go find *your* incredible something, no matter where it may be or how hard you need to look.

Chapter 1

INTRODUCTION

1.1 Problem

In the clinical domain, electronic health record (EHR) data are underutilized in research and can be mined to guide diagnosis and treatment through secondary uses, including retrospective studies and clinical decision support (CDS) systems [1]. Through retrospective studies, existing EHR data can be used to identify disease co-occurrences and assess treatment outcomes. Historical and real-time patient EHR data can also be utilized by CDS systems to assist clinicians in determining diagnoses and treatments [2]. EHR data includes administrative data (e.g. patient demographics, diagnosis codes), clinical data (e.g. service request, prescriptions, imaging results), and clinical text [1]. Clinical notes are a fundamental component of documentation and decision processes and often capture information not represented in structured data, like social history, medical history, and physical examination data [1, 3]. To use the information contained in clinical text for secondary use, natural language processing (NLP) information extraction (IE) techniques must be used to extract salient information from the text, converting the unstructured text to structured data [3].

Creating machine learning-based IE models requires high-quality annotated data that are representative of the target data. Unfortunately, clinical IE is challenged by data heterogeneity and the limited availability of annotated data. Clinical notes are extremely heterogeneous, as the content, structure, lexicon, and shorthand vary by domain and author [2]. Additionally, the structure and formatting of the notes varies by institution, and notes include grammatical and spelling errors [2]. The same document may contain both text and structured data. Annotated clinical corpora are limited by the high cost of annotation and by privacy regulations defined in the Health Insurance Portability & Accountability Act

(HIPAA) [4], which governs the use of protected health information (PHI). For these reasons, corpus creation and learning in a low resource setting are important elements of this thesis and clinical IE more broadly.

There is a wide range of text-encoded information within the clinical narrative, and this work focuses on social determinants of health (SDOH) and Coronavirus disease 2019 (COVID-19) symptoms, diagnoses and testing. SDOH are the conditions in which people work and live that impact health outcomes [5–7]. Prominent SDOH, like substance use, living situation, and employment, impact morbidity and mortality [8–12]. Understanding SDOH, including behaviors influenced by these social factors, can inform clinical decision-making [12]. COVID-19 is a global pandemic, with 20.2 million confirmed infections and 737 thousand related deaths, as of August 12, 2020 [13]. Tracking the spread of COVID-19 and estimating the true number of COVID-19 infections remains a challenge, even as the availability of COVID-19 testing increases. Symptom information would provide useful indicators for tracking potential COVID-19 infections and disease clusters [14]. For example, Elmore et al. [15] identified elevated rates of patient respiratory complaints starting in December 2019, suggesting COVID-19 spread prior to the establishment of testing capabilities. Certain initial symptoms may be associated with higher risk of complications, and correlations between symptoms and COVID-19 outcomes are not well understood. Additionally, COVID-19 outcomes (infection, hospitalization, need for intensive care unit, etc.) are impacted by SDOH, including smoking, obesity exercise, diet, and homelessness [16–18]. Automatically extracting the text-encoded SDOH and COVID-19 information in the clinical narrative may contribute to improved healthcare and public health.

1.2 Contributions

This work explores clinical IE using an existing clinical data set with SDOH annotations, referred to as *YVnotes* [19], and two new annotated clinical corpora: Social History Annotation Corpus (SHAC) and COVID-19 Annotated Clinical Text (CACT) Corpus. Novel IE architectures are introduced for *YVnotes*, *SHAC*, and *CACT*, and the extractors created

using these corpora are used to explore secondary uses for the extracted information. The primary contributions of this work include:

1. *Neural architectures for clinical IE with limited training data*: This work presents multiple neural IE architectures designed to extract clinical information with limited training data. The architectures are customized to the specific annotation schemes but can be easily generalized to other clinical IE data sets or tasks. All of the architectures utilize multi-task learning, where early model layers share parameters across prediction tasks. The shared parameters leverage similarities and learn dependencies between labeled phenomena. The architectures include:
   (a) *Multi-task Substance Extractor*: The Multi-task Substance Extractor is a neural, multi-task IE model designed to extract a subset of SDOH, specifically alcohol, drug, and tobacco use information [20]. The Multi-task Substance Extractor characterizes substance use across multiple dimensions (e.g. status, extent, temporality) and achieved state-of-the-art performance on YVnotes.
   (b) *Multi-task Event Extractor*: The Multi-task Event Extractor is a generalization of the Multi-task Substance Extractor that utilizes end-to-end training for jointly extracting a range of SDOH information [21]. The Multi-task Event Extractor was trained and evaluated on SHAC. For several critical SDOH, the Multi-task Event Extractor achieved near-human performance (i.e. performance comparable to inter-annotator agreement).
   (c) *Span-based Event Extractor*: The Span-based Event Extractor jointly extracts all event information and is a more flexible architecture that handles co-occurring events and overlapping spans [22]. The Span-based Event Extractor is used to extract the COVID-19 related phenomenon of CACT. The Span-based Event Extractor achieved near-human performance in the extraction of key symptom information.
2. *Annotated corpora*: This work presents two new annotated clinical corpora: SHAC and CACT. Both corpora include detailed event-based annotations, characterizing a range

of attributes, including diagnoses, testing, severity, status, and temporal information. *SHAC* is comprised of 4,480 social history sections with detailed annotations for 12 critical SDOH [21]. SHAC utilizes clinical notes from MIMIC-III [23] and the University of Washington (UW) and Harborview Medical Centers and includes more than 18K distinct events. SHAC is unique in its size and detailed characterization of SDOH. *CACT* consists of 1,472 clinical notes from the UW and includes 30K distinct events characterizing COVID-19 diagnoses, testing, and symptoms [22]. We are unaware of any other clinical corpora with COVID-19 annotations.

3. *Active learning using Surrogate Classifiers*: Active learning identifies samples (e.g. sentences or documents) for annotation that maximize model learning. Active learning query functions typically incorporate classifier predictions to identify samples near the decision boundary or improve coverage across the label space. Active learning is well-established within text classification and sequence tagging tasks; however, it is less explored in relation and event extraction tasks, which involve more complex predictions where spans are labeled and linked. This work introduces a novel active learning framework, referred to as Active Learning using Surrogate Classifiers (ALSC), which uses a combination of text classification tasks as a proxy for a more complex event extraction task [21]. A portion of the SHAC training set was selected using *ALSC*, which increased the diversity and richness of the annotations and improved IE extraction performance, relative to random selection. Active selection improved IE performance the most for less frequent, high-risk SDOH, including drug and tobacco use, homelessness, and living with others.
4. *Secondary use with automatic labels*: The relationship between the COVID-19 infection and the automatically extracted symptom information was explored through a secondary use application. In a set of notes paired with EHR data, the likelihood of COVID-19 infection was predicted using both structured data and automatically extracted symptom data. Incorporating the automatically extracted symptom data improves COVID-19 prediction performance, and two of the top three most predictive

features are automatically extracted cough and fever information.

1.3 Overview

Chapter 2, Background, provides a general overview of recent work related to neural language processing, IE, annotated clinical corpora, and active learning. Task-specific literature is discussed in the applicable chapters.

Chapter 3, Architectures for IE, describes the three neural architectures presented in this work: Multi-task Substance Extractor, Multi-task Event Extractor, and Span-based Event Extractor.

Chapter 4, Substance Use, explores the extraction of substance (alcohol, drug, and tobacco) use information from clinical text, using YVnotes. The performance of the Multi-task Substance Extractor, when trained and evaluated using YVnotes, is presented. The extractor is applied to a large corpus of publicly available clinical notes to explore the prevalence of substance use.

Chapter 5, Social Determinants of Health, broadens the exploration of SDOH, introduces the newly annotated SHAC, and presents the ALSC active learning framework. The distribution of annotated risk factors is explored, comparing random sampling to ALSC. Experimentation demonstrates ALSC improves extraction performance, beyond random sampling. Using the Multi-task Event Extractor, the initial extraction performance for SHAC is presented.

Chapter 6, COVID-19, presents COVID-19 related work, including the new CACT corpus. The CACT annotation scheme is described and a summary of the annotated data is presented. The Span-based Event Extractor is trained and evaluated on CACT, providing the first extraction results for the corpus.

Chapter 7, Secondary Use with Automatic Labels, presents an initial secondary use application for the automatically extracted COVID-19 symptom data.

Chapter 8, Conclusions, summarizes the main contributions of this work and presents promising areas for future work.

Chapter 2
BACKGROUND

This section presents relevant background for this work. Section 2.1 describes common IE tasks, provides a high-level discussion of methodologies applied to these tasks, and motivates the use of neural IE techniques. Section 2.2 describes contemporary approaches for language representation, multi-task learning, and neural modeling techniques relevant to IE and NLP more broadly. Section 2.3 describes annotated clinical corpora related to SDOH and symptoms. Section 2.4 describes active learning and relevant work.

2.1 Information Extraction

The goal of IE is to create structured representations from unstructured text. There are several frequently explored IE tasks, including entity recognition, relation extraction, coreference resolution, and event extraction. *Entity recognition* involves identifying and classifying noun phrases in text, based on a pre-defined set of categories (e.g. identifying and classifying person names, organizations, and locations). *Relation extraction* involves detecting and classifying entities (similar to entity recognition) and identifying the semantic relationship between the identified entities (e.g. determining a city is located in a specific country, or determining the relationship between two people). *Coreference resolution* is the task of identifying all mentions of the same entity in a text (e.g. determining the proper noun referred to by pronouns). *Event extraction* involves identifying the phrase that indicates an event is present (called the "trigger"), classifying the trigger span, identifying argument (attribute) spans that characterize the event, and classifying the roles (relations) of the arguments (e.g. identifying the phrase "outbreak" in a news feed as an indicator of a type of event) . Relation extraction, coreference resolution, and event extraction tasks all involve identifying

spans of interest and predicting links between identified spans, and there are similarities in the extraction architectures applied to these tasks.

These information extraction tasks are relevant to a range of clinical IE problems. As examples, identifying protected health information (de-identification of medical records) can be approached as an entity recognition task [24]. Identifying medical problems, tests, and treatments and determining the relationship between these identified entities can be framed as a relation extraction task [25]. Identifying prescription drugs and associated adverse outcomes can also be explored as a relation extraction task [26, 27]. Characterizing multiple aspects of alcohol, drug, and tobacco use, like status, type, extent, and temporal information, can be framed as an event extraction task [19]. This work approaches the extraction of SDOH and COVID-19 information as entity recognition and event extraction tasks.

There is a long history of IE in general and clinical domains, and the techniques have evolved over time, starting with rule-based systems, then transitioning to data-driven discrete modeling approaches, and currently utilizing neural networks. The approaches used in clinical IE tend to lag the methodologies used in the general domain. In a literature survey, Wang et al. [28] found that more than 60% of clinical IE studies from 2009-2016 used only rule-based systems. In contrast, Chiticariu et al. [29] found that rule-based systems represented less than 4% of recent general domain IE works from 2003-2012. In a survey of general domain conference papers, Young et al. [30] found that approximately 30%-40% of papers in 2012 used neural networks and that this proportion grew to approximately 70% by 2017. Neural approaches are becoming increasingly prominent in the clinical domain, as well [31]. The subsequent section describes prominent neural modeling approaches, which are relevant to this work.

2.2 Neural Networks for Language Processing

This section presents relevant neural modeling approaches in IE and NLP more broadly, focusing on general methods and specific systems that are built on in this work.

2.2.1 Learned Word Representations

Pre-trained Word Embeddings: Within the context of neural modeling, it is common to learn a vector representation (embedding) of a word or word sequence (e.g. phrase, sentence, document, etc.). There are many ways to create word embeddings, including the popular word2vec (Skip-gram and CBOW variants) [32] and GloVe [33] approaches. Word embeddings provide a rich word representation, which can include syntactic and semantic information. These approaches can be used to pre-train word embeddings on large corpora of unlabeled text (i.e. millions or billions of words). Word embeddings are typically created by applying unsupervised learning techniques to unlabeled text.

Pre-trained Language Models: Many recent NLP systems, including IE, use pre-trained language models, like Embeddings from Language Models (ELMo) [34], Bidirectional Encoder Representations from Transformers (BERT) [35], and XLNet [36], that leverage large corpora of annotated text with billions of words [37–40]. Pre-trained word embeddings, like word2vec or Glove embeddings, reflect a broad notion of context, based on general context statistics across a training corpus. In contrast, pre-trained language models generate contextualized word embeddings that reflect the specific context of each word within the sentence or document. There are many domain-specific BERT variants, including Alsentzer et al. [41]'s *Bio+Clinical BERT*, which is trained on abstracts and papers from PubMed and clinical notes from MIMIC-III [23], and *Bio+Discharge Summary*, which PubMed and discharge summaries from MIMIC-III.

2.2.2 Multi-task Learning

In multi-task modeling, a single model generates multiple outputs and/or leverages shared parameters across multiple data sets or tasks. Multi-task learning can be useful in low-resource tasks (limited data) for two reasons: parameter sharing between tasks can reduce training requirements and learning with multiple objectives can lead to more robust models. Collobert and Weston [42] use a single model to predict multiple phenomena, including part

of speech (POS), chunks, named entities, semantic roles, etc. Neural multi-task models have achieved state-of-the-art performance in a variety of IE tasks [42–47]. Liu et al. [44], Luan et al. [45], and Peters et al. [47] use a multi-task approach where input layers are shared by multiple models that operate on different data sets. In the clinical domain, Maldonado et al. [48] uses a multi-task model to simultaneously extract multiple electroencephalography concepts and attributes, Harutyunyan et al. [49] used clinical time series data and neural multi-task modeling to predict in-hospital mortality, length of stay, phenotyping, and decompensation, and Jaques et al. [50] predicted health, stress, and happiness using a neural multi-task model and data from wearable sensors and smartphone logs. Both Harutyunyan et al. [49] and Jaques et al. [50] explored these prediction tasks retrospectively, and prospective performance has been not evaluated.

2.2.3 Text Classification

In text classification, labels are assigned to variable-length word sequences (e.g. sentences, documents). Examples of text classification tasks include predicting the sentiment of customer reviews ("positive," "neutral," or "negative") or identifying social media posts with toxic (hate) speech. Text classification models generally include a mechanism for converting variable length sequence representations to a fixed-length vector, which then feeds into a feedforward neural network (FFNN) or other classifier layer. Popular text classification approaches include convolutional neural networks (CNN) [51], recurrent neural networks (RNN) [52], and self-attention [53]. Recent state-of-the-art text classification models utilize large transformer models, like BERT [35].

2.2.4 Sequence Tagging

In sequence tagging, labels are predicted for each token in a word sequence. Examples of sequence tagging tasks include predicting POS tags and identifying named entities. Several IE works achieved high performance using the Conditional Random Field (CRF) model [54–56]. More recent IE work utilizes RNNs, including the Long Short Term Memory (LSTM)

network and bidirectional-LSTM (bi-LSTM), which capture long-range word dependencies [46, 46]. A popular LSTM-based approach to sequence tagging incorporates a CRF layer at the output of the LSTM or bi-LSTM [57–59]. The inclusion of the CRF allows the model to learn allowable transitions between labels and conditionally independent predictions.

2.2.5 General Domain IE

Many contemporary coreference resolution, relation extraction, and event extraction works use end-to-end multi-layer neural models that encode an input word sequence using a recurrent layer, classify spans (entities, arguments, etc.), and predict the relationship between spans (coreference, relation, role, etc.) [60–62]. In this context, the phrase "end-to-end" means the input to the model is (tokenized) text and the output is the extracted data, without intermediate algorithms/steps. Typically, "end-to-end" implies the model is jointly trained and error is back propagated from the output predictions to the input layer. Joint end-to-end training is achieved through a multi-task learning, where the loss associated with span classification and span relationship prediction is aggregated during training. The Multi-task Event Extractor and Span-based Event Extractor introduced in this work utilize joint end-to-end training.

In a relation extraction task, Zheng et al. [60] predict entities using an augmented LSTM layer that generates sequentially dependent predictions, similar to a CRF, and resolves relations using CNN. In an event extraction task, Orr et al. [61] uses attention to combine outputs from separate gated recurrent unit (GRU) layers that encode temporal (word sequence) and syntactic (dependency) information. Pang et al. [62] jointly extract entities and relations by encoding sentences using both bi-LSTM and Transformers and decode triplet predictions using a dual-pointer algorithm. In a relation extraction task, Huang et al. [37] fine-tuned BERT with a CRF entity extraction layer and multi-head attention relation classification layer. Also in a relation extraction task, Wang et al. [38] fined-tuned BERT, predicting entities and relations using a linear layers operating on average pooled BERT output states. The extraction models introduced in this thesis use many of the same modeling layers, including

BERT, bi-LSTM, self-attention, CNN, and CRF.

Of most relevance to the final event extractor in this thesis work, the Span-based Event Extractor, is a series of developments starting with Lee et al. [63], which introduces a span-based coreference resolution model that enumerates all spans in a word sequence, predicts entities using a FFNN operating on span representations, and resolves coreferences using a FFNN operating on entity span-pairs. Luan et al. [64] adapted this framework to entity and relation extraction, with a specific focus on scientific literature. Luan et al. [39] extended the method to take advantage of co-reference and relation links in a graph-based approach for jointly predicting entity spans, co-references, and relations. By updating span representations in multi-sentence co-reference chains, the graph-based approach achieved state-of-the-art on several IE tasks representing a range of different genres. Wadden et al. [40] expands on Luan et al. [39]'s approach, adapting it to event extraction tasks. Our Span-based Event Extractor builds on Luan et al. [64] and Wadden et al. [40]'s work, augmenting the modeling framework to fit the CACT annotation scheme, which includes argument subtypes. In CACT, event arguments are generally close to the associated trigger, and inter-sentence events linked by co-reference are infrequent, so the graph-based extension, which adds complexity, is unlikely to benefit our extraction task.

2.2.6 Clinical IE

Recent clinical IE work has benefited from neural modeling approaches, including CNN [65, 66], autoencoders [67], RNN [68–70], attention networks [71], multi-task learning [48–50], and other neural frameworks. Dernoncourt et al. [68] explores a de-identification task using an LSTM with an output classification layer that models label transition probabilities. In a clinical entity extraction task, Shi et al. [70] uses a stacked LSTM-CRF approach. In a medical concept slot for task, Shi et al. [70] encode input sentences using a bi-LSTM, predict entity and attribute spans using a CRF, and resolve relations using a FFNN operating on the averaged hidden states of the entity-attribute pairs. Chen et al. [26] extract adverse drug reaction information, identifying entities (drugs, dosage, etc.) using a knowledge-based

system (i.e. Unified Medical Language System) and resolving relations using a multi-layer bi-LSTM-attention model. Also exploring an adverse drug event extraction task, Christopoulou et al. [27] identify drugs and reactions using bi-LSTM and attention layers and predict drug interactions (dependencies) using a binary classifier operating on the drug/reaction representations. Similar to several of these works, we employed the stacked bi-LSTM-CRF approach, also including self-attention layers at the bi-LSTM output.

2.3 Corpora

Creating machine learning IE models requires annotated corpora for model training and evaluation. To achieve high extraction performance, the annotated corpora must be sufficiently large, diverse, and representative of the target data. Corpora size is especially important when using neural IE approaches, which tend to require more annotated data than simpler discrete models. Existing corpora associated with the two tasks explored in this thesis are described below.

SDOH: Multiple corpora with note-level SDOH annotations have been developed. For example, the i2b2 NLP Smoking Challenge introduced a publicly available corpus where tobacco use status is labeled at the note-level [72]. Gehrmann et al. [73] annotated MIMIC-III discharge summaries with note-level phenotype labels, including substance abuse and obesity. Feller et al. [74] annotated 38 different SDOH at the note-level. While the note-level labels are informative, the annotation scheme is insufficient to fully characterize the SDOH. For example, the note-level tobacco status labels in the i2b2 NLP Smoking Challenge provide useful information but are insufficient to characterize the severity of patients' smoking history and habit. Additionally, the note-level labels cannot distinguish between descriptions of past and current tobacco usage within a given note.

Melton et al. [75] reviewed three widely used public health surveys to understand the questions used by practitioners to measure behaviors that may be relevant to clinical care. From this investigation, Melton et al. [75] proposed an information model for survey items related to alcohol, drug, and tobacco use, that included dimensions of temporality, degree

of exposure, and frequency. Melton et al. [75]'s model reflects the practitioners' needs and insights into how these substances impact patient health. Chen et al. [76] surveyed the use of free-text describing alcohol use, coding for "type," "status," "temporal," and "amount." Similar dimensions or characteristics have been proposed and implemented by others, including Carter et al. [77] and Wang et al. [78]. Yetisgen and Vanderwende [19] annotated a corpus of 364 social history sections with SDOH, including substance use, using an annotation scheme similar to the models/schemas of Melton et al. [75], Chen et al. [76], Carter et al. [77], Wang et al. [78]. Wang et al. [79] introduced a corpus with detailed substance use annotations for 691 clinical notes. The annotated corpus introduced in this manuscript, SHAC, follows a similar annotation scheme as these works [19, 75–78].

Unfortunately, existing publicly available corpora with SDOH annotations are lacking in either annotation detail, size, and/or heterogeneity. To fill this gap, we introduce SHAC, which is a relatively large corpus with high quality, detailed SDOH annotations. SHAC is heterogeneous in that it includes clinical notes from multiple institutions and note types, and in the use of active selection to encourage a richer representation of SDOH events.

COVID-19 and symptoms: Given the recent onset of COVID-19, there are limited COVID-19 corpora for NLP experimentation. Corpora of scientific papers related to COVID-19 are available [80, 81], and automatic labels for biomedical entity types are available for some of these research papers [82]. However, we are unaware of corpora of clinical text with supervised COVID-19 annotations.

Multiple clinical corpora are annotated for symptoms. As examples, South et al. [83] annotated symptoms and other medical concepts with negation (present/not present), temporality, and other attributes. Koeling et al. [84] annotated a pre-defined set of symptoms related to ovarian cancer. For the i2b2/VA challenge, Uzuner et al. [25] annotated annotated medical concepts, including symptoms, with assertion values and relations. While some of these corpora may include the annotation of symptoms relevant to COVID-19 (e.g. "cough" or "fever"), the distribution and characterization of symptoms in these corpora may not be consistent with the symptoms in COVID-19 related notes. To fill the gap in clinical COVID-

19 annotations, including symptoms, we introduce CACT to provide a relatively large corpus with COVID-19 diagnosis, testing, and symptoms annotations.

2.4 Active Learning

Annotated corpora are generally created by having human-annotators label phenomena of interest in unannotated (unlabeled) text. The annotators assign labels to documents, sentences, or phrases, following a set of predefined annotation guidelines. The available unlabeled text is often significantly larger than the annotation budget. Randomly selecting samples (e.g. documents or sentences) for annotation is suboptimal from a model learning perspective, as samples vary in their usefulness, particularly when the phenomena of interest are infrequent. *Active learning* is an approach for selecting samples for annotation that maximizes model learning [85, 86]. In text annotation projects, the active learning query function typically scores sample informativeness, representativeness, and/or diversity [87–89]. Informativeness describes the potential for a sample to reduce classification uncertainty (i.e. proximity to the decision boundary). The literature varies in the usage of the terms "representativeness" and "diversity." Here, "representativeness" describes the degree to which a sample describes the structure of the data, and "diversity" characterizes the variation in the samples selected.

Active learning is well-established for classification tasks, where a single label is predicted for each sample. Multiple studies have applied active learning to text classification tasks, where a sample is a sentence or a document. Sample informativeness is derived from classification uncertainty scores, such as maximizing entropy [90] or minimizing a support vector machine margin [91, 92]. Du et al. [89] assesses diversity based on classifier posterior distributions, and Wu and Ostendorf [90] assesses diversity and representativeness based on sample similarity within the observation space.

Approaches for applying active learning to sequence tagging problems are also well-established [93–98]. Although predictions are made at the token-level, sample selection is typically performed at the sentence or document-level. Representativeness and/or diver-

sity are often assessed by calculating sentence similarity metrics in the observation space [93–95, 97]. Sequence-level uncertainty scores are calculated by various measures, like normalized prediction sequence likelihood and minimum token-level confidence. In the clinical and biomedical domain, uncertainty scores are generated with conditional random field (CRF) models [93–97] or a neural tagger based on contextualized embeddings from ELMo and BERT [98].

Active learning is less explored in relation and event extraction tasks, where triggers (heads), arguments, and/or relations are annotated. The predictions are more complex, involving labeling and linking spans of text. Maldonado et al. [48] apply active learning to a clinical relation extraction task, selecting samples using the average entropy of all predicted phenomena as an uncertainty score. More recently, Maldonado and Harabagiu [99] explore active learning in a medical concept and relation extraction task. In lieu of a heuristic query function, an optimal selection strategy is learned from data with strong and weakly supervised labels, including 1,000 electroencephalogram (EEG) reports with automatic annotations generated by existing extraction models.

A portion of the SHAC training set was actively selected, to improve extraction performance and data heterogeneity. SHAC is annotated using an event-based structure, where SDOH are characterized through multiple argument types. These argument types are not equally important for secondary use applications, and the entropy of different determinant-argument combinations may differ significantly. Without sufficient annotated data to learn an optimal selection strategy, we use a simplified text classification task as a surrogate for assessing sample uncertainty, to prevent under sampling the critical phenomena. We hypothesized that the surrogate task would improve extraction performance in the more complex event extraction task and validated the hypothesis with experiments on SHAC data.

Chapter 3
ARCHITECTURES FOR IE

This chapter describes the general event-based IE task that is explored throughout this work and the IE architectures introduced in this work: Multi-task Substance Extractor, Multi-task Event Extractor, and Span-based Event Extractor. This work describes models developed for clinical IE from 2017-2020 [20–22], and the sequence of architectures introduced represents an evolution both in the framing of the event extraction task and in the neural architectures employed. The experimental assessment has involved multiple tasks (presented in subsequent sections), but the models are presented together to highlight the general applicability and different advances.

3.1 Event Extraction Task

In the clinical domain, event-based annotations characterize the phenomena of interest across multiple dimensions. Events in the clinical narrative capture changes to the status, extent, temporality, and other attributes of risk factors and diagnoses in the patient timeline. Each event includes a trigger and all associated arguments to describe a specific change/incident in this timeline. The annotated corpora used in this work (YVnotes, SHAC, and CACT) all use a similar event-based structure, although the annotated features differ. Figure 3.1 contains SDOH event annotations from SHAC, which includes descriptions of employment and tobacco, alcohol, and drug use, and Figure 3.2 contains event annotations from CACT that describe COVID-19 diagnosis and symptoms. The event structure identifies and labels multi-word spans of interest (arguments) and links related spans, creating a more complex and descriptive representation. Although the identified arguments may not strictly be noun phrases, the argument identification task is similar to the entity identification. Predicting the

links between triggers and arguments (referred to as the argument roles) is similar to predicting relations between entities, in that the argument roles captures the semantic relationship between triggers and arguments.

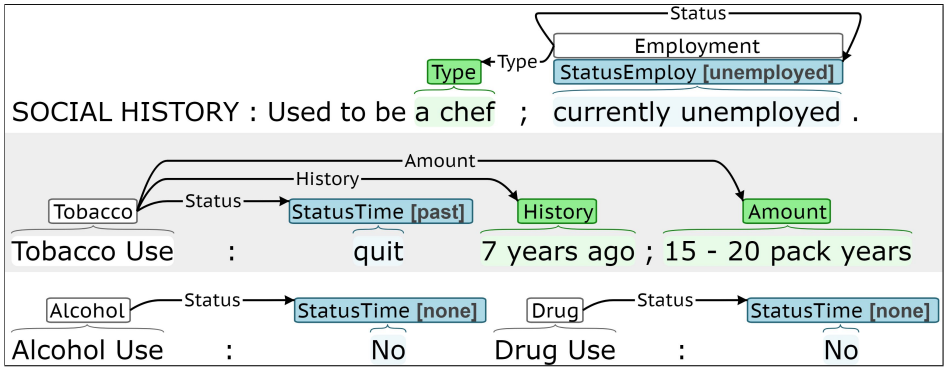

Figure 3.1: SHAC annotation examples for *Employment*, *Tobacco*, *Alcohol*, and *Drug* events

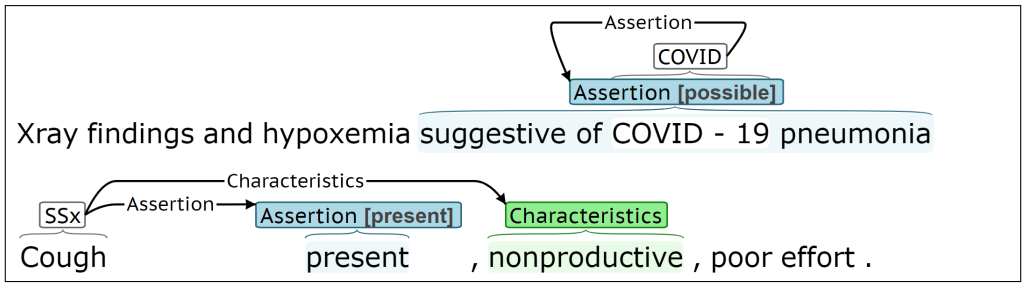

Figure 3.2: CACT annotation examples for *COVID* and *Symptom* (*SSx*) events

In this work, trigger annotation includes the selection of a multi-word span and the identification of the event type (e.g. *Employment*, *Tobacco*, or *COVID*) and is denoted by white labels in Figures 3.1 and 3.2. For example, the first line in Figure 3.1 includes an event where the trigger span is "currently employed" and event type is *Employment*. The arguments identify specific attributes of the event and connect to the trigger through the role (relation). There are two types of arguments: *labeled arguments* and *span-only arguments*. The annotation of labeled arguments includes the argument type, span, and subtype. For

example, the first line of Figure 3.1 includes a labeled argument for which the argument type is *StatusEmploy*, span is "currently unemployed," and subtype is *unemployed*. Similar to labeled arguments, the annotation of span-only arguments includes the argument type and span; however, span-only arguments do not include an additional subtype label. For example, the first line of Figure 3.1 includes a span-only argument for which the argument type is *Type* and the span is "a chef." For labeled arguments, the identified subtype captures the most important argument information, as the subtype essentially represents the normalization of the argument span. For span-only arguments, the identified span cannot easily be mapped to a fixed set of classes, and the identified span captures the most important information.

3.2 Multi-task Substance Extractor

The Multi-task Substance Extractor extracts substance use information across multiple dimensions, describing status, extent, type, and temporality [20]. It is designed based on the substance use annotation schema of Yetisgen and Vanderwende [19]'s *YVnotes* corpus, which was used to train and evaluate the extraction framework. Figure 3.3 is a diagram of the Multi-task Substance Extractor. The extracted event types include *Alcohol*, *Drug*, and *Tobacco*. Given the similarities between all three event types, the same arguments are used to characterize each substance. These event types include a single labeled argument, *Status*. The span-only arguments include *Amount*, *Frequency*, *Exposure History*, *Type*, and *Quit History*. Each event is required to have a trigger and a *Status* argument. The Multi-task Substance Extractor leverages shared information and similarities between event types and arguments.

The Multi-task Substance Extractor generates sentence-level predictions for trigger and labeled arguments (treating these problems as sentence-level text classification tasks) and token-level predictions for span-only arguments (sequence tagging). Similar to previous work with YVnotes, this work assessed the performance of the Multi-task Substance Extractor through these sentence-level and token-level predictions, without explicit consideration of the event structure of the annotations (see Chapter 4 for details). In other words, the

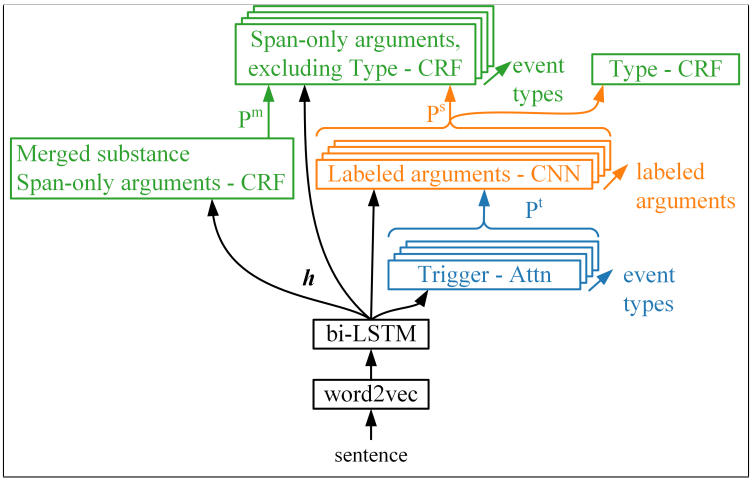

Figure 3.3: Multi-task Substance Extractor

performance assessment does not consider the prediction of argument roles and only evaluates individual sentence-level and token-level predictions.

Shared Layers: The inputs to the model are pre-trained word embeddings, which feed into a bi-LSTM layer with layer normalization [100]. The forward and backward outputs states of the LSTM are concatenated resulting in $n \times 2v_h$ matrix, $\boldsymbol{h} = [\boldsymbol{h}_f, \boldsymbol{h}_b]$, where v_h is the LSTM hidden size and n is the sequence length. $\boldsymbol{h}$ is used as features in downstream output classifiers.

Trigger: Self-attention is used to estimate the probability of each event type occurring in the sentence, automatically identifying word positions that best predict a given substance. For sentences predicted to have a given event type present, the token position with the largest attention weight is the trigger span. Attention weights, $\boldsymbol{a}_c$, are calculated as

$$\boldsymbol{a}_c = softmax(tanh(\boldsymbol{w}_c^a \boldsymbol{h}^T)) \tag{3.1}$$

where c denotes the event type, $\boldsymbol{w}_c^a$ is a $1 \times 2v_h$ learned vector, and $\boldsymbol{a}_c$ is a $1 \times n$ vector. The probability of a substance event is calculated using the weighted average of the hidden

states as

$$\boldsymbol{P}_c^t = softmax(\boldsymbol{w}_c^t(\boldsymbol{a}_c\boldsymbol{h})^T + \boldsymbol{b}_c^t) \tag{3.2}$$

where $\boldsymbol{w}_c^t$ is a $2 \times 2v_h$ weight matrix and $\boldsymbol{b}_c^t$ is a 2×1 bias vector. The probability of the events is concatenated to form a 3-dimensional vector (# substances) for use in *Status* classification. The ground truth for learning $\boldsymbol{P}_c^t$ is determined based on whether or not a given event occurred at least once in the sentence. Error from the *Status* classification is not back-propagated to the trigger network in training.

Labeled Arguments: *Status* is predicted using separate, sentence-level text classifiers for each event type. A CNN with max-pooling creates a low dimensional sentence representation, $\boldsymbol{g}_c$, where c denotes the event type. The use of this CNN approach is motivated by the ability of CNNs to highlight salient information in the input sequence and preserve word order information. The CNNs uses multiple filter widths, and multiple filters are created for each filter width. The trigger probability vector, $\boldsymbol{P}_c^t$, is concatenated with $\boldsymbol{g}_c$ to form a vector of size m-by-1 to predict *Status* as

$$\boldsymbol{P}_c^s = softmax(\boldsymbol{w}_c^s[\boldsymbol{P}_c^t, \boldsymbol{g}_c] + \boldsymbol{b}_c^s) \tag{3.3}$$

where $\boldsymbol{w}_c^s$ is a $|y_c^s| \times q$ weight matrix, $\boldsymbol{b}_c^s$ is a $|y_c^s| \times 1$ bias vector (q = #event types + #filters $\times$ $2v_h$), and $|y_c^s|$ is the number of *Status* subtypes, including a *null* class. *Status* probabilities, $\boldsymbol{P}_c^s$, for each event type are concatenated to form $\boldsymbol{P}^s$, which is used as input features in the sequence tagging tasks. To prevent the sequence tagging tasks from negatively impacting the *Status* classification, error from the sequence tagging tasks is not back-propagated to the *Status* classifiers.

Span-only Arguments: Linear-chain CRF models are used to extract span-only arguments using the begin-inside-outside (BIO) labeling formatting. The CRF is used because of its ability learn and enforce allowable transitions between sequence labels (e.g. I-frequency label may follow a B-frequency label but not I-amount). The *Type* arguments for all substances were extracted using a single CRF, with input features $\boldsymbol{h}$ and $\boldsymbol{P}^s$. The *Amount*,

Frequency, *Exposure History*, and *Quit History* labels were merged across all three event types for training a substance-independent CRF that estimates probabilities, $\boldsymbol{P}^m$, for these arguments with input features $\boldsymbol{h}$ and $\boldsymbol{P}^s$. Then, substance-specific *Amount*, *Frequency*, *Exposure History*, and *Quit History* arguments are extracted using separate, substance-specific CRFs, with input features $\boldsymbol{h}$, $\boldsymbol{P}^s_c$, and $\boldsymbol{P}^m$.

3.3 Multi-task Event Extractor

The Multi-task Event Extractor is a generalization of the Multi-task Substance Extractor and is shown in Figure 3.4. It utilizes contextualized word embeddings generated using BERT, which did not exist during the development of the Multi-task Substance Extractor. The Multi-task Event Extractor generates sentence and token-level predictions that are assembled into events, allowing the performance assessment to consider the full event structure of the annotations (see Chapter 5 for details). The Multi-task Event Extractor was trained and evaluated using the newly annotated SHAC, which involved a larger number of event types including substance use, employment, and living situation. The extraction framework can be expanded to any number of event types or arguments.

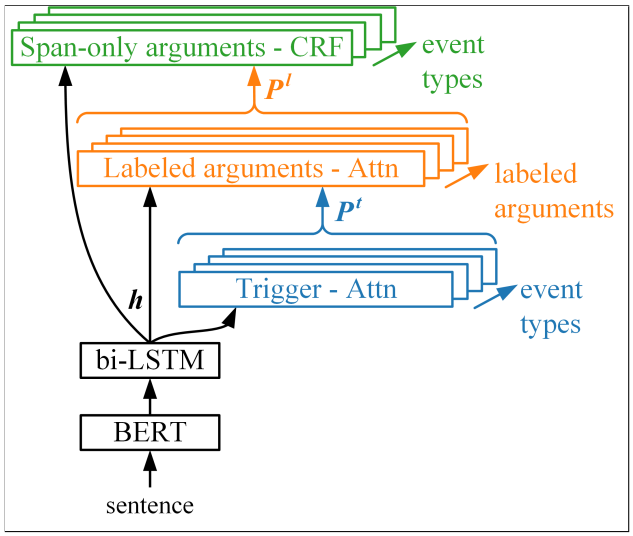

Figure 3.4: Multi-task Event Extractor

Shared layers: Similar to previous multi-task work [50, 66, 101–104], the Multi-task Event Extractor shares information across tasks (event types and arguments in this application). Individual sentences are encoded using *Bio+Discharge Summary BERT*, creating an $n \times v_b$ matrix, where n is the sentence length in tokens and v_b is the BERT embedding size. Similar to other work [105], only the last word piece embedding for each token is used, to simplify the downstream sequence tagging. The BERT encoding feeds into a bi-LSTM. The forward and backward outputs states of the bi-LSTM are concatenated resulting in $n \times 2v_h$ matrix, $\boldsymbol{h}$, where v_h is the hidden size. $\boldsymbol{h}$ feeds into event type and argument-specific output layers. The BERT weights are frozen, due to the limited amount of annotated data, and the bi-LSTM provides a layer to adapt the BERT output to this task.

Trigger: The presence of each event type is predicted using separate self-attentive binary classifiers (not present/present). Positive predictions serve as the trigger for assembling events, and the token position with the maximum attention weight serves as the trigger span. During training, event type c is considered *present*, if the sentence contains one or more events of type c. Normalized attention weights, $\boldsymbol{a}_c$, are calculated as

$$\boldsymbol{a}_c^t = softmax(\boldsymbol{w}_{a,c}^t \boldsymbol{h}^T) \tag{3.4}$$

where c denotes the event type, $\boldsymbol{w}_{a,c}^t$ is a $1 \times 2v_h$ learned vector, and $\boldsymbol{a}_c^t$ is a $1 \times n$ vector. The self-attention mechanism presented in Equation 3.4 differ slightly from the self-attention mechanism presented with the Multi-task Substance Extractor, in that it omits the nonlinearity, *tanh*. The trigger probability for event type, $\boldsymbol{P}_c^t$, is calculated similar to the Multi-task Substance Extractor (see Equation 3.2). The trigger probabilities, $\boldsymbol{P}_c^t$, are concatenated to form a $2 \times m$ matrix, $\boldsymbol{P}^t$, for the labeled argument prediction. An event is detected if it has probability greater than 50%. Because triggers are predicted using sentence-level classifiers, the Multi-task Event Extractor can only represent a single event of a given event type within a sentence.

Labeled arguments: Labeled argument prediction is also treated as a text-classification

task, and utilizes separate self-attentive output layers for each labeled argument, similar to trigger prediction. The token position with the maximum attention weight serves as the argument span. The probability of labeled argument l for event type c is calculated as

$$\boldsymbol{P}_c^l = softmax(\boldsymbol{w}_c^l[\boldsymbol{P}^t, (\boldsymbol{a}_c^l \boldsymbol{h})^T] + \boldsymbol{b}_c^l) \tag{3.5}$$

where $\boldsymbol{w}_c^l$ is a weight matrix, $\boldsymbol{a}_c^l$ is a vector of attention weights, and $\boldsymbol{b}_c^l$ is a bias vector. The dimension of $\boldsymbol{P}_c^l$ depends on the number of possible labels for that event-argument combination. The labeled argument probabilities, $\boldsymbol{P}_c^l$, are concatenated to form $\boldsymbol{P}^l$, for use in span-only argument detection.

Span-only arguments: Span-only arguments are predicted using CRF [54] layers at the output of the bi-LSTM. The bi-LSTM network learns sequential word dependencies, and the CRF learns conditional dependencies between labels. A separate CRF extracts the span-only arguments for each event type, with input features $\boldsymbol{h}$ and $\boldsymbol{P}^l$. Sequence labels are represented using the BIO approach. The Multi-task Substance Extractor uses a separate CRF to extract all substance *Type* spans, because of the overlap between some *Amount* and *Type* spans. To create a more generalized framework, the Multi-task Event Extractor uses a single CRF to extract all the span-only arguments for each event type, including *Type*. Prediction errors associated with overlapping spans in the supervised labels are captured by the scoring rubric.

3.4 Span-based Event Extractor

The Span-based Event Extractor is a span-based, end-to-end, multi-layer event extraction model that jointly predicts all event phenomena, including the trigger span, event type, argument spans, types, and subtypes, and argument roles [22]. Figure 3.5 presents Span-based Event Extractor. Although the Multi-task Event Extractor is well suited for SHAC, the Span-based Event Extractor is a more flexible and powerful extraction architecture. Unlike the Multi-task Event Extractor, which can only represent a single event of a given type per

sentence, the Span-based Event Extractor can generate multiple trigger (event) predictions of the same event type. This functionality is extremely important for corpora where multiple events of the same type frequently co-occur in sentences. Additionally, the Multi-task Event Extractor cannot represent overlapping spans, due to the CRF-based approach. The Span-based Event Extractor overcomes this limitation and can represent overlapping spans. Lastly, the Multi-task Event Extractor does not generate argument role predictions, rather separate output classifiers are used for each event type. The Span-based Event Extractor jointly predicts arguments and argument roles, creating a more flexible architecture.

The CACT annotation scheme differs from typical event extraction tasks, like ACE05 [106], in that labeled arguments require the argument type and subtype to be predicted. Resolving the argument subtypes requires a classifier with additional predictive capacity, and the Span-based Event extractor differs from prior related work in that multiple span classifiers are used to accommodate the argument subtypes.

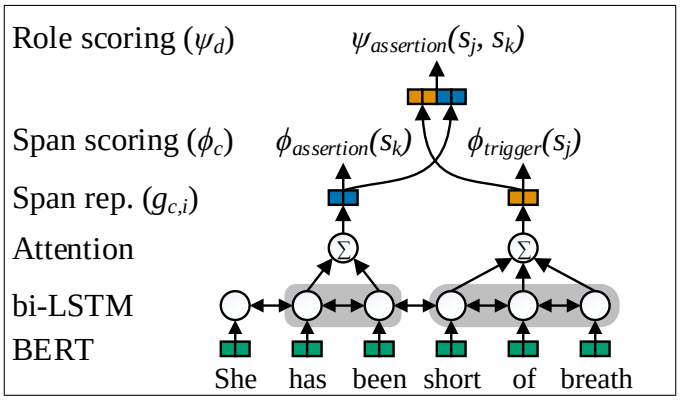

Figure 3.5: Span-based Event Extractor

Each input sentence consists of tokens, $X = \{x_1, x_2, ...x_n\}$, where n is the number of tokens. For each sentence, the set of all possible spans, $S = \{s_1, s_2, ...s_m\}$, is enumerated, where m is the number of spans with token length less than or equal to M tokens. The Span-based Event Extractor generates trigger and argument predictions for each span in S and predicts the pairing between arguments and triggers to create events from individual

span predictions.

Input encoding: Input sentences are mapped to contextualized word embeddings using *Bio+Clinical BERT* [41]. Similar to the Multi-task Event Extractor, the contextualized word embeddings feed into a bi-LSTM without fine tuning BERT (no backpropagation to BERT). The primary motivation for freezing BERT in the Span-based Event Extractor was to limit computational cost. The bi-LSTM has hidden size v_h. The forward and backward states, $\boldsymbol{h}_{t,f}$ and $\boldsymbol{h}_{t,b}$, are concatenated to form the $1 \times 2v_h$ dimensional vector $\boldsymbol{h}_t = [\boldsymbol{h}_{t,f}, \boldsymbol{h}_{t,b}]$, where t is the token position.

Span representation: Each span is represented as the attention weighted sum of the bi-LSTM hidden states. Separate attention mechanisms, c, are implemented for trigger and each labeled argument, and a single attention mechanism is implemented for all span-only arguments, $c \in \{1, 2 \dots p\}$ ($p = 1$ for trigger + #labeled arguments + 1 for span-only arguments). The attention score for span representation c at token position t is calculated as

$$\alpha_{c,t} = \boldsymbol{w}_{\alpha,c} \boldsymbol{h}_t^T \tag{3.6}$$

where $\boldsymbol{w}_{\alpha,c}$ is a learned $1 \times 2v_h$ vector. For span representation c, span i, and token position t, the attention weights are calculated by normalizing the attention scores as

$$a_{c,i,t} = \frac{\exp(\alpha_{c,t})}{\sum_{k=start(s_i)}^{end(s_i)} \exp(\alpha_{c,k})}, \tag{3.7}$$

where $start(s_i)$ and $end(s_i)$ denote the start and end token indices of span s_i. Span representation c for span i is calculated as the attention-weighted sum of the bi-LSTM hidden state as

$$\boldsymbol{g}_{c,i} = \sum_{t=start(s_i)}^{end(s_i)} a_{c,i,t} \boldsymbol{h}_t. \tag{3.8}$$

Span prediction: Similar to the span representations, separate span classifiers, c, are implemented for trigger and each labeled argument, and a single classifier predicts all span-

only arguments, $c \in \{1, 2 \dots p\}$ ($p = 1$ for trigger + #labeled arguments + 1 for span-only arguments). Label scores for classifier c and span i are calculated as

$$\boldsymbol{\phi}_c(s_i) = \boldsymbol{w}_{s,c}\text{FFNN}_{s,c}(\boldsymbol{g}_{c,i}), \tag{3.9}$$

where $\phi_c(s_i)$ yields a vector of label scores of size $|L_c|$, $\text{FFNN}_{s,c}$ is a non-linear projection from size $2v_h$ to v_s, and $\boldsymbol{w}_{s,c}$ has size $|L_c| \times v_s$.

The trigger prediction label set, $L_{trigger}$, is the union of event types and a *null* label. Separate classifiers are used for each labeled argument with label set, $L_c = \{null \cup L_l\}$, where L_l is the argument subtype label set (see Table 6.1). A single classifier predicts all span-only arguments with label set, $L_{span-only}$, which is the union of the span-only arguments and a *null* label.

Argument role prediction: The argument role layer predicts the assignment of arguments to triggers using separate binary classifiers, d, for each labeled argument and one classifier for all span-only arguments, $d \in \{1, 2, \dots q\}$ (d = #labeled arguments + 1 for span-only arguments). Argument role scores for trigger j and argument k using argument role classifier d are calculated as

$$\boldsymbol{\psi}_d(s_j, s_k) = \boldsymbol{w}_{r,d}\text{FFNN}_{r,d}([\boldsymbol{g}_j, \boldsymbol{g}_k]) \tag{3.10}$$

where $\boldsymbol{\psi}_d(s_j, s_k)$ is a vector of binary scores of size 2, $\text{FFNN}_{r,d}$ is a non-linear projection from size $2v_s$ to v_r, and $\boldsymbol{w}_{r,d}$ has size $2 \times v_r$.

Span pruning: To limit time and space complexity of the pairwise argument role predictions, only the top-K spans for each span classifier, c, are considered during argument role prediction. The span score is calculated as the maximum label score in ϕ_c, excluding the *null* label score.

3.5 Summary

This section introduces three neural IE architectures: Multi-task Substance Extractor, Multi-task Event Extractor, and Span-based Event Extractor. This sequence of extractors represents an evolution of neural architectures over the course of this work's execution, which reflects broader NLP developments. The Multi-task Event Extractor is a generalization of the Multi-task Substance Extractor that incorporates contextualized word embeddings, uses self-attention to predict labeled arguments, and uses fewer CRF output layers. Labeled arguments are predicted using self-attention, rather than a CNN, to facilitate the visualization of predictions and reduce model complexity. The Span-based Event Extractor is the most flexible of the architectures introduced. By using span-based argument detection, rather than text-classification and sequence tagging, the Span-based Event Extractor overcomes some of the key limitations of the preceding multi-task approaches, specifically the assumption of one event per event type in a sentence and no overlapping arguments.

The Multi-task Substance Extractor was developed for a scenario with very limited data. Experimentation with the Multi-task Event Extractor included a substantially larger data set; however, experimentation in Chapter 5 demonstrates it is well suited for event extraction tasks where events of the same type infrequently occur in the same sentence. The Span-based Event Extractor is a more complex framework that is advantageous for event extraction tasks with frequent cooccurrence of events within sentences.

Chapter 4

SUBSTANCE USE

4.1 Overview

The negative impact of substance abuse on health is increasingly recognized as a key factor for morbidity and mortality [8, 10, 107]. There is some evidence that 5-10% of cancers can be attributed to hereditary factors, while 90-95% have been found correlated with lifestyle and environmental factors, such as smoking and alcohol consumption [108]. Alcohol and tobacco use, specifically, are leading risk factors for all-cause and cancer-related mortality [109, 110]. The consequences of illicit drug and prescribed opioid abuse are also widespread, causing permanent physical and emotional damage to users. In many cases, users die prematurely from drug overdoses or other drug-associated illnesses. Characterization of a patient's substance abuse history can be used to assess risk of future negative health outcomes related to substance abuse. Clinical notes contain rich information detailing the history of substance abuse from caregivers' perspective, beyond what is available from structured EHR databases and which can be used to quantify severity of abuse.

We extend prior work using machine learning to automatically extract substance abuse information from clinical text. We introduce and evaluate the Multi-task Substance Extractor using a corpus of clinical notes annotated in Yetisgen and Vanderwende [19]. The corpus, which is referred to here as *YVnotes*, was created using a publicly available data source (MTSamples) and annotated for alcohol, drug, and tobacco abuse information documented in social history sections of history and physical notes.

The presented neural multi-task model outperforms the initial extraction performance in Yetisgen and Vanderwende [19] and strong baselines we created using discrete models. To assess the generalizability of the extraction model, we used the multi-task model to annotate

59.7K discharge summaries from the MIMIC-III corpus [23], and hand-scored the substance status predictions of a randomly selected subset of notes. The performance results are encouraging and demonstrate the feasibility and generalizability of our extraction approach.

This chapter presents our initial work focused on extracting substance use and is largely based on Lybarger et al. [20], published in 2018. In Lybarger et al. [20], I contributed to experiment design, data analysis, and writing, performed all software development (programming), and executed all experimentation.

4.2 Related Work

At the time of this work, prior work on characterizing patient substance use involved rule-based or discrete statistical models. In the i2b2 NLP Smoking Challenge [72, 111], common approaches included Support Vector Machines (SVM) and latent Dirichlet allocation (LDA) Cohen [112], Clark et al. [113], Jonnagaddala et al. [114]. Wang et al. [79] extracted more detailed alcohol, drug, and tobacco use information from clinical notes describing status, type, extent, and temporal information using a rule-based system. Gehrmann et al. [73] predicted patient phenotypes, including substance use, using discrete modeling and CNNs. Yetisgen and Vanderwende [19] extracted substance use information through the sentence-level and token-level predictions related to status, type, extent, and temporal labels, using maximum entropy (MaxEnt) and CRF models and rule-based approaches. Yetisgen and Vanderwende [19], in particular, is relevant because it uses the same corpus as our work and provides a baseline for our study.

4.3 Methods

This section describes the data that is the foundation for experimental work and the discrete models implemented as a baseline. The neural extraction model used in this chapter, Multi-task Substance Extractor, is defined in Section 3.2. The training set and test set assignments from Yetisgen and Vanderwende [19] were unavailable, so we reimplemented discrete models similar to Yetisgen and Vanderwende [19] as a baseline. In the reimplementation of the

discrete models, some additional features were explored.

4.3.1 Data

Table 4.1 summarizes the annotation scheme for substance use in YVnotes. The event type defines the substance, whether *Alcohol*, *Drug*, or *Tobacco*. The event type is defined by the trigger, which is required for all events. Given the similarities between the substance use event types, a common set of argument types is defined for all substances. There is one labeled argument, *Status*, which is the only required argument. The span-only arguments include *Amount*, *Exposure History*, *Frequency*, *Method*, *Type*, and *Quit History*. The span-only arguments are optional. The spans associated with substance-related span-only arguments often have a similar format. For example, *Amount* spans often have the format, (*[quantity] [units]*); however, the units generally vary by substance (e.g. "three packs" vs. "one beer"). The phrases used to describe *Frequency*, *Exposure History*, and *Quit History* are similar for all substances. *Method* was not extracted, as there were insufficient *Method* occurrences to evaluate performance.

Table 4.1: Substance use argument types. * indicates the argument is required.

Event type, e	**Argument type,** a	**Argument Subtypes,** y_l	**Span examples**
Alcohol, Drug, or Tobacco	Trigger*	–	"drinks," "tobacco"
	Status*	{none, current, past}	"denies," "smokes"
	Amount	–	"2 packs," "3 drinks"
	Exposure History	–	"for the past 8 years"
	Frequency	–	"daily," "monthly"
	Method	–	"iv" "chews"
	Type	–	"beer," "cocaine"
	Quit History	–	"seven years ago"

Supervised Labels: YVnotes includes 364 social history sections from 516 history and

physical notes from MTSamples website.[1] The annotated dataset is available for download at the UW-BioNLP website.[2] Table 4.2 contains a summary of the argument frequencies by event and argument type. For labeled arguments (e.g. *Status*), the counts reflect the number of spans. For span-only arguments (e.g. *Amount*), the counts indicate the number of tokens in associated spans.

Table 4.2: Annotation statistics for YVnotes

Argument type	**Alcohol**	**Drug**	**Tobacco**
Status	254	154	278
Type	26	112	50
Method	0	10	4
Amount	69	25	78
Frequency	65	6	59
Exposure History	7	10	37
Quit History	6	2	37

Unlabeled Text: The MIMIC-III corpus [23] discharge summaries (59.7K notes) and physician notes (142K notes) were used in unsupervised learning to pre-train word embeddings for the neural multi-task model. The discharge summaries were also used to evaluate the generalizability of the multi-task model and provide data with automatically detected labels. We experimented with using the entirety of these notes vs. only the "Social History" and "History of Present Illness" sections. Performance was similar, so the smaller subset was used. Note sections were identified using simple pattern matching. Some notes in the MIMIC-III corpus include extraneous line breaks within sentences. All of the lines within a given section were merged into a single line, and then a sentence boundary detector [115] was used to parse the section into sentences. The extracted sections resulted in a corpus of 19M tokens from 113K notes for word embedding training.

[1] http://mtsamples.com

[2] http://depts.washington.edu/bionlp/index.html?corpora

4.3.2 Task

In Yetisgen and Vanderwende [19]'s experimentation on YVnotes, a separate module was designed to determine whether or not a sentence is associated with any substance events (single binary detector), and the result was used to filter out sentences with no relevant information in both frameworks. To be consistent with Yetisgen and Vanderwende [19], the prediction of labeled arguments, specifically *Status*, was treated as a sentence-level text classification task, and the identification of span-only arguments was treated as a sequence tagging problem, using the BIO approach. Yetisgen and Vanderwende [19] evaluated performance using precision (P), recall (R), and F1 score (F1). *Status* performance was computed at the sentence-level, micro-averaging across labels. Span-only argument performance was evaluated at the token-level, which is common in clinical information extraction tasks such as this, particularly when the data set size is limited. Similar to Yetisgen and Vanderwende [19], a separate detector is used to filter sentences without relevant information.

4.3.3 Discrete Models

Trigger: A substance detection model was trained to predict the presence of any events (*Alcohol*, *Drug*, or *Tobacco*) within a sentence. The predictions from the substance detection model were used as a first-stage mask, such that subsequent modeling only used sentences that were predicted to contain a substance event. The substance detection model was created using logistic regression (LR). The same substance detection model was used as the first-stage mask for both discrete and neural modeling approaches. Substance indicator models were also trained to predict each individual substance using LR. The substance indicator predictions were used as the trigger. LR models used word n-gram features (unigram-trigram) and gazetteer features. The gazetteer features consisted of three binary features, indicating alcohol, drug, or tobacco. The gazetteer word lists were generated by searching WordNet for hyponyms of each substance, resulting in 324 alcohol, 271 drug, and 46 tobacco words (search terms included "tobacco," "alcoholic_drink," "sedative," "narcotic," and "controlled_-

substance").

Labeled Arguments: The only labeled argument is *Status*. *Status* is classified using separate Maximum Entropy (MaxEnt) models for each substance. The MaxEnt models used word n-gram features (unigram-trigram) and same gazetteer features as the LR models.

Span-only Arguments: The span-only arguments (*Amount*, *Frequency*, *Exposure History*, *Quit History*, and *Type*) are extracted using linear-chain CRF models [54]. Features included word n-grams (unigram-trigram), POS tags, capitalization indicators (lowercase, uppercase, title case, and other), and string type indicators (punctuation, number, alphabetic, alphanumeric, and other).

Span-only arguments are extracted using two approaches. In the first approach, a separate CRF model is trained for each substance to extract substance-specific *Amount*, *Frequency*, *Exposure History*, and *Quit History* arguments. In the second approach, these labels are merged across all substances, and a single CRF model is created to extract these substance-independent arguments. Then, the substance indicator models are used to associate a substance with each extracted argument using the heuristic that any predicted entity in the sentence is assigned to all substances detected for the sentence. In the results in Section 4.5, the first approach is referred to as "CRF," and the second, two-stage, approach is referred to as "CRF+LR."

4.3.4 Multi-task Substance Extractor

The Multi-task Substance Extractor is as defined in Section 3.2, but it is applied after the same LR model as in the discrete model was used to filter out sentences with no events. Word embeddings were pretrained using word2vec [32] on the MIMIC-III data and held constant during the training of the Multi-task Substance Extractor, due to the limited size of the annotated data. A 38K token vocabulary was chosen using tokens that occurred at least five times in the MIMIC-III subset, corresponding to a relatively low out-of-vocabulary rate of 2.3%.

4.4 Experimental Setup

The annotated corpus was split into training and test sets using an 80%/20% split. All models were tuned using 5-fold cross validation (CV) on the training set. After determining the best configuration, models were retrained on the entirety of the training set. Only the highest performing discrete and multi-task models were applied to the withheld test set.

Discrete model tuning included the selection of feature types, including n-gram order, regularization type (L1 or L2), and regularization strength. Discrete models were created using Python scikit-learn [116]. The best performing MaxEnt *status* models used the following features: unigrams for alcohol; unigrams and gazetteer for drug; and unigrams-bigrams and gazetteer for tobacco.

Multi-task Substance Extractor tuning included: selecting the word embedding training set, determining the connections within the multi-task network, regularization strength, LSTM size, learning rate, number of epochs, CNN filter widths, number of CNN filters at each width, and layer normalization. The Multi-task Substance Extractor was regularized using dropout in the LSTM and *status* CNN layers. The model was trained in multiple stages. In the first stage, the entire model was trained jointly to minimize average loss across all classifiers (tasks), updating all graph variables except for the pre-trained word embeddings. In the second stage, the learned parameters in the LSTM were frozen, and each set of argument-specific classifiers were trained jointly, with the other classifier variables fixed. The argument-specific classifiers were trained in the following order: 1) *trigger*, 2) labeled arguments, 3) *Amount*, *Frequency*, *Exposure History*, and *Quit History*, and 4) *Type*. The Multi-task Substance Extractor configuration that achieved the best overall performance had the following configuration: H size = 200, H dropout = 20%, # joint epochs = 1000, # argument-specific epochs = 50, learning rate = 0.005, batch size = 20, CNN dropout = 40%, # CNN filters = 10, and CNN filter widths = [2, 3]. The Multi-task Substance Extractor was created models using Google's TensorFlow [117].

All of the CV and test results presented reflect the end-to-end performance, i.e., any

event detection errors impact the final results. Performance is presented in terms of the true positive count (TP), false negative count (FN), false positive count (FP), precision (P), recall (R) and F1 score (F1).

4.5 Results

The substance detector, which was used as a first-stage classifier, achieved a performance of F1=0.97 during CV and F1=0.98 on the test set. Table 4.3 presents the training-CV and test set *Status* prediction results for the MaxEnt and Multi-task Substance Extractor (MultiSub) models. Results are micro-averaged across labels. High performance was obtained for all substances in both data sets, with the Multi-task Substance Extractor giving slightly better results in all cases. The Multi-task Substance Extractor achieved higher performance on *Status* extraction than in Yetisgen and Vanderwende [19] (F1 scores of 0.91 for alcohol, 0.86 for drug, and 0.80 for tobacco on the test set), though the results are not directly comparable because of the train/test differences.

Table 4.3: Substance Status performance

Event Type	**Model**	**Train-CV**				**Test**			
		TP	**P**	**R**	**F1**	**TP**	**P**	**R**	**F1**
Alcohol	MaxEnt	168	0.91	0.84	0.87	49	0.91	0.91	0.91
	MultiSub	178	0.89	0.89	**0.89**	52	0.93	0.96	**0.95**
Drug	MaxEnt	94	0.95	0.82	0.88	20	0.87	0.77	0.82
	MultiSub	99	0.93	0.86	**0.89**	24	0.96	0.92	**0.94**
Tobacco	MaxEnt	172	0.84	0.80	0.82	49	0.83	0.82	0.82
	MultiSub	185	0.85	0.86	**0.86**	53	0.88	0.88	**0.88**

Table 4.4 presents the span-only argument CV development and test set performance results for the discrete and Multi-task Substance Extractor models. Results are presented for each argument, micro-averaged across substances. For *Amount*, *Frequency*, *Exposure History*, and *Quit History* extraction, the best performing CRF models used the following

features: unigrams and gazetteer for alcohol; unigrams, POS, capitalization, string type, and gazetteer for drug; and unigrams and POS for tobacco. For the "CRF+LR" approach, the best performing CRF model used unigrams-trigrams, POS, capitalization, string type, and gazetteer features. The substance indicator models used in the "CRF+LR" approach achieved F1 performance of 0.95 for alcohol, 0.93 for drug, and 0.97 for tobacco during CV and F1 performance of 0.95 for alcohol, 0.88 for drug, and 0.95 for tobacco on the test set. The best-performing CRF model for *Type* extraction used unigram and string type features. On the test set, the Multi-task Substance Extractor produced similar or better results in all categories. The performance is robust in moving to the test set except for a small degradation for *Exposure History*, which is from a drop in the drug subset where data is sparse. For *Amount*, *Frequency*, *Exposure History*, and *Quit History*, the Multi-task Substance Extractor outperformed Yetisgen and Vanderwende [19] (F1 scores of 0.76 for *Amount*, 0.75 for *Frequency*, 0.63 for *Exposure History*, and 0.75 for *Quit History*)[19] and both CRF baselines across each argument.

Table 4.4: Substance span-only argument performance

Event Type	**Model**	**Train-CV**				**Test**			
		TP	**P**	**R**	**F1**	**TP**	**P**	**R**	**F1**
Type	CRF	119	0.96	0.69	0.80	31	0.97	0.91	**0.94**
	MultiSub	123	0.94	0.72	**0.81**	31	0.94	0.91	0.93
Amount	CRF	151	0.90	0.61	0.73	51	0.86	0.64	0.73
	CRF+LR	168	0.65	0.68	0.66	61	0.73	0.76	0.75
	MultiSub	180	0.82	0.73	**0.77**	65	0.84	0.81	**0.83**
Exposure History	CRF	79	0.73	0.60	0.66	19	0.66	0.32	0.43
	CRF+LR	89	0.79	0.67	**0.73**	31	0.53	0.52	0.53
	MultiSub	85	0.83	0.64	**0.73**	37	0.76	0.62	**0.68**
Frequency	CRF	104	0.88	0.56	0.68	32	0.86	0.60	0.71
	CRF+LR	120	0.63	0.64	0.64	35	0.80	0.66	0.72
	MultiSub	142	0.85	0.76	**0.80**	39	0.83	0.74	**0.78**
Quit History	CRF	59	0.91	0.52	0.66	18	0.60	0.67	0.63
	CRF+LR	82	0.81	0.72	**0.76**	18	0.82	0.67	0.73
	MultiSub	65	0.74	0.57	0.64	21	0.84	0.78	**0.81**

Table 4.5 presents the detailed *Status* test set results for *Alcohol*, *Drug*, and *Tobacco*. The *none* case is the dominant class for all substances. Again, the Multi-task Substance Extractor gives the best result on the test set. For the cases where the difference between CV and test performance is greatest, the numbers of samples are small.

Table 4.5: Substance use Status test set performance by argument subtype

Event Type	Argument Subtype	Model	TP	P	R	F1
Alcohol	current	MaxEnt	17	0.89	0.85	0.87
		MultiSub	20	0.91	1.00	**0.95**
	none	MaxEnt	30	0.91	0.97	0.94
		MultiSub	30	0.97	0.97	**0.97**
	past	MaxEnt	2	1.00	0.67	**0.80**
		MultiSub	2	0.67	0.67	0.67
Drug	current	MaxEnt	1	1.00	0.20	0.33
		MultiSub	3	1.00	0.60	**0.75**
	none	MaxEnt	19	0.90	0.95	0.93
		MultiSub	20	0.95	1.00	**0.98**
	past	MaxEnt	0	0.00	0.00	0.00
		MultiSub	1	1.00	1.00	**1.00**
Tobacco	current	MaxEnt	8	0.89	0.73	**0.80**
		MultiSub	8	0.89	0.73	**0.80**
	none	MaxEnt	34	0.85	0.92	0.88
		MultiSub	35	0.95	0.95	**0.95**
	past	MaxEnt	7	0.70	0.58	0.64
		MultiSub	10	0.71	0.83	**0.77**

4.6 Application

The substance detection and Multi-task Substance Extractor were used to predict substance events in the discharge summaries within the MIMIC-III corpus (59.7K notes). The substance detection model predicted 40.3K MIMIC-III discharge summaries to have a substance event. Table 4.6 presents the argument occurrence counts by event type from the unsupervised labeling of these notes. Of the notes predicted to contain at least one substance event,

50 randomly sampled notes were hand-scored to evaluate the precision of *Status* labels. Table 4.7 summarizes the manual review of 50 discharge summaries. The *Status* classification precision was high for all substances (0.84-0.89).

Table 4.6: MIMI-III annotation summary

Argument Type	**Alcohol**	**Drug**	**Tobacco**
Status	44,536	20,725	45,244
Type	4,756	14,509	11,443
Amount	13,262	2,551	11,298
Frequency	12,200	355	7,183
Exposure History	770	396	4,639
Quit History	176	72	10,241

Table 4.7: MIMIC-III Status evaluation

Event Type	**TP**	**FP**	**P**
Alcohol	76	14	0.84
Drug	80	10	0.89
Tobacco	80	10	0.89

4.7 Conclusions

In summary, we implemented the Multi-task Substance Extractor to extract substance use information from clinical notes, achieving state-of-the-art performance. The Multi-task Substance Extractor outperformed discrete baselines comparable to Yetisgen and Vanderwende [19] on the test set for all entities, except *Type*. For *Type*, the multi-task model and the CRF baseline had similar performance. Excluding *Type*, the performance gap between the multi-task model and the discrete baselines was larger on the test set than the training CV runs, suggesting the Multi-task Substance Extractor generalized better to the test set. The improved extraction performance of the Multi-task Substance Extractor can benefit downstream applications, including clinical decision support systems. Additionally, the multi-task modeling framework is well-suited to extending the current model to handle other types of information, such as socio-demographic, behavioral, and environmental exposure factors, as we explore in the next chapter. A limited evaluation of the unsupervised labeling of the MIMIC-III discharge summaries indicated the multi-task model maintained high precision in the prediction of *Status*.

This work is limited by the size and homogeneity of the annotated corpus, which may negatively impact generalizability of the extraction models to other clinical data sets. However, the initial results on the MIMIC-III data are encouraging.

Chapter 5

SOCIAL DETERMINANTS OF HEALTH

5.1 Overview

Decreasing life expectancy may be partly attributable to deteriorating social determinants of health (SDOH) [118, 119]. For example, substance abuse (including alcohol, drug, and tobacco use) is increasingly recognized as a key factor for morbidity and mortality [8–10]. More Americans are living alone, leading to increased social isolation and negative health outcomes [11]. Employment and occupation impact income, societal status, hazards encountered, and health [120]. Socioeconomic status impacts transportation, exercise, and air quality exposure, which in turn influences lung and coronary health [121, 122].

This chapter presents a new annotated corpus, SHAC. To achieve high SDOH extraction performance that generalizes across clinicians, institutions, and specialties, annotated corpora must be large and diverse. Currently available corpora with SDOH annotations are lacking in either annotation detail, public availability, size, and/or heterogeneity. SHAC addresses limitations of existing corpora by providing a relatively large, heterogeneous corpus with high quality, detailed SDOH annotations. SHAC includes detailed event-based annotations for 12 critical SDOH: substance use (alcohol, drug, and tobacco), physical activity, employment, insurance, living status, sexual orientation, gender identity, country of origin, race, and environmental exposure. *SHAC* is comprised of 4,480 social history sections. SHAC utilizes clinical notes from MIMIC-III [23] and an existing data set from the UW and Harborview Medical Centers. It includes event-based annotations for more than 55K annotated spans and 18K distinct events across four note types.

Hand annotation of detailed SDOH information in clinical notes is costly, and many critical SDOH are infrequent. To address these budget and data sparsity limitations, the corpus

development used active learning to select samples for annotation. Because extracting the SDOH events is a complex sequence labeling task, standard active learning methods are not practical. This work introduces the novel Active Learning using Surrogate Classifiers (ALSC) framework that uses a simplified surrogate task for assessing sample informativeness. Our experiments show that this method increases the diversity and richness of the annotations and improves extraction performance for a variety of event types. The largest performance gains achieved by the active learning framework are associated with infrequent, but extremely important risk factors, like drug use, homelessness, and unemployment.

With the annotated SHAC corpus, we provide the Multi-task Event Extractor and present the first reported extraction results on SHAC for the most frequently annotated SDOH: substance use, employment, and living status. The event extraction model identifies substance use, employment, and living status events at 0.89-0.98 F1 and characterizes the status of these determinants with 0.81-0.96 F1.

This chapter expands on the substance use extraction work of the previous chapter to include a broader set of SDOH and is largely based on Lybarger et al. [21], which is currently under review. In Lybarger et al. [21], I contributed to experiment design, data analysis, and writing, performed all software development (programming), and executed all experimentation.

5.2 Materials

5.2.1 Data

This work utilized two clinical data sets without SDOH annotations: *MIMIC-III* and *UW Dataset*. *MIMIC-III* (referred to here as *MIMIC*) is a publicly available, deidentified health database for over 40K critical care patients at Beth Israel Deaconess Medical Center from 2001-2012 [23]. MIMIC contains clinical notes, diagnosis codes, and other data. This work utilized 60K MIMIC discharge summaries. *UW Dataset* is an existing clinical data set from the UW and Harborview Medical Centers generated between 2008-2019. This work

utilized 83K emergency department, 22K admit, 8K progress, and 5K discharge summary notes from UW Dataset. An existing corpus with SDOH annotations, *YVnotes*, was used for model training during active learning [19].

5.2.2 Annotation Scheme

We created detailed annotation guidelines for 12 SDOH (*event types*). Table 5.1 summarizes the annotation of the SHAC event types extracted using the Multi-task Event Extractor: *Alcohol*, *Drug*, *Tobacco*, *Employment*, and *Living Situation*. Table A.1 in the Appendix contains a summary of all annotated event types.

Event type, e	**Argument type,** a	**Argument subtypes,** y_l	**Span examples**
Alcohol, Drug, or Tobacco	Trigger*	–	"alcohol"
	Status*	{none, current, past}	"denies," "smokes"
	Duration	–	"for 8 years"
	History	–	"seven years ago"
	Type	–	"beer," "cocaine"
	Amount	–	"2 packs"
	Frequency	–	"daily," "monthly"
Employment	Trigger*	–	"works," "nurse"
	Status*	{employed, unemployed, retired, on disability, student, homemaker}	"works"
	Duration	–	"for five years"
	History	–	"15 years ago"
	Type	–	"nurse"
Living status	Trigger*	–	"lives"
	Status*	{current, past, future}	"lives," "lived"
	Type*	{alone, with family, with others, homeless}	"with husband"
	Duration	–	"for 6 months"
	History	–	"last month"

Table 5.1: SHAC annotation guideline summary for the most frequent event types. *indicates the argument is required.

5.2.3 Annotation Cycle

Social history sections, referred to here as *samples*, were extracted from MIMIC and the UW Dataset, using pattern matching to identify section headings (alphanumeric, forward slash, backslash, ampersand, or white space characters followed by a colon). SHAC includes *train*, *development*, and *test* sets. Samples for the train set were randomly and actively selected. Training samples were randomly selected for initial model training in active learning, then the initial model is used in actively selecting samples to bias the training set towards diverse samples that frequently contain the phenomena of interest. All development and test samples were randomly selected to approximate the true distribution within the corpora used. Samples were annotated by four 4^{th}-year medical students through 12 rounds of annotation (8 randomly selected and 4 actively selected). Table A.2 in the Appendix describes each round of annotation. The first two rounds were randomly sampled and double-annotated, to assess inter-annotator agreement. After the initial annotation round, the annotation guidelines were revised, and the initial annotations were updated.

5.2.4 Evaluation and Annotation Scoring

We treat event annotation and extraction as a slot filling task, as this is most relevant to secondary use applications. As such, there can be multiple equivalent span annotations. Figure 5.1 presents the same sentence annotated by two annotators (labeled A and B), along with the populated slots. Both annotators labeled two *Drug* events: *Event 1* and *Event 2*. Event 1 describes past intravenous drug use (IVDU), and Event 2 describes current cocaine use. Event 1 is annotated identically by both annotators. However, there are differences in the annotation spans of Event 2, specifically for the *Trigger* ("cocaine" versus "cocaine use") and *Status* ("use" vs. "Recent"). From a slot perspective, the annotations for Event 2 are equivalent. Thus, scoring of automatic detection and annotator agreement is based on relaxed span match criteria, as described below. Trigger and argument performance is evaluated using precision, recall, and F1, micro averaged over the event types, argument

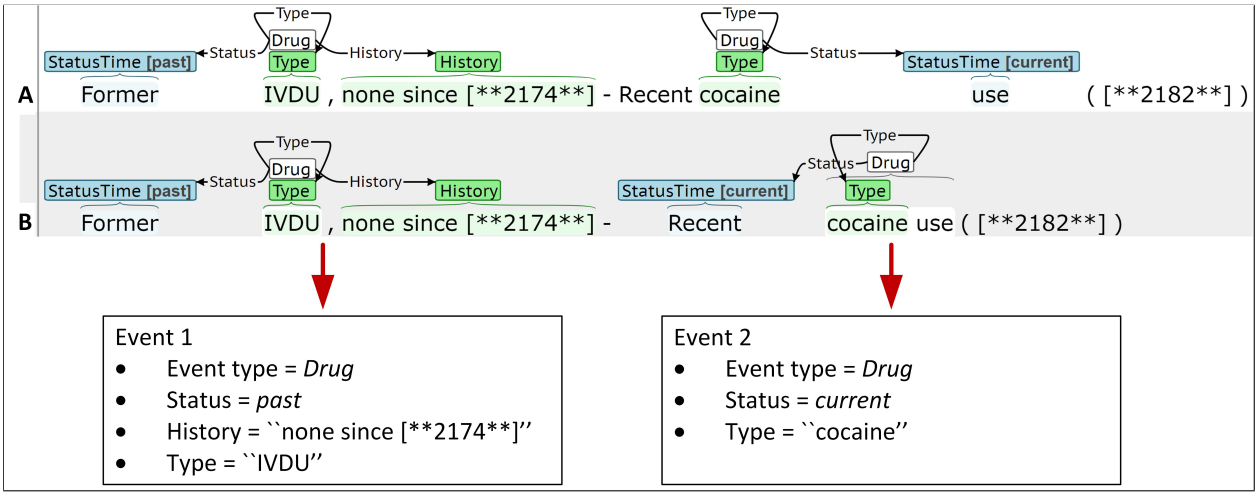

Figure 5.1: SHAC annotation examples describing event extraction as a slot filling task

types, and/or argument subtypes.

Trigger: Triggers, T_i, are represented by a pair (event type, e_i; token indices, x_i). For *Event 2* in Figure 5.1, $T_{A,2} = (e_{A,2} = Drug; x_{A,2} = [8])$ and $T_{B,2} = (e_{B,2} = Drug; x_{B,2} = [8, 9])$. Triggers of the same event type, e, are aligned by minimizing the distance between span centers computed from the token indices. Trigger equivalence is defined as

$$T_i \equiv T_j \text{ if } (e_i \equiv e_j) \wedge (T_i \text{ aligned with } T_j). \tag{5.1}$$

Although there are two drug events in the Figure 5.1 example, $T_{A,2}$ aligns with $T_{B,2}$ because of the overlapping spans

Argument: Events are aligned based on trigger equivalence, and the arguments of aligned events are compared using different criteria for *labeled arguments* and *span-only arguments*. Labeled arguments, A^l_i, are represented as a triple (argument type, a_i; token indices, x_i; subtype, l_i). For *Event 2* in Figure 5.1, $A^l_{A,2} = (a_{A,2} = Status; x_{A,2} = [9], l_{A,2} = current)$ and $A^l_{B,2} = (a_{B,2} = Status; x_{B,2} = [7], l_{B,2} = current)$. For labeled arguments, the argument type, a, and subtype, l, capture the salient information and equivalence is defined

as

$$A_i^l \equiv A_j^l \text{ if } (T_i \equiv T_j) \wedge (a_i \equiv a_j) \wedge (l_i \equiv l_j). \tag{5.2}$$

Span-only arguments, A_i^s, are represented as a pair (argument type, a_i; token indices, x_i). For *Event 2* in Figure 5.1, $A_{A,3}^s = (a_{A,3} = Type; x_{A,3} = [7])$ corresponds to "cocaine." Span-only arguments are not easily mapped to a fixed set of classes, and the identified span, x, contains the most salient argument information. Span-only arguments with equivalent triggers and argument types, $(T_i \equiv T_j) \wedge (a_i \equiv a_j)$, are compared at the token-level (rather than the span-level) to allow partial matches. Partial match scoring is used as partial matches can still contain useful information.

We evaluate annotator agreement using Cohen's Kappa, κ, coefficient, where higher κ denotes better annotator agreement [123]. Calculating κ for the full event structure is not informative, because the probability of random agreement is close to zero. Instead, we calculate κ for trigger annotation in the subset of sentences with zero or one trigger for a given event type in either set of annotations, which covers most of the data. We focus on this subset of sentences, because triggers for a given event type are equivalent, if the annotated sentences both include one trigger of that type. We assess annotator agreement on the full event structure using F1 scores.

5.2.5 Annotation Statistics

SHAC consists of 4,480 annotated social history sections (70% train, 10% development, 20% test). Table 5.2 presents the corpus composition by source. The SHAC training samples are 29% randomly selected and 71% actively selected. All development and test data are randomly sampled.

Figure 5.2 presents the event type distribution. The most frequent event types are *Drug*, *Tobacco*, *Alcohol*, *Living status*, and *Employment*, with the remaining event types occurring infrequently.

Source	Train	Dev	Test
MIMIC	1316	188	376
UW Dataset	1820	260	520
TOTAL	3,136	448	896

Table 5.2: SHAC composition by source

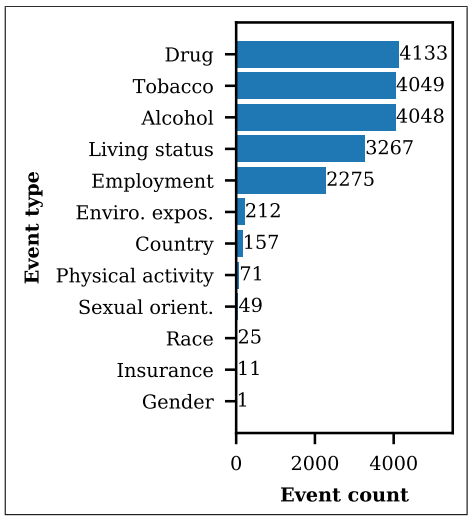

Figure 5.2: Event type distribution

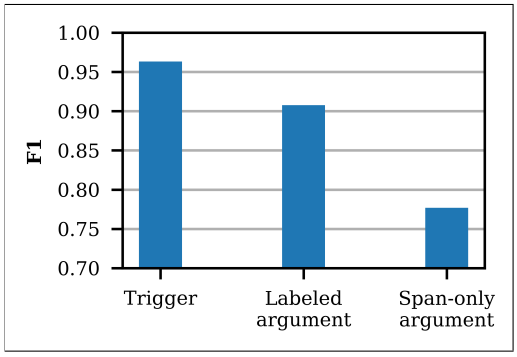

Figure 5.3: Annotator agreement for 300 doubly annotated MIMIC samples

Figure 5.3 presents the annotator agreement for all event types in terms of F1 score for 300 doubly annotated notes from the first two rounds of annotation. For *Alcohol*, *Drug*, *Tobacco*, *Employment*, and *Living status*, trigger κ is $0.94 - 0.97$. For the remaining event types, trigger κ is $0.61 - 0.90$. κ is calculated for sentences with 0-1 events for each type ($\geq 99\%$ of all sentences). The trigger agreement is very high, in terms of F1 and κ, indicating the annotators are consistently identifying and distinguishing between events. The argument agreement is also high for labeled arguments. The somewhat lower agreement for span-only arguments is primarily due to small differences in the start and end token spans (e.g. "construction worker" vs. "construction").

5.3 *Active Learning*

This section presents the active learning framework used create SHAC and describes the associated performance gains.

5.3.1 Methods

A portion of the SHAC training samples were selected using active learning, where a sample is a social history section. Specifically, batch-mode active learning was used to facilitate coordination with human annotators through the cyclical process shown in Figure 5.4. A

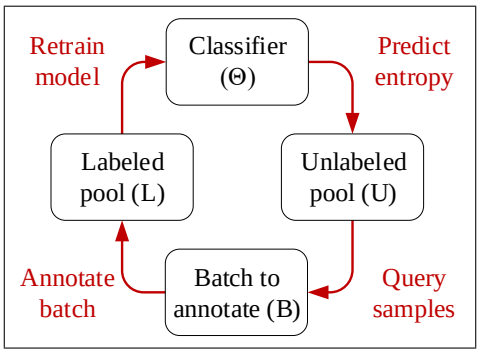

Figure 5.4: Active learning annotation cycle

batch of samples, B, was annotated and added to the labeled pool, L. The surrogate classifier was trained on L and then generated uncertainty scores for unlabeled data U. Using the uncertainty scores, the query function identified the next batch of samples, B. This process was repeated until the annotation objective was met.

Similar to Wu and Ostendorf [90], a query score is designed to combine informativeness and diversity scores of a batch of samples, B. Here, the query score has the form:

$$Q(B) = \sum_{i \in B} (1 - s_i)^{\alpha} u(i) \tag{5.3}$$

where $u(i)$ is the uncertainty entropy of sample i, s_i is the similarity score of sample i relative

to B, and $(1-s_i)$ is the diversity score. α is a weight used to balance the relative importance of the two scores ($\alpha > 0$). The objective is to maximize the batch score, $Q(B)$. We explored different forms for the uncertainty and similarity scores for this multi-label scenario. We implemented a greedy approach to selecting examples, as shown in Algorithm 1.

Algorithm 1: Greedy query function

Input: unlabeled samples U, batch size N

Output: batch of samples B

$B \leftarrow \emptyset$;

while $|B| < N$ **do**

  $k \leftarrow \text{argmax}_{i \epsilon U}\, Q(B \cup i)$;

  $B \leftarrow B \cup \{k\}$;

  $U \leftarrow U - \{k\}$;

end

Diversity: Sample diversity is assessed in the observation space using two different similarity metrics: *average similarity* and *maximum similarity*, defined as

$$s_i^a = \frac{1}{|B|} \sum_{j \in B, j \neq i} a_{j,i} \qquad s_i^m = \max_{j \in B, j \neq i} a_{j,i},$$

respectively, where $a_{j,i}$ is the cosine similarity of samples j and i.

The maximum similarity approach is a stricter condition that pushes the batch of samples farther apart in the observation space, especially with larger batch sizes. Similar to Lilleberg et al. [124], unsupervised vector representations of samples were learned as the TF-IDF weighted averages of pre-trained word embeddings. Word embeddings were created using the word2vec skip-gram model [32] and trained on the entirety of the MIMIC discharge summaries (not just the social history sections). Separate TF-IDF weights were calculated for MIMIC and UW Dataset samples.

Uncertainty: Active learning query functions typically assess sample informativeness

(uncertainty) using the target classification task. In this work, sample uncertainty was assessed using a simplified surrogate classification task, as a proxy for the more complex event-based annotation scheme. The SHAC annotation scheme includes some arguments (e.g. *Status* for *Alcohol*) that are more predictive of negative health outcomes than others (e.g. *Type* for *Alcohol*), and the prediction uncertainty varies across event types and arguments. To ensure the query function biases selection towards the most salient arguments, each of the five most frequent event types in SHAC were represented using the single argument that is most predictive of negative health outcomes: *Alcohol-Status*, *Drug-Status*, *Tobacco-Status*, *Employment-Type*, and *Living status-Status*. To cover samples with multiple events of the same type (e.g. both previous and current tobacco use described), an additional class, "multiple," is added to the argument subtypes, y_l, in Table 5.1, $y_c = \{y_l \cup \text{“}multiple\text{”}\}$.

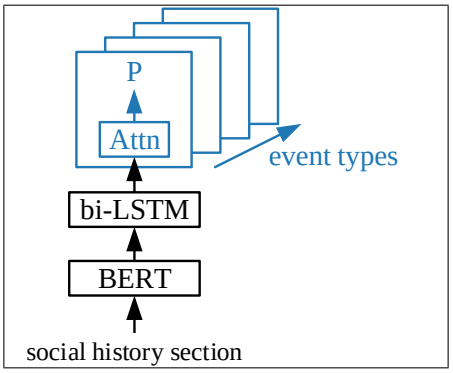

Figure 5.5: Surrogate Classifier used to assess sample uncertainty in active learning

The text classification model, *Surrogate Classifier* in Figure 5.5, was used to assess sample uncertainty. The Surrogate Classifier operates on a sample, as a single sequence of n tokens without line breaks. The input social history section is mapped to contextualized word embeddings using *Bio+Discharge Summary BERT* [41], a version of *BERT* [35] trained on clinical text from MIMIC. The BERT output feeds into a bidirectional long short-term memory (bi-LSTM) layer, the output of which feeds into event-specific output layers. Separate self-attention (Attn) output layers for each event type make sample-level predictions.

Details of the Surrogate Classifier are similar to the shared and event-argument layers of the Multi-task Event Extractor in Section 3.3. The Surrogate Classifier generates a set of five multi-class predictions for each sample, one for each event type.

We explored two approaches to characterizing sample uncertainty: i) the sum of the five event entropy values, similar to previous work [48, 125–127], and ii) entropy for an individual event type, iterating over all types (referred to as "loop"). As a "loop" example, *Alcohol-Status* entropy is used for sample 1, *Drug-Status* entropy is used for sample 2, and so forth, starting over with *Alcohol-Status* entropy for sample 6. The second method was motivated by the concern that summing the entropy values (referred to as "sum") could overly bias the selection process in favor of high-entropy event types, reducing the diversity of event types.

5.3.2 Experiments & Results

Query strategy selection: Due to limitations in the annotation budget, the query strategy was determined early in the annotation effort. We used the first 700 annotated samples, L_Q, which consists of random MIMIC samples. L_Q was partitioned into $L_Q^T \coloneqq \{620 \text{ train samples}\}$ and $L_Q^D \coloneqq \{80 \text{ development samples}\}$. For random sampling and each active sampling configuration, 10 runs were performed:

(i) $L_{T1} \leftarrow 100$ samples from L_Q^T. Train model, M_1, on L_{T1}.

(ii) $L_{T2} \leftarrow 100$ samples from $\{L_Q^T - L_{T1}\}$ (random or active). Train model, M_2, on $\{L_{T1} \cup L_{T2}\}$.

(iii) Evaluate the performance of M_2 on L_Q^D

Active sampling experimentation included different uncertainty types ("loop" vs. "sum"), similarity types ("average" vs "maximum"), and α values $\{0.1, 1, 2\}$. The hyperparameters of the Surrogate Classifier were tuned on L_Q^D (parameter values in Table A.3 of the Appendix). Table 5.3 presents the results for the best α value for each uncertainty-similarity type combination. Performance is assessed using precision, recall, and F1-score, micro-averaged across classes and event types. All active learning configurations outperform the random baseline with significance ($p < 0.05$). The best configuration, uncertainty type ="sum", similarity

Uncertainty	**Similarity**	α	**F1**
loop	average	1.0	0.788*
loop	maximum	0.1	0.776*
sum	average	2.0	0.788*
sum	maximum	0.1	0.794*

Table 5.3: Query function tuning performance. *indicates statistical significance ($p < 0.05$) relative to a random baseline of 0.752 F1.

type="maximum", and $\alpha = 0.1$, was used in active selection.

Active learning performance: After the first round of active learning, performance of the Surrogate Classifier was evaluated to confirm the effectiveness of the active learning framework. Model training included the sets: $L_Y \coloneqq \{284$ YVnotes samples$\}$ and $L_R \coloneqq \{532$ random MIMIC train samples$\}$. YVnotes was used to train the Surrogate Classifier to improve its accuracy and thereby obtain a better uncertainty score. L_R was partitioned into $L_R^I \coloneqq \{288$ initial training samples$\}$ and $L_R^P \coloneqq \{244$ remaining samples, $L_R - L_R^I\}$. For the first round of active selection, an initial model, M_I, was trained on $\{L_R^I \cup L_Y\}$ and used to select 400 MIMIC samples, L_A. L_R^P was withheld when training M_I to validate the active learning approach. Hyperparameters were tuned on $L_D \coloneqq \{188$ MIMIC development samples$\}$ (parameter values in Table A.3 of the Appendix).

Figure 5.6 presents the performance of four cases on $L_E \coloneqq \{376$ MIMIC test samples$\}$:

- *MIMIC-only initial*: Models trained only on the MIMIC samples, L_R^I.
- *initial*: The initial model, M_I, trained on $\{L_R^I \cup L_Y\}$ and from the first active round.
- *+random*: Models trained on the initial set and additional random samples, $\{L_R^I \cup L_Y \cup L_R^P\}$.
- *+active*: Models trained on the initial set and additional active samples, $\{L_R^I \cup L_Y \cup$ 244 from $L_A\}$.

For *MIMIC-only initial*, *+random*, and *+active*, 10 runs were performed to account for variance in model initialization. For *MIMIC-only initial* and *+random*, the training sets are

fixed, as all data is used each run. For *+active*, the training set varies because only a subset of L_A is randomly selected each run, so sampling variance is introduced. The error bars in Figure 5.6 indicate the standard deviation of the F1 scores across runs.

Comparing *MIMIC-only initial* to *initial* demonstrates that including YVnotes improves performance. Adding active samples to the initial training set yields a statistically significant improvement over adding random samples ($p < 0.06$), demonstrating the effectiveness of the active learning framework on the surrogate task.

The effectiveness of the active learning framework under the same conditions as Figure 5.6 and on the target event extraction task using the Multi-task Event Extractor is presented in Figure 5.7, where scores are averaged across event types.[1] The details of the Multi-task Event Extractor are presented in Section 3.3. The performance achieved by adding active samples outperforms that of adding random samples for labeled argument and span-only argument extraction, with significance ($p < 0.01$). The addition of actively selected notes improved extraction performance, relative to the random baseline, across most argument types and subtypes. However, **the largest active learning performance gains were achieved for prominent health risk factors, including past and current drug use, current tobacco use, unemployment, homelessness, and living with others** (+0.09 ΔF1 for *current Drug Status*, +0.14 ΔF1 for *past Drug Status*, +0.07 ΔF1 for *current Tobacco Status*, +0.04 ΔF1 for *unemployed Employment Status*, +0.06 ΔF1 for *homeless Living Status Type*, and +0.07 ΔF1 for *with others Living Status Type*). The difference in trigger performance is not statistically significant. This result validates the use of the simplified surrogate text classification task as a proxy for the more complex event extraction task. After validating the active learning strategy, three additional rounds of active selection were performed (see Table A.2 of the Appendix for details), and the Surrogate Classifier model was retrained prior to each active round. Due to the limited number of random samples, further comparisons of active vs. random sampling are not possible.

---

[1]For the Multi-task Event Extractor, we exclude L_Y since YVnotes do not include all of the labeled phenomena of SHAC.

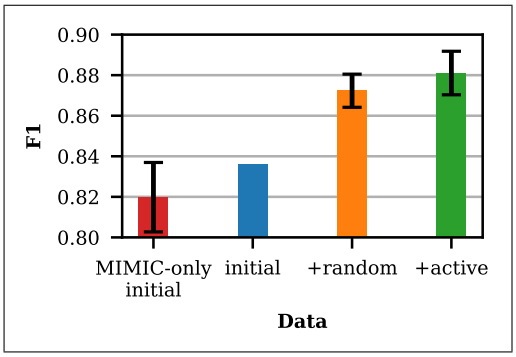

Figure 5.6: Surrogate Classifier performance with random and active samples, evaluated on MIMIC test samples.

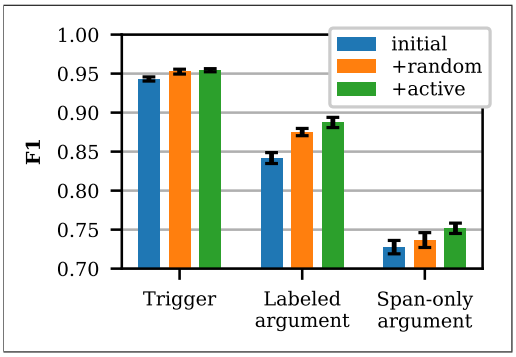

Figure 5.7: Multi-task Event Extractor performance with random and active samples, evaluated on MIMIC test samples.

We hypothesized the Surrogate Classifier uncertainty would bias the selection process to include more health risk factors (e.g. positive substance abuse, unemployment, being on disability, homelessness, etc.), which tend to be more challenging to automatically extract than less risky behavior (e.g. no substance use, being employed, and living with family). Active learning successfully identified samples with richer, more detailed SDOH descriptions. Figure 5.8 presents the label frequency per sample (note section) for random and active samples for the entirety of SHAC. The frequency of positive substance use ($Status \in \{current, past\}$) is 83% higher in active samples than random samples, with the frequency of positive drug use 151% higher with active selection. Active sampling produced higher rates for all *Employment Status* labels, except *retired*. Descriptions of retirement, tend to have low entropy, because of the reliable presence of keywords like "retired" or "retirement." Regarding *Living Status*, the rate of *homeless* is 109% higher in active samples than random samples, and the rate of *with others* is 81% higher. The rate of *alone* is slightly lower in active samples, likely due to lower entropy associated with the limited vocabulary used to describe living alone (e.g. "alone" or "by herself").

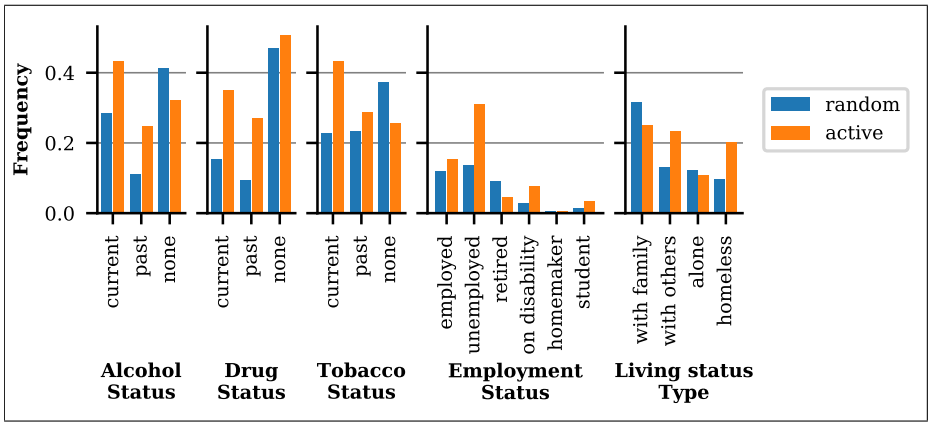

Figure 5.8: Label frequency per social history section, comparing random and active sampling

5.4 Event Extraction

5.4.1 Methods

The Multi-task Event Extractor used in this chapter is defined in Section 3.3. Experimentation included six labeled arguments: *Status* for *Alcohol*, *Drug*, and *Tobacco*; *Status* for *Employment*; and *Status* and *Type* for *Living status*, and 20 span-only arguments: *Duration*, *History*, *Type*, *Amount*, and *Frequency* for *Alcohol*, *Drug*, and *Tobacco*; *Duration*, *History*, and *Type* for *Employment*; and *Duration* and *History* for *Living status*. The Multi-task Event Extractor hyperparameters were tuned on the development set, L_D, and Table 5.4 presents the selected parameters.

5.4.2 Results

Figure 5.9 presents the trigger and argument performance of the Multi-task Event Extractor trained on the entire SHAC train set and evaluated on the MIMIC and UW Dataset test sets. Overall, performance is higher on MIMIC, even though there are more UW Dataset training samples, including more active samples. The UW Dataset portion of SHAC includes four different note types, whereas the MIMIC portion includes only one note type, which likely contributes to the lower performance on the UW Dataset.

Parameter	Figure 5.7, Figure 5.9, and Table 5.5
batch size	50
learning rate	0.005
maximum gradient L2 norm	0.5
maximum length	30
number of epochs	250
LSTM hidden size	100
dropout, input to LSTM	0.6
dropout, output of LSTM	0.4
dropout, self-attention	0.4

Table 5.4: Multi-task Event Extractor hyperparameters

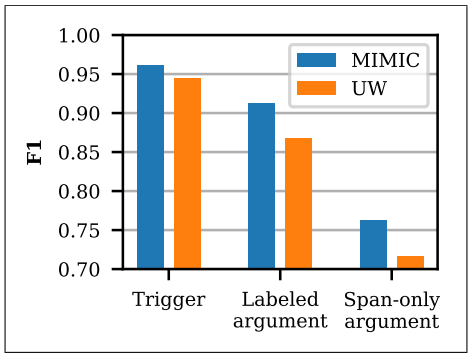

Figure 5.9: Multi-task Event Extractor micro-averaged trigger and argument performance comparing the MIMIC and UW Dataset test sets.

Table 5.5 presents detailed results for the same Multi-task Event Extractor model and data configuration as Figure 5.9. Trigger performance is greater than 0.89 F1 for all event types in both data sets. Labeled argument performance is similar in both data sets for *Alcohol* and *Tobacco Status*; however, there are performance differences for *Drug*, *Employment*, and *Living status* labeled arguments. In substance use *Status* prediction, the *none* label is typically less confusable and easier to predict than *past* and *current*. In the test set, the relative frequency of *none Status* labels for *Drug* events is higher in MIMIC samples (80%) than UW Dataset samples (57%), which contributes to the higher performance on MIMIC. *Living status Status* performance is lower in the UW Dataset, even though the distribution of

Field	Event type	Argument	MIMIC				UW			
			#	P	R	F1	#	P	R	F1
Trigger	Alcohol	–	314	0.99	0.96	0.97	404	0.97	0.99	0.98
	Drug	–	194	0.96	0.95	0.96	481	0.97	0.92	0.94
	Tobacco	–	324	0.98	0.95	0.97	432	0.97	0.97	0.97
	Employment	–	169	0.93	0.96	0.94	148	0.86	0.91	0.89
	Living status	–	244	0.96	0.97	0.97	343	0.93	0.88	0.90
Labeled argument	Alcohol	Status	314	0.92	0.89	0.90	404	0.92	0.94	0.93
	Drug	Status	194	0.91	0.89	0.90	481	0.85	0.80	0.82
	Tobacco	Status	324	0.91	0.89	0.90	432	0.91	0.90	0.90
	Employment	Status	169	0.84	0.88	0.86	148	0.79	0.83	0.81
	Living status	Status	244	0.96	0.95	0.96	343	0.92	0.86	0.89
		Type	244	0.93	0.93	0.93	343	0.85	0.78	0.81
Span-only argument	Alcohol	Amount, Duration, Frequency, History, Type	396	0.70	0.74	0.72	420	0.67	0.80	0.73
	Drug		219	0.67	0.75	0.71	583	0.62	0.63	0.62
	Tobacco		799	0.81	0.83	0.82	880	0.78	0.81	0.79
	Employment	Duration, History, Type	441	0.80	0.74	0.77	261	0.77	0.77	0.77
	Living status	Duration, History	21	0.21	0.57	0.31	57	0.19	0.26	0.22

Table 5.5: Multi-task Event Extractor trigger and argument role performance trained on the entire SHAC train set, evaluated on the MIMIC and UW Dataset test sets.

Status labels is similar in both data sets. *Living status Type* performance is 0.12 F1 higher in MIMIC than the UW Dataset. In the test set, the distribution of *Living status Type* labels differs greatly between the data sets with the UW Dataset at 37% *with family*, 22% *with others*, 26% *homeless*, and 15% *alone* and MIMIC at 57% *with family*, 16% *with others*, 2% *homeless*, and 25% *alone*. For the span-only arguments, the performance is calculated at the token-level and micro averaged across the arguments for each event type. Span-only argument performance is comparable for *Alcohol*, *Tobacco*, and *Employment*. However, it is higher for *Drug* span-only arguments in MIMIC than the UW Dataset. *Living status* span-only argument performance is very low for both data sets, primarily due to sparsity in the training set (only 167 *Duration* and *History* arguments among 3,267 *Living status* events).

5.4.3 Limitations

Although the Multi-task Event Extractor achieved high performance for most target phenomena, the extraction framework has several limitations. The Multi-task Event Extractor treats trigger and labeled argument prediction as a text classification task and can only represent a single event of a given type per sentence. Figure 5.10a presents predicted labels for a sentence with multiple gold *Drug* events describing current marijuana use and previous cocaine use. While the *Type* predictions in this example are correct, the *Status* prediction of *past* is incorrectly associated with both marijuana and cocaine. Of the sentences with at least one event in SHAC, 6% contain multiple events of the same type. Span-only arguments for each event type are extracted using a single CRF, which cannot accommodate overlapping spans. Figure 5.10b presents predictions for a sentence where the gold span-only argument spans overlap. The *Amount* is correctly labeled as "about 1 pint of vodka," but there should also be a *Type* argument of "vodka." Approximately 6% of span-only arguments in events of the same type overlap in SHAC. The Multi-task Event Extractor treats sentences independently. It does not incorporate context from the preceding sentences and cannot generate events that span multiple sentences. Figure 5.10c presents predictions for an example where past tobacco use is described in concurrent sentences. The first sentence includes a strong cue for *past Status*, "quit"; however, the *Status* in the second sentence is less clear without previous context. Fewer than 2% of SHAC events span multiple sentences.

5.5 Conclusions

We present a new clinical corpus, SHAC, with detailed event-based annotations for 12 SDOH. SHAC includes approximately 4.5K social history sections from multiple institutions and note types and contains frequent descriptions of alcohol, drug, and tobacco use, employment, and living status. Approximately 71% of the SHAC training set was selected using the novel active learning framework, ALSC, that utilizes a surrogate task for assessing sample uncertainty, which increased the prevalence of critical risk factors in the annotated training data,

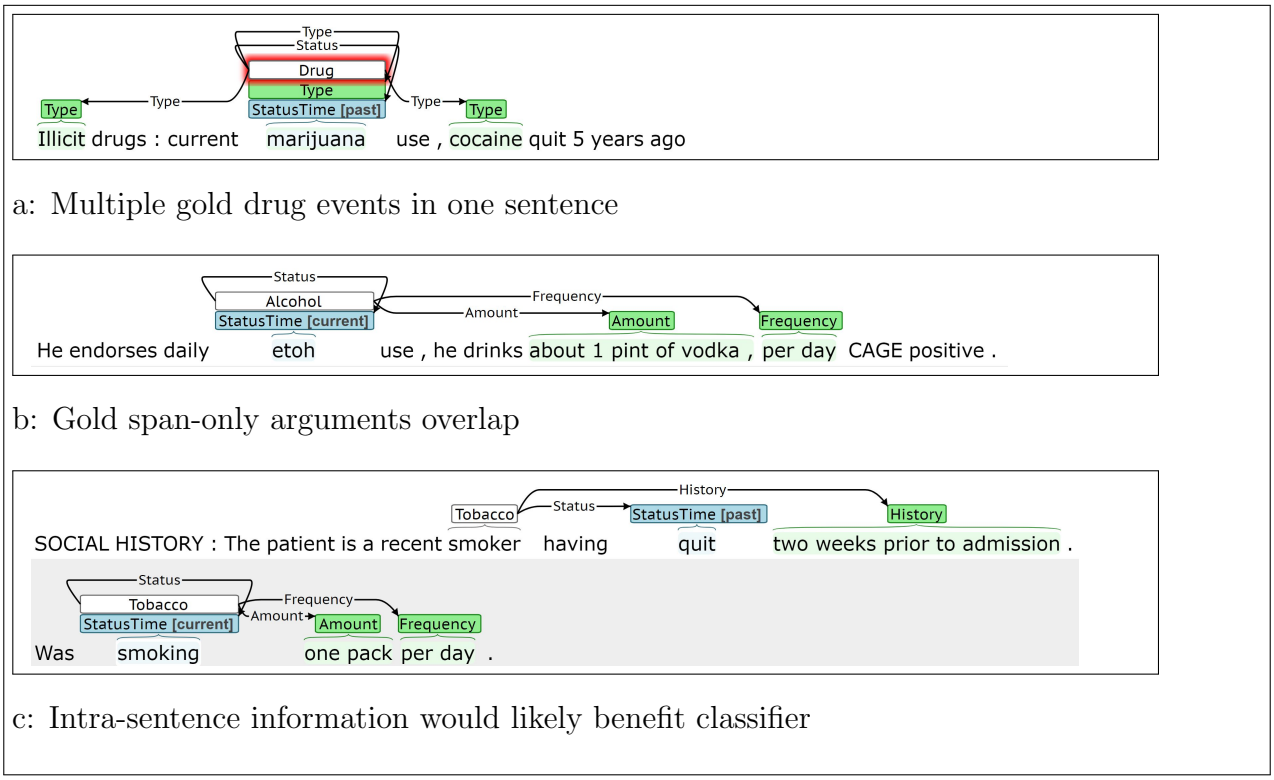

Figure 5.10: Error analysis examples

including positive substance use, unemployment, disability, and homelessness, and increased event extraction performance, relative to using only randomly selected samples. The actively selected samples improve performance in both the surrogate task and the target event extraction task, validating the surrogate task approach. The Multi-task Event Extractor model achieves high performance on the MIMIC and UW Dataset: 0.89-0.98 F1 in identifying distinct SDOH events, 0.82-0.93 F1 for substance use status, 0.81-0.86 F1 for employment status, and 0.81-0.93 F1 for living status type.

The ALSC approach is predicated on identifying the most important annotated phenomena in the annotation scheme, as this defines the Surrogate Classifier prediction task(s). For this SDOH extraction task, the information "importance" was determined based on how indicative it is of negative health outcomes. Making this determination was relatively

straightforward for this task; however, there are likely other relation or event extraction tasks where this determination is less clear. Additional experimentation with other relation and/or event extraction tasks is needed to understand the limitations of the ALSC approach and identify a generalizable approach that can be applied to a wide range of annotated phenomena.

Chapter 6

COVID-19

6.1 Overview

COVID-19 is a global pandemic, and efforts to track its spread remains a challenge for policy makers, healthcare workers, and researchers, despite increased availability of testing. Symptom information can inform COVID-19 infection tracking [14]. Certain symptoms and underlying comorbidities have directed COVID-19 testing, and this direction has changed over time as the understanding of COVID-19 and the availability of testing has changed. The clinical presentation of COVID-19 varies significantly in severity and symptom profiles [128]. The most prevalent COVID-19 symptoms reported to date are fever, cough, fatigue, and dyspnea [129], but emerging reports identify additional symptoms, including diarrhea and neurological symptoms, such as changes in taste or smell [130–132]. Certain initial symptoms may be associated with higher risk of complications; in one study, dyspnea was associated with a two-fold increased risk of Acute Respiratory Distress Syndrome [133]. The relationship between symptoms, positive tests, and rapid clinical deterioration are not well understood in ambulatory care and emergency department settings, especially in the presence of covariate risk factors, like diabetes, obesity, and heart disease.

Information collected by the Electronic Health Record (EHR) can provide crucial COVID-19 testing, diagnosis, and symptom data needed to address these knowledge gaps. Test results can easily be queried and analyzed at scale from structured EHR data. However, more detailed and nuanced descriptions of COVID-19 diagnoses, exposure history, symptoms, and clinical decision-making are typically only documented in the clinical narrative. To leverage this textual information in large-scale studies, the salient COVID-19 and symptom information must be automatically extracted.

This chapter presents the new corpus, CACT, which consists of 1,472 notes from the UW clinical repository with detailed event-based annotations for COVID-19 diagnosis, testing, and symptoms. Given the recent rapid emergence of the pandemic, CACT is one of the first clinical data sets with COVID-19 annotations and includes 29.9K distinct events. We also present the first information extraction results on CACT using the Span-based Event Extractor, establishing a strong baseline for identifying COVID-19 and symptom events.

This chapter presents our work annotating and automatically extracting COVID-19 diagnoses, testing, and symptoms, and is largely based on Lybarger et al. [22], which is currently not published. In Lybarger et al. [22], I contributed to experiment design, data analysis, and writing, performed all software development (programming), and executed all experimentation.

6.2 Materials

6.2.1 Data

This work used inpatient and outpatient clinical notes from the UW clinical repository. COVID-19-related notes were identified by searching for variations of the terms "coronavirus," "covid," "sars-cov," and "sars-2" in notes authored between February 20-March 31, 2020, resulting in a pool of 92K notes. This work utilized a subset of 53K notes, including only notes with at least five sentences and corresponding to one of six types: telephone encounters, outpatient progress, emergency department, inpatient nursing, intensive care unit, and general inpatient medicine. Multiple note types were used to improve the extraction model generalizability.

Early in the outbreak, the UW EHR did not include COVID-19 specific structured data; however, structured fields indicating COVID-19 test types and results were added as testing expanded. We used these structured fields to assign a *COVID-19 Test* label describing COVID-19 polymerase chain reaction (PCR) testing to each note based on patient test status:

- *none*: patient will be tested
- *positive*: patient will test positive
- *negative*: patient will test negative

More nuanced descriptions of COVID-19 testing (e.g. conditional or unordered tests) or diagnoses (e.g. possible infection or exposure) are not available as structured data. For the 53K note subset, the *COVID-19 Test* label distribution is 90.8% *none*, 1.3% *positive*, and 7.9% *negative*.

Given the sparsity of *positive* and *negative* notes, CACT is intentionally biased to increase the prevalence of these labels. To ensure adequate *positive* training samples, the CACT training partition includes 50% *positive* notes and 50% *none* and *negative* notes. Ideally, the test set would be representative of the true distribution; however, the expected number of *positive* labels with random selection is insufficient to evaluate extraction performance. Consequently, the CACT test partition includes 50% *positive* and *negative* notes and 50% *none* notes. Notes were randomly selected in equal proportions from the six note types.

6.2.2 Annotation Scheme

We created detailed annotation guidelines for two event types, *COVID* and *Symptom*, which are summarized in Table 6.1. Similar to the annotations in YVnotes and SHAC, CACT is annotated using an event-based approach. CACT includes two event types, *COVID* and *Symptom*, which are summarized in Table 6.1. *COVID* trigger is generally an explicit COVID-19 reference, like "COVID-19" or "coronavirus." *COVID Test Status* characterizes implicit and explicit references to testing, and *Assertion* captures diagnoses and hypothetical references to COVID-19. *Symptom* events capture subjective, often patient reported, indications of disorders and diseases (e.g. "cough"). *Symptom* trigger identifies the specific symptom, for example "wheezing" or "fever," which are characterized through *Assertion*, *Change*, *Severity*, *Anatomy*, *Characteristics*, *Duration*, and *Frequency* arguments. *Labeled arguments* (e.g. *Assertion*) include an argument span, type, and subtype (e.g. *present*). *Span-only arguments*, like *Characteristics*, include an argument span and type, without a subtype label.

Notes were annotated using the BRAT annotation tool [134].

Event type, e	**Argument type,** a	**Argument Subtypes,** y_l	**Span examples**
COVID	Trigger*	–	"COVID-19"
	Test Status†	{positive, negative, pending, conditional, not ordered, not patient, indeterminate}	"tested positive"
	Assertion†	{present, absent, possible, hypothetical, not patient}	"low suspicion"
Symptom	Trigger*	–	"cough," "fever"
	Assertion*	{present, absent, possible, conditional, hypothetical, not patient}	"admits," "denies"
	Change	{no change, worsened, improved, resolved}	"improved"
	Severity	{mild, moderate, severe}	"milde"
	Anatomy	–	"chest wall,"
	Characteristics	–	"diffuse"
	Duration	–	"for two days"
	Frequency	–	"occasional"

Table 6.1: CACT annotation guideline summary. * indicates the argument is required. † indicates at least one of the arguments, *Test Status* or *Assertion*, is required

6.2.3 Annotation Scoring and Evaluation

Annotation and extraction is scored as a slot filling task, focusing on information most relevant to secondary use applications. Figure 6.1 presents the same sentence annotated by two annotators, along with the populated slots for the *Symptom* event. Both annotations include the same trigger and *Frequency* spans ("cough" and "intermittent", respectively). The *Assertion* spans differ ("presenting with" vs. "presenting"), but the assigned subtypes (*present*) are the same, so the annotations are equivalent for purposes of populating a database. Annotator agreement and extraction performance are assessed using scoring criteria that reflects this slot filling interpretation of the labeling task.

The *Symptom* trigger span identifies the specific symptom. For *COVID*, the trigger

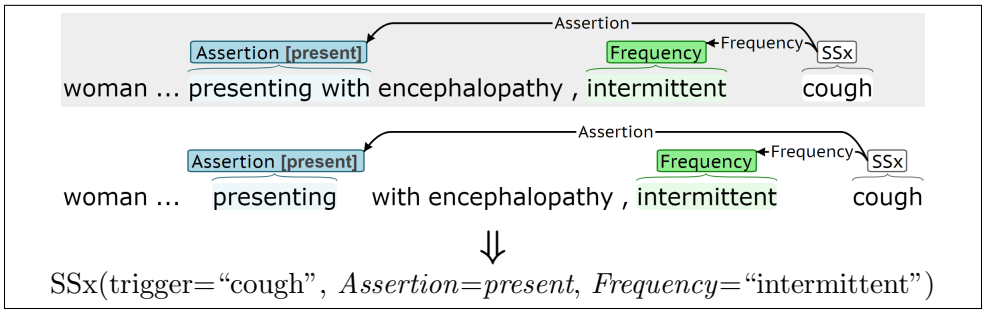

Figure 6.1: Annotation examples describing event extraction as a slot filling task

anchors the event, although the span text is not salient to downstream applications. For labeled arguments, the subtype label captures the most salient information, and the identified span is less informative. For span-only arguments, the spans are not easily mapped to a fixed label set, so the selected span contains the salient information. Performance is evaluated using precision, recall, and F1.

Trigger: Triggers, T_i, are represented by a pair (event type, e_i; token indices, x_i). Trigger equivalence is defined as

$$T_i \equiv T_j \text{ if } (e_i \equiv e_j) \wedge (x_i \equiv x_j).$$

Arguments: Events are aligned based on trigger equivalence. The arguments of events with equivalent triggers are compared using different criteria for *labeled arguments* and *span-only arguments*. Labeled arguments, A_i^l, are represented as a triple (argument type, a_i; token indices, x_i; subtype, l_i). For labeled arguments, the argument type, a, and subtype, l, capture the salient information and equivalence is defined as

$$A_i^l \equiv A_j^l \text{ if } (T_i \equiv T_j) \wedge (a_i \equiv a_j) \wedge (l_i \equiv l_j).$$

Span-only arguments, A_i^s, are represented as a pair (argument type, a_i; token indices, x_i). Arguments with equivalent triggers and argument types, $(T_i \equiv T_j) \wedge (a_i \equiv a_j)$, are compared at the token-level (rather than the span-level) to allow partial matches. Partial match scoring

is used as partial matches can still contain useful information.

6.2.4 Annotation Statistics

CACT includes 1,472 notes with a 70%/30% train/test split and 29.9K events annotated (5.4K *COVID* and 24.4K *Symptom*). Notes were annotated by four 4th-year medical students. Figure 6.2 contains a summary of the *COVID* annotation statistics for the train/test subsets. By design, the training and test sets include high rates of COVID-19 infection (*present* subtype for *Assertion* and *positive* subtype for *Test Status*), with higher rates in the training set. CACT includes high rates of *Assertion hypothetical* and *possible* subtypes. The *hypothetical* subtype applies to sentences like, "She is mildly concerned about the coronavirus" and "She cancelled nexplanon replacement due to COVID-19." The *possible* subtype applies to sentences like, "risk of Covid exposure" and "Concern for respiratory illness (including COVID-19 and influenza)." *Test Status pending* is also frequent.

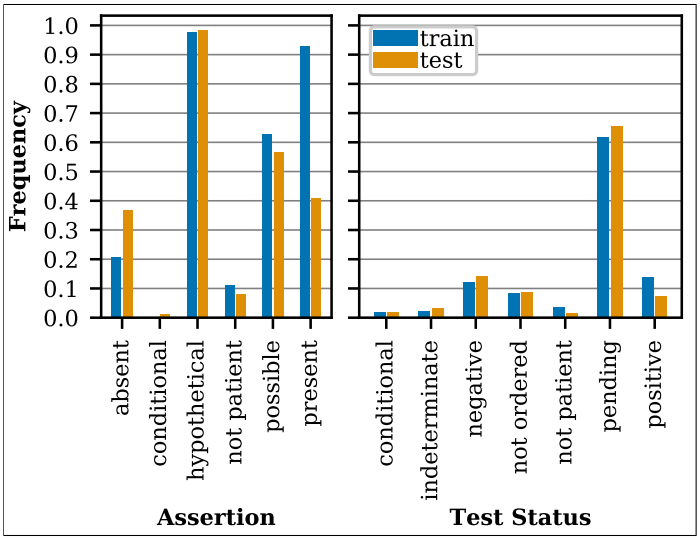

Figure 6.2: COVID annotation summary

There is some variability in the endpoints of the annotated *COVID* trigger spans (e.g. "COVID" vs. "COVID test"); however 98% of the *COVID* trigger spans in the training

set start with the tokens "COVID," "COVID19," or "coronavirus." Since the *COVID* trigger span is only used to anchor and disambiguate events, the *COVID* trigger spans were truncated to the first token of the annotated span in all experimentation and results.

The training set includes 1,756 distinct uncased *Symptom* trigger spans, 1,425 of which occur fewer than five times. The identified symptoms were not normalized to canonical forms (e.g. "shortness of breath" and "sob" considered distinct symptoms). Figure 6.3 presents the frequency of the 20 most common *Symptom* trigger spans in the training set by *Assertion* subtypes *present*, *absent*, and other (*possible*, *conditional*, *hypothetical*, or *not patient*). These 20 symptoms account for 49% of the training set *Symptom* events. There is ambiguity in delineating between some symptoms and other clinical phenomena (e.g. exam findings and medical problems), which introduces some annotation noise.

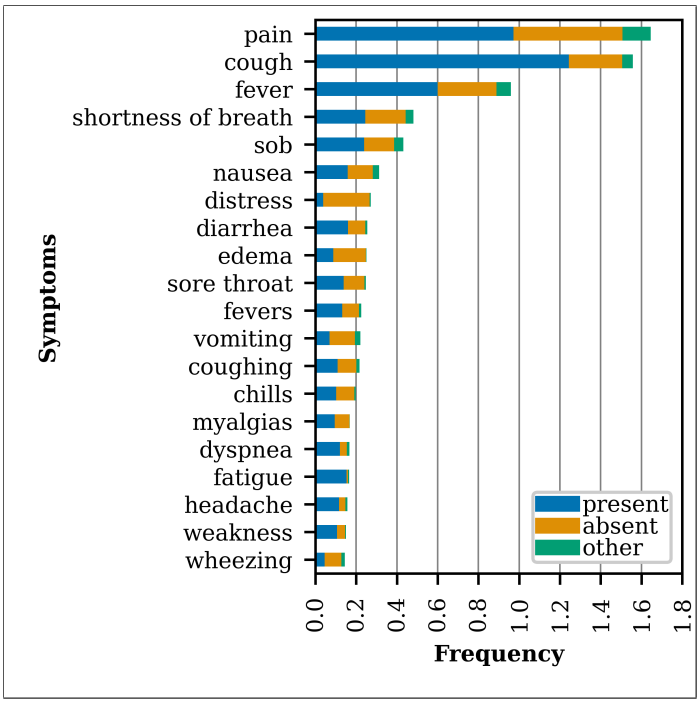

Figure 6.3: Most frequent symptoms in the training set broken down by *Assertion* subtype

Given the long tail of the symptom distribution and our desire to understand the more prominent COVID-19 symptoms, we focused annotator agreement assessment and extraction

model training/evaluation on the symptoms that occurred at least 10 times in the training set, resulting in 185 distinct symptoms that cover 82% of the training set *Symptom* events. The set of 185 symptoms was determined only using the training set, to allow unbiased experimentation on the test set. All subsequent results and experimentation only incorporate these 185 most frequent symptoms.

6.2.5 Annotator Agreement

The first two rounds of annotation were doubly annotated (72 notes in *round 1* and 96 notes in *round 2*). Figure 6.4 presents the annotator agreement for each annotation round. For labeled arguments, F1 scores are micro-average across subtypes. After *round 1*, annotator disagreements were carefully reviewed, the annotation guidelines were updated, and annotators received additional training. Starting with *round 2*, potential *COVID* triggers were pre-annotated using pattern matching ("COVID," "COVID-19," "coronavirus," etc.), to improve the recall of *COVID* annotations. Pre-annotated *COVID* triggers were modified as needed by the annotators, including removing, shifting, and adding trigger spans. The guideline updates, additional annotator training, and pre-annotation of *COVID* triggers resulted in improved agreement across all labeled phenomena, except for *Change*.

6.3 Methods

The Span-based Event Extractor used in this chapter is defined in Section 3.4. The model configuration was selected using 3-fold cross validation (CV) on the training set. Table 6.2 summarizes the selected configuration. Training loss was calculated by summing the cross entropy across all span and argument role classifiers.

During initial experimentation, *Symptom Assertion* extraction performance was high for the *absent* subtype and lower for *present*. The higher *absent* performance is primarily associated with the consistent presence of negation cues, like "denies" or "no." While there are affirming cues, like "reports" or "has," the *present* subtype is often implied by a lack of negation cues. For example, an entire sentence could be "Short of breath." To pro-

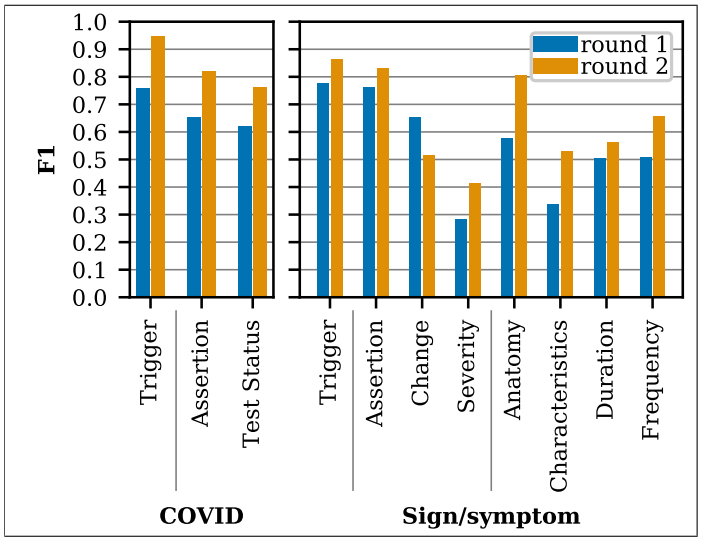

Figure 6.4: Annotator agreement

vide the *Symptom Assertion* span classifier with a more consistent span representation, we substituted the *Symptom* trigger token indices for the *Symptom Assertion* token indices in each event and found that performance improved. We extended this trigger token indices substitution approach to all labeled arguments and found performance improved. By substituting the trigger indices for the labeled argument indices, trigger and labeled argument prediction is roughly treated as a multi-label classification problem, although the model is not constrained to require trigger and labeled argument predictions to be paired with the same spans. As previously discussed, the scoring routine does not consider the span indices of labeled arguments.

6.4 Results

Table 6.3 presents the extraction performance of the Span-based Event Extractor on the training set using CV and withheld test set. Extraction performance is similar on the train and test sets, even though the training set has higher rates of COVID-19 positive notes. *COVID* trigger extraction performance is very high (0.97 F1) and exceeds the *round 2* annotator agreement (0.95 F1). The *COVID Assertion* performance (0.73 F1) is higher

Parameter	Value
Maximum sentence length, n	30
Maximum span length, M	6
Top-K spans per classifier	n
Batch size	100
Number of epochs	100
Learning rate	0.001
Optimizer	Adam
Maximum gradient L2-norm	100
BERT embedding dropout	0.3
bi-LSTM hidden size, v_h	200
bi-LSTM activation function	tanh
bi-LSTM dropout	0.3
Span classifier projection size, v_s	100
Span classifier activation function	ReLU
Span classifier dropout	0.3
Role classifier projection size, v_r	100
Role classifier activation function	ReLU
Role classifier dropout	0.3

Table 6.2: Span-based Event Extractor hyperparameters

than *Test Status* performance (0.62 F1), which is likely due to the more consistent *Assertion* annotation. *Symptom* trigger and *Assertion* extraction performance is high (0.83 F1 and 0.79 F1, respectively), approaching the *round 2* annotator agreement (0.86 F1 and 0.83 F1, respectively). *Anatomy* extraction performance (0.61 F1) is lower than expected, given the high *round 2* annotator agreement (0.81 F1). *Duration* extraction performance is comparable to annotator agreement, and *Frequency* extraction performance is lower than annotation agreement. *Change*, *Severity*, and *Characteristics* extraction performance is low, again likely related to low annotator agreement for these cases.

6.5 Conclusions

We present CACT, a novel corpus with detailed annotations for COVID-19 diagnoses, testing, and symptoms. CACT includes 1,472 unique notes across six note types with more than 500

Event type	Argument	Train-CV				Test			
		# Gold	P	R	F1	# Gold	P	R	F1
COVID	Trigger	3,931	0.95	0.97	0.96	1,497	0.96	0.97	0.97
	Assertion	2,936	0.70	0.74	0.72	1,075	0.72	0.74	0.73
	Test Status	1,068	0.60	0.62	0.61	457	0.63	0.60	0.62
Symptom	Trigger	13,823	0.82	0.85	0.83	5,789	0.81	0.85	0.83
	Assertion	13,833	0.77	0.79	0.78	5,791	0.77	0.80	0.79
	Change	739	0.45	0.03	0.06	341	0.45	0.05	0.09
	Severity	743	0.47	0.30	0.37	327	0.45	0.31	0.37
	Anatomy	3,839	0.76	0.59	0.66	1,959	0.78	0.50	0.61
	Characteristics	3,145	0.59	0.26	0.36	1,441	0.66	0.25	0.36
	Duration	3,744	0.62	0.44	0.51	1,344	0.54	0.56	0.55
	Frequency	801	0.64	0.39	0.48	250	0.60	0.51	0.55

Table 6.3: Extraction performance

notes from patients with positive COVID-19 tests. We introduce the Span-based Event Extractor that jointly extracts all annotated phenomena, including argument types and subtypes. The Span-based Event Extractor performs well in the extraction of *COVID* trigger (0.97 F1) and *Assertion* (0.73 F1) and achieves near-human performance in the extraction of *Symptom* trigger (0.83 F1) and *Assertion* (0.79 F1).

While CACT includes six different note types, it only includes data from a single institution over a relatively short period from the COVID-19 pandemic, which may reduce the generalizability of extractors trained on CACT. The COVID-19 documentation protocols at UW changed over the course of the pandemic, including the use of templated symptom information (e.g. "[x] fever [ ] cough"), and the performance of the trained extractors may vary, depending on the timing of the notes. Additionally, each institution has its own documentation protocols, including protocols related to COVID-19, and the generalizability of extraction models trained on CACT to other institutions is unknown.

Chapter 7

SECONDARY USE WITH AUTOMATIC LABELS

7.1 Overview

This chapter explores a secondary use application for the automatically extracted information. Specifically, the relationship between COVID-19 infection (*positive* or *negative*) and automatically extracted symptom information is explored through a COVID-19 infection prediction task.

7.2 Related Work

There are many pre-print and published works exploring the prediction of COVID-19 outcomes, including COVID-19 infection, hospitalization, acute respiratory distress syndrome (ARDS), need for intensive care unit (ICU), need for a ventilator, and mortality [135–141]. These COVID-19 outcomes are generally predicted using existing structured data within the EHR, including demographics, diagnosis codes, vitals, and lab results.

Wollenstein-Betech et al. [135] used an existing clinical data set from Mexico to explore the prediction of COVID-19 related hospitalization, ICU need, ventilator need, and mortality (pre-print publication). The outcomes are predicted using logistic regression and SVM. Wollenstein-Betech et al. [135] identifies the most salient preconditions for the COVID-19 outcomes as: i) age, gender, chronic renal insufficiency, diabetes, and immunosuppression for hospitalization; ii) pneumonia, cardiovascular disease, and asthma for ICU need; iii) pneumonia, age, gender, cardiovascular disease, obesity, and pregnancy for ventilator need; and iv) age, immunosuppression and pregnancy for mortality.

Bertsimas et al. [136] explore the prediction of COVID-19 infection risk and mortality using data from healthcare institutions in Spain and Italy (pre-print publication). Out-

comes are predicted using gradient boosted decision trees (XGBoost specifically) with demographics, vitals, and lab values as input features. The results indicate the main indicators of mortality are age, Blood Urea Nitrogen (BUN), C- reactive protein (CRP), Aspartate Aminotransferase (AST), and low oxygen saturation. The primary indicators of COVID-19 infection risk are CRP, white blood cell count (WBC), Calcium, AST, and temperature. In the absence of lab results, the indicators of infection risk are age, oxygen saturation, temperature, and heart rate. Bertsimas et al. [136] find that demographics and vitals are predictive of COVID-19 infection when lab values are not available; however demographics and vitals become secondary features when lab values are available.

Izquierdo et al. [137] predict ICU admission using a data set with 10.5K COVID-19 infections, 1.4K hospitalizations, and 83 ICU admissions (pre-print publication). ICU admission is predicted using decision trees using structured EHR data and information automatically extracted from clinical notes. Information was extracted from free-text notes using the existing EHRead tool. A comprehensive list of the extracted information is not provided, although common diseases (e.g. diabetes) are mentioned. Izquierdo et al. [137] found age, temperature, and respiratory frequency are the best predictors of ICU admission.

There are multiple review papers focused on synthesizing predictors of COVID-19 infection and severity. Wynants et al. [142] finds the primary predictors of COVID-19 infection to be age, body temperature, signs and symptoms, sex, blood pressure and creatinine. Wynants et al. [142] also finds the primary predictors of hospitalization to be age, sex, previous hospitalization, comorbidities, and SDOH. In a review paper exploring the epidemiology and clinical features of COVID-19, Siordia [143] identifies elderly age, hypertension, cardiovascular disease, cerebrovascular disease, and chronic kidney disease as salient comorbidities and elevated lactic acid dehydrogenase (LDH), elevated CRP, and lymphopenia as laboratory predictors for severe COVID-19. In a review paper focused laboratory predictors of COVID-19 severity and outcomes, Zhang et al. [144] identified elevated leukocyte count, ALT, AST, LDH, and procalcitonin as risk factors for ICU admission.

Table 7.1 summarizes the laboratory, vital sign, and demographic structured fields found

to be most predictive of COVID-19 infection in the literature described above. While there are some frequently cited laboratory results, like CRP and lymphyocytes, there does not appear to be a consensus across the literature, regarding the most prominent predictors of COVID-19 infection. Of the four sources cited in Table 7.1, three sources [142–144] are review papers, so this table represents the distillation of many recent works. The predictive parameters in Table 7.1 informed the development of the COVID-19 infection prediction model in the subsequent section.

Table 7.1: Prominent parameters available through structured EHR fields that are predictive of COVID-19 infection.

Parameter	**Source**
age	[136, 142]
alanine aminotransferase (ALT)	[143]
albumin	[144]
aspartate aminotransferase (AST)	[136, 143]
calcium	[136]
C-reactive protein (CRP)	[136, 143, 144]
D-dimer	[143]
eosinopenia	[143]
heart rate	[136]
lactate dehydrogenase (LDH)	[143, 144]
lymphocyte	[142–144]
neutrophils	[142, 143]
oxygen saturation	[136]
prothrombin time (PT)	[143]
respiratory rate	[136]
temperature	[136, 142]
troponin	[143]
white blood cell (WBC) count	[136]

7.3 Methods

An existing data set from the UW from January 2020 through May 2020 was used to explore the prediction of COVID-19 infection and to identify the most predictive features. The data

set is from the UW Enterprise Data Warehouse, which is a subset of the UW EHR. This data set has some overlap with the data set used in Chapter 6 but is treated as a separate data set in this COVID-19 infection prediction task.

The data set includes clinical notes (telephone encounters, progress notes, and emergency department (ED) notes) and structured data (demographics, vitals, laboratory results, diagnosis codes, etc.). For each patient in this data set, all of the COVID-19 tests with either a *positive* or *negative* result and an ED note within the seven days preceding the test result were identified, resulting in a set of 1,580 negative tests and 115 positive tests (6.8% infection rate). COVID-19 tests with an indeterminate result were not included in experimentation, as the indeterminate result is a function of testing procedures rather than patient factors. Additionally, COVID-19 tests without an ED note within the past seven days were not included in experimentation, to provide a fair comparison for the predictive power of the structured and automatically extracted data. Each of these test results was treated as a sample in model training and evaluation. The likelihood of COVID-19 infection was predicted using structured EHR data and ED notes within a seven-day window preceding the COVID-19 test result.

Symptom information was automatically extracted from the ED notes using the Span-based Event Extractor trained on CACT [1]. The extracted symptoms were manually normalized to aggregate different extracted spans with similar meanings (e.g. "sob"$\rightarrow$ "shortness of breath" or "fatigued" $\rightarrow$ "fatigue"). As an input feature, each extracted symptom was assigned a numerical value, based on the *Assertion* value: $present = +1$ and $absent = -1$. Symptoms not referenced in the note were assigned a value of 0.

For each of the structured fields in Table 7.1 the corresponding UW EHR field was identified. Experimentation with structured fields was limited to this subset of literature-supported COVID-19 predictors, given the limited size of the available data set. Table 7.2 lists the 32 structured fields used in the COVID-19 prediction task. The following EHR

[1]Only automatically extracted symptom data were used. No supervised (hand annotated) labels were used.

field pairs measure the same phenomena and were treated as a single feature, resulting in 29 distinct structured EHR fields: {"Temperature - C," "Temperature (C)"}, {"HR," "Heart Rate"}, and {"O2 Saturation (%)," "Oxygen Saturation"}. All structured fields used in experimentation are numerical (e.g. "Temperature (C)"=38.1) with the exception of "Troponin I Interpretation," which is categorical (abnormal or normal).

Within the seven-day history window, features may occur multiple times (e.g. multiple temperature measurements). For each feature, the series of measurements/values was represented as either the minimum or maximum of the values depending on the specific feature. For example, temperature was represented as the maximum of the temperature measurements to detect any fever, and oxygen saturation was represented as the minimum of the saturation values to capture any low oxygenation events. Table 7.2 includes the aggregating function, $\boldsymbol{f}$, used for each structured field. For all extracted symptoms, the maximum feature value was used, to capture any present symptoms within the time window.

COVID-19 infection was predicted using the Random Forest model using the scikit-learn Python implementation [116]. Alternative prediction algorithms include Logistic Regression, SVM, or FFNN. Random Forest was selected over Logistic Regression, because the feature independence and linear assumptions associated with Logistic Regression are not valid this prediction task. Random Forest was selected over SVM, because the Random Forest model provides improved interpretability. A FFNN was not used because of the small amount of data available for this study.

The available data was split into train/test sets using an 80%/20% split by patient. Performance was evaluated using the receiver operating characteristic (ROC) area under the curve (AUC). Given the relatively small data set size, the train/test splits were randomly created 1,000 times through repeated hold-out testing [145]. Kim [145] demonstrated that repeated hold-out testing can improve the robustness of the results in low resource settings. For each training/test split, the ROC was calculated, and an averaged ROC was calculated across all random hold-out iterations. The random holdout iterations yield a distribution of ROC and AUC values, which facilitate significance testing. The significance of the AUC

Table 7.2: Structured fields from UW EHR used to predict COVID-19 infection. ***f*** indicates the with function used to aggregate multiple measurements/values

Parameter	**Field in UW EHR**	***f***
age	"AgeIn2020"	max
ALT	"ALT (GPT)"	max
albumin	"Albumin," "Albumin/Globulin Ratio," "Albumin (Micro), URN", and "Albumin/Creatinine Ratio, URN"	max
AST	"AST (GOT)"	max
calcium	"Calcium"	max
CRP	"CRP, high sensitivity"	max
D-dimer	"D_Dimer Quant"	max
eosinopenia	"Eosinophils," "% Eosinophils," and "Body Fluid Eosinophils"	max
heart rate	"Heart Rate" and "HR"	max
LDH	"Lactate Dehydrogenase"	max
lymphocyte	"Lymphocytes," "% Lymphocytes," and "Body Fluid Lymphocytes"	max
neutrophils	"Neutrophils," "% Neutrophils," and "Body Fluid Neutrophils"	max
oxygen saturation	"Oxygen Saturation" and "O2 Saturation (%)"	min
PT	"Prothrombin Time Patient" and "Prothrombin INR"	max
respiratory rate	"Respiratory Rate"	max
temperature	"Temperature - C" and "Temperature (C)"	max
troponin	"Troponin_I" and "Troponin_I Interpretation"	max
WBC count	"WBC"	max

performance was assessed using a two-sided T-test. The Random Forest models were tuned using 3-fold cross validation on the training set and evaluated on the withheld test set. COVID-19 prediction experimentation included three feature sets: *structured* (29 structured EHR fields), *notes* (automatically extracted symptoms from ED notes), and *structured+notes* (combination of structured fields and automatically extracted symptoms).

The relative importance of features in predicting COVID-19 infection was explored using

Lundberg et al. [146]'s SHAP (SHapley Additive exPlanations) approach, which is implemented in the SHAP Python module.[2] SHAP generates interpretable, feature-level explanations for nonlinear model predictions. For each prediction, SHAP scores are estimated for each feature, where larger absolute scores indicate more important features, and the absolute values of the scores sum to 1.0 for each prediction.

7.4 Results

Figure 7.1 presents the ROC for the COVID-19 infection prediction task (*positive* or *negative*), with three feature sets: *structured*, *notes*, *structured+notes*. This ROC curve is averaged across the 1,000 repeated hold-out partitions. On the withheld test set, the inclusion of the automatically extracted symptom information (*structured+notes*) improves the average AUC over structured data only (*structured*) from 0.69 to 0.73 with significance ($p < 0.001$). The shaded region around each mean ROC indicates one standard deviation.

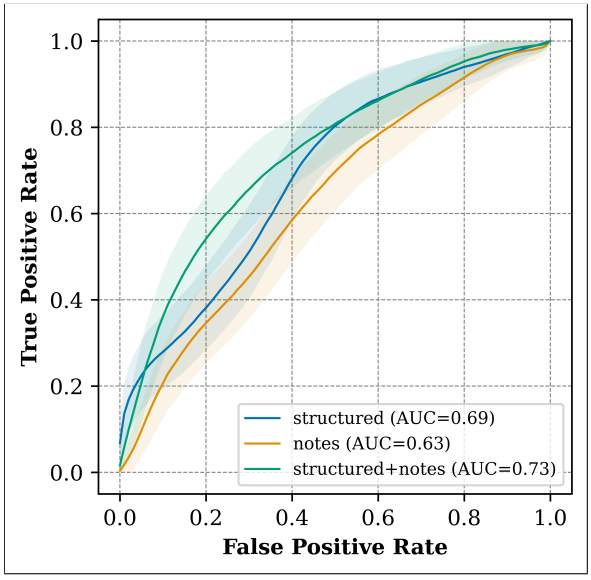

Figure 7.1: COVID-19 infection prediction ROC on the withheld test set, averaged across repeated hold-out runs

[2] `https://pypi.org/project/shap/`

Figure 7.2 is a SHAP value plot for the 20 most predictive features from a single Random Forest model utilizing the *structured+notes* feature set. In this SHAP plot, each point represents a single prediction for the test set, and the SHAP value (x-axis) describes the importance of that feature in making each prediction. Positive SHAP values indicate support for a positive COVID-19 test result, and negative SHAP values indicate support for negative test result. The most predictive features include four automatically extracted symptoms: fever, pain, cough, and myalgia (muscle pain/ache).

For some features, like "Calcium" and "fever," the feature values and COVID-19 positivity are highly correlated (i.e. feature values correlated with SHAP values), which is indicated by a clear separation of the feature values (i.e. blue and red data points well separated). However, for some features, like "Albumin," the correlation is lower, which is indicated by a mixing of the feature values (i.e. blue and red data points mixed). Features with lower correlation but higher predictive power suggests the features are predictive when combined with other features (i.e. feature interdependence). Additionally, there are features that characterize the same phenomena, just with different criteria, for example {"fever", "Temperature (C)"} and {"Eosinophils", "% Eosinophils"}.

Given the relatively small sample size and low proportion of positive COVID-19 tests, the SHAP impact values presented in Figure 7.2 were aggregated across repeated hold-out runs. Figure 7.3 presents the distribution of the mean SHAP values for the *structured+notes* feature set. For each repeated hold-out run, the absolute value of the SHAP values were averaged, yielding a single feature score per repetition. In Figure 7.3, the mean SHAP values (x-axis) represents the importance of the feature with predicting COVID-19 infection (positive or negative). Only features with a mean absolute correlation greater than or equal to 0.5 are included in Figure 7.3, to emphasize features that are predictive with less reliance on contributions from other features (i.e. more independent features).

The most predictive features are temperature/fever, WBC, cough, calcium, eosinophils, age, respiratory rate, and oxygen saturation. Fever, cough, pain, myalgia, and fatigue are the most predictive automatically extracted symptoms. These results suggest that patient

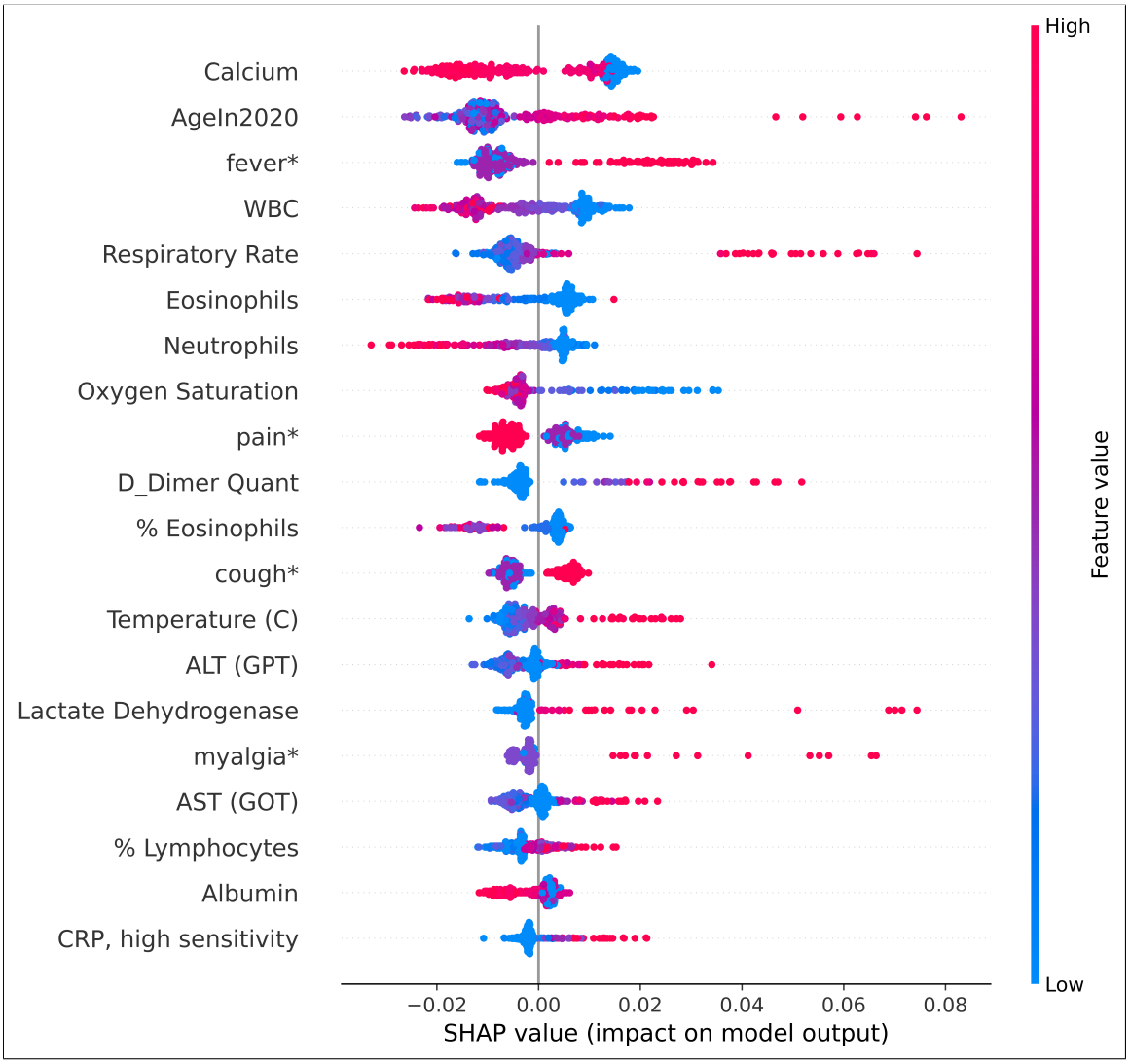

Figure 7.2: SHAP plot for Random Forest model utilizing the *labs+vitals+notes* feature set, explaining the importance of features in making predictions for the withheld test set. * indicates the feature is an automatically extracted symptom

reported fever is more predictive of COVID-19 infection than the raw temperature measurements (i.e. "Temperature (C)"), which did not meet the correlation threshold of 0.5.

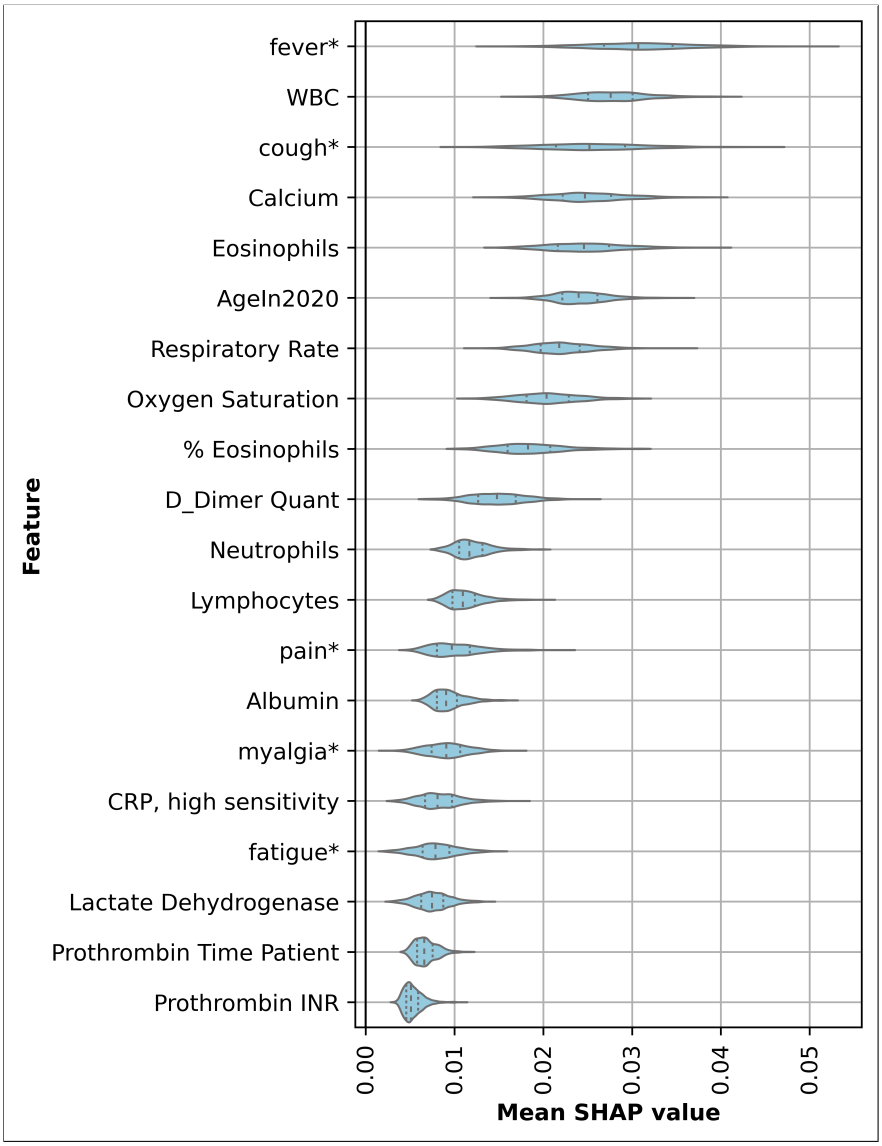

Figure 7.3: Distribution of averaged SHAP values. The vertical lines in each violin indicate the quartiles. * indicates the feature is an automatically extracted symptom

7.5 Conclusions

In the COVID-19 infection prediction task, automatically extracted symptom information improved extraction performance (with significance) beyond just using structured data, and two of the top three most predictive features, fever and cough, were automatically extracted

from the ED notes. This secondary use application is limited by the size and scope of the available data. In ongoing COVID-19 work, the Span-based Event Extractor trained on CACT will be applied to a broader set of over 2 million ambulatory care and emergency department notes created at UW during the first five months of the pandemic for a more comprehensive analysis of reported symptoms, patient characteristics, and outcomes.

Chapter 8

CONCLUSIONS

This chapter summarizes the primary contributions of this work, including the presented methodologies and empirical findings. It also presents future directions for continuing this research and plans for maximizing the impact of the resources created.

8.1 Summary

This dissertation explores the extraction of information from clinical text, to facilitate the use of this text-encoded information in secondary use applications. Specifically, it explores the automatic extraction of SDOH and COVID-19 information and demonstrates the utility of the extracted data. The main contributions of this work are described below.

Neural architectures for clinical IE with limited training data: The limited availability of annotated clinical text, due to privacy concerns and the cost of human annotation, is one of the primary challenges associated with clinical IE. The proposed IE methodologies address this challenge, by introducing multi-task modeling frameworks that leverage shared information across event and argument types. The progression of the introduced IE architectures reflects the evolution of this work and field of NLP, as the generalizability, flexibility, and modeling power improves in the successive models. Multi-task Event Extractor incorporates contextualized word embeddings and utilizes a more generalized set of output layers, relative to the Multi-task Substance Extractor. The Span-based Event Extractor overcomes two primary limitations of the earlier approaches, specifically the ability to accommodate multiple events of the same type within a sentence and model overlapping spans. The Multi-task Substance Extractor achieved state-of-the-art performance on YVnotes. The Multi-task Event Extractor and Span-based Event Extractor achieved extraction performance comparable to

inter-annotator agreement for several key phenomena in the respective corpora on which they were trained.

The self-attention trigger prediction approach of the Multi-task Event Extractor can only generate a single prediction for each event type within a sentence, and this approach performs well on SHAC, because co-occurring events of the same type are infrequent. In SHAC, only 6% of sentences with at least one event include multiple events of the same type. The Multi-task Event Extractor was not applied to CACT, because of the high prevalence of co-occurring symptoms within sentences. In CACT, 35% of sentences with at least one symptom include multiple symptoms, and these sentences with co-occurring symptoms account for 61% of all annotated symptoms. Extracting the symptom information in CACT requires an extraction approach that can represent multiple triggers within a single sentence, which is accomplished by the Span-based Event Extractor.

The extraction models are tailored to specific annotation/extraction schemes, to maximize extraction performance, and are only evaluated on a single data set. The presented IE frameworks can be extended to a range of information represented in the clinical narrative; however, additional experimentation is required to assess the performance of the architectures in other event extraction tasks.

Annotated corpora: This work presents two new annotated corpora, SHAC and CACT, which include detailed event-based annotations for SDOH and COVID-19, respectively. Both corpora frequently contain the risk-factors and conditions of greatest interest. SHAC contains rich descriptions of SDOH, including substance use, employment, and living situation. CACT includes detailed annotations of COVID-19 testing, diagnoses, and symptoms, including high proportions of notes associated with COVID-19 positive patients.

SHAC includes multiple note types from three institutions, and CACT includes multiple note types from a single institution. While we attempted to make these corpora as heterogeneous as possible, SHAC and CACT are limited by the publicly and internally available data sets used. The degree to which the content, structure, and format of the SHAC and CACT notes is representative of other institutions is unknown. The portions of the notes

that use the most natural language (e.g. "Patient reports cough and fever but denies shortness of breath") will likely generalize better than more structured/templated content (e.g. "[x] cough [x] fever [ ] shortness of breath"). Additional annotation and experimentation is needed to better understand the generalizability of the corpora created.

Active learning using Surrogate Classifiers: A novel active learning framework, ALSC, is introduced and used to select samples for SHAC. The presented surrogate classifier approach increased the prevalence of salient health risk factors. The actively selected notes improved extraction performance beyond that of random selection, with the biggest performance improvements associated with underrepresented but extremely important health risk factors (i.e. drug and tobacco use, homelessness, and living with others).

The performance of the ALSC approach was only evaluated on a single domain, and additional experimentation is needed to assess the generalizability of this approach and better understand its limitations. In the SDOH extraction task, it is relatively clear which arguments (attributes) are the most salient, which motivated the surrogate classifier approach. However, identifying the most salient information within other clinical IE tasks may not be as clear or even possible, which is limits the generalizability of the ALSC approach. The ALSC approach improved event extraction performance for the specific extraction architecture and scoring rubric used, and different extraction architectures and/or scoring rubrics may impact the performance gains achieved using ALSC.

Secondary use with automatic labels: The usefulness and importance of the automatically extracted data is demonstrated through secondary use applications. COVID-19 infection prediction performance improves with the incorporation of automatically extracted symptom data, and the results indicate the most predictive symptoms are fever, cough, pain, myalgia, and fatigue.

The COVID-19 infection prediction task used a relatively small data set, specifically a relatively small number of positive COVID-19 test results. A larger multi-institution data set is needed to draw broader conclusions regarding the most prominent risk factors for COVID-19 positivity, including things like symptoms, laboratory results, and vital signs.

8.2 Future Work

8.2.1 IE

There are several promising avenues for building on the presented IE models. The extracted information from labeled arguments (e.g. *Tobacco Status*) can easily be represented as a one-hot encoding in downstream prediction tasks. However, incorporating the the extracted information from span-only arguments (e.g. *Tobacco Amount*, *Frequency*, and *Duration*) is more challenging, as the extracted span(s) must first be normalized to a fixed set of classes (e.g. "mild," "moderate," or "severe"), a scalar representation (e.g. score of 2 out of 3), or a distributional representation (e.g. vector of real numbers). A data-driven approach for learning a mapping or transformation to these normalized quantities would improve the usefulness of the extracted arguments.

The presented IE approaches only uses the information contained in the clinical notes. The IE approaches do not leverage non-text data in the EHR or data from biomedical knowledge sources. It is not clear that utilizing such data sources would assist in the extraction of SDOH or symptom information; however, it may be beneficial in other clinical information extraction tasks. If the present IE architectures are utilized in other clinical IE tasks, like medical problem or diagnosis extraction, augmenting the frameworks to incorporate structured sources may improve performance.

All of the presented approaches utilize sequential word information; however, incorporating dependency information may improve extraction performance, especially where multiple events are in close proximity or overlap. Work in other domains and to a lesser degree the clinical domain has benefited from such approaches [61, 147, 148]. However, such an approach would require a high performing dependency parser that is trained on clinical text.

8.2.2 Active Learning

The presented active learning framework, ALSC, was successful because the surrogate classifier prediction task captured salient aspects of the more complex event extraction task. In

other words, the gains achieved by the surrogate classifier resulted in gains for the event extractor. In future work, the surrogate classifier approach could be explored in different clinical IE tasks to identify and demonstrate a generalizable approach for representing complex annotation structures (e.g. events or relations) through a simplified proxy task in active learning.

8.2.3 Shared Task

SHAC includes clinical text from MIMIC-III, which is deidentified and publicly available. We are planning to create a shared IE task using this deidentified portion of SHAC. The shared task will increase the visibility of this work and advance the exploration of SDOH extraction. The MIMIC-III portion of SHAC is relatively large and will facilitate experimentation with more contemporary neural extraction approaches. Additionally, the detailed event-based structure of the SHAC annotations will facilitate a range of possible challenge tasks (e.g. relation extraction, event extraction).

8.2.4 Secondary Use

The uncertainty associated with the automatically extracted data varies, depending on the frequency and similarity of the target phenomena within the training set. The COVID-19 infection prediction task explored in this work incorporated the automatically extracted symptom data, without utilizing the uncertainty of the predictions. Incorporating prediction uncertainty in secondary use applications may provide performance improvements, by emphasizing more confident predictions and deemphasizing less confident predictions.

Using its Enterprise Data Warehouse (EDW), UW Medicine applies a range of tools to large repositories of clinical data, including clinical notes, to enable large-scale outcomes analyses. The SDOH and COVID-19 extractors created using SHAC and CACT, respectively, are integrated into the EDW analysis pipeline to automatically populate databases with the extracted information. The integration of the SDOH and COVID-19 extractors into the EDW pipeline will allow UW investigators to explore negative health outcomes associated

with SDOH and symptoms. Additionally, we are working to release the pre-trained extraction models to collaborators outside the UW.

As part of the integration of the extractors into the EDW pipeline, the SDOH and COVID-19 extractors will be applied to a much larger set of clinical ambulatory care and emergency department notes from UW and collaborating institutions nationally. The extracted symptom information will also be combined with routinely coded data (e.g. diagnosis and procedure codes, demographics) and automatically extracted data (e.g. social determinants of health). Using these data, we will develop models for predicting risk of COVID-19 infection amongst individuals who are tested. These models could better inform clinical indications for prioritizing testing with constrained test availability and more accurately determine pre-test probability. Additionally, the presence or absence of certain symptoms can be used to inform clinical care decisions with greater precision. This future work may also identify combinations of symptoms (including their presence, absence, severity, sequence of appearance, duration, etc.) associated with clinical outcomes and health service utilization, such as deteriorating clinical course and need for repeat consultation or hospital admission. The use of detailed symptom information will be highly valuable in informing these models, but potentially only with the level of nuance that our extraction models provide. With the COVID-19 pandemic continuing for the foreseeable future, accelerating the research outlined in this paper will inform key clinical and health service decision making.

BIBLIOGRAPHY

[1] Peter B Jensen, Lars J Jensen, and Søren Brunak. Mining electronic health records: towards better research applications and clinical care. *Nature Reviews Genetics*, 13(6): 395, 2012. doi:10.1038/nrg3208.

[2] Stéphane M Meystre, Guergana K Savova, Karin C Kipper-Schuler, John F Hurdle, et al. Extracting information from textual documents in the electronic health record: a review of recent research. *Yearbook of Medical Informatics*, 35(8):128–144, 2008. doi:10.1055/s-0038-1638592.

[3] Dina Demner-Fushman, Wendy W Chapman, and Clement J McDonald. What can natural language processing do for clinical decision support? *Journal of Biomedical Informatics*, 42(5):760–772, 2009. doi:10.1016/j.jbi.2009.08.007.

[4] Centers for Medicare & Medicaid Services. The Health Insurance Portability and Accountability Act of 1996 (HIPAA), 1996. URL `http://www.cms.hhs.gov/hipaa/`.

[5] Lucy Gilson, Jane Doherty, Rene Loewenson, and Victoria Francis. Challenging inequity through health systems. final report knowledge network. *WHO commission on the social determinants of health*, 2007. URL `https://www.who.int/social_determinants/resources/csdh_media/hskn_final_2007_en.pdf`.

[6] Centers for Disease Control and Prevention. Social determinants of health: Know what affects health, 2018. URL `https://www.cdc.gov/socialdeterminants/index.htm`.

[7] Office of Disease Prevention and Health Promotion. Social determinants, 2020. URL `https://www.healthypeople.gov/2020/leading-health-indicators/2020-lhi-topics/Social-Determinants`.

[8] Centers for Disease Control and Prevention. Annual smoking-attributable mortality, years of potential life lost, and productivity losses–united states, 1997-2001. *Morbidity and Mortality Weekly Report*, 54(25):625, 2005. doi:10.1001/jama.294.7.788.

[9] World Health Organization. Global status report on alcohol and health 2018. *World Health Organization*, 2019. URL `https://www.who.int/substance_abuse/publications/global_alcohol_report/gsr_2018/en/`.

[10] Louisa Degenhardt and Wayne Hall. Extent of illicit drug use and dependence, and their contribution to the global burden of disease. *The Lancelot*, 379(9810):55–70, 2012. doi:10.1016/S0140-6736(11)61138-0.

[11] John T Cacioppo and Louise C Hawkley. Social isolation and health, with an emphasis on underlying mechanisms. *Perspectives in Biology and Medicine*, 46(3):S39–S52, 2003. doi:10.1353/pbm.2003.0063.

[12] Katherine D Blizinsky and Vence L Bonham. Leveraging the learning health care model to improve equity in the age of genomic medicine. *Learning Health Systems*, 2 (1):e10046, 2018. doi:10.1002/lrh2.10046.

[13] World Health Organization. Coronavirus disease (covid-19) situation report – 205, 2020. URL `https://www.who.int/emergencies/diseases/novel-coronavirus-2019/situation-reports`.

[14] Hagai Rossman, Ayya Keshet, Smadar Shilo, Amir Gavrieli, Tal Bauman, Ori Cohen, Esti Shelly, Ran Balicer, Benjamin Geiger, Yuval Dor, et al. A framework for identifying regional outbreak and spread of covid-19 from one-minute population-wide surveys. *Nature Medicine*, pages 1–4, 2020. doi:https://doi.org/10.1038/s41591-020-0857-9.

[15] Joann G Elmore, Pin-Chieh Wang, Kathleen F Kerr, David L Schriger, Douglas E Morrison, Ron Brookmeyer, Michael A Pfeffer, Thomas H Payne, and Judith S Currier. Excess patient visits for cough and pulmonary disease at a large us health system in

the months prior to the covid-19 pandemic: A time-series analysis. *Journal of Medical Internet Research*. doi:10.2196/21562.

[16] Elissa M Abrams and Stanley J Szefler. COVID-19 and the impact of social determinants of health. *The Lancet Respiratory Medicine*, 2020. doi:10.1016/S2213-2600(20)30234-4.

[17] Jack Tsai and Michal Wilson. COVID-19: a potential public health problem for homeless populations. *The Lancet Public Health*, 5(4):e186–e187, 2020. doi:10.1016/S2468-2667(20)30053-0.

[18] Constantine I Vardavas and Katerina Nikitara. Covid-19 and smoking: A systematic review of the evidence. *Tobacco induced diseases*, 18, 2020. doi:10.18332/tid/119324.

[19] Meliha Yetisgen and Lucy Vanderwende. Automatic identification of substance abuse from social history in clinical text. *Artificial Intelligence in Medicine*, pages 171–181, 2017. doi:10.1007/978-3-319-59758-4_18.

[20] Kevin Lybarger, Meliha Yetisgen, and Mari Ostendorf. Using neural multi-task learning to extract substance abuse information from clinical notes. In *AMIA Annual Symposium Proceedings*, pages 1395–1404, 2018. URL `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6371261`.

[21] Kevin Lybarger, Mari Ostendorf, and Meliha Yetisgen. Annotating social determinants of health using active learning, and characterizing determinants using neural event extraction. *arXiv*, 2020. URL `https://arxiv.org/abs/2004.05438`.

[22] Kevin Lybarger, Mari Ostendorf, Matthew Thompson, and Meliha Yetisgen. Extracting covid-19 diagnoses and symptoms from clinical text: A new annotated corpus and neural event extraction framework. In *Open Review Preprint*, 2020. URL `https://openreview.net/forum?id=-o9Z32xu9Se`.

[23] Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. MIMIC-III, a freely accessible critical care database. *Scientific Data*, 3:160035, 2016. doi:10.1038/sdata.2016.35.

[24] Amber Stubbs and Özlem Uzuner. Annotating longitudinal clinical narratives for de-identification: The 2014 i2b2/uthealth corpus. *Journal of Biomedical Informatics*, 58: S20–S29, 2015. doi:10.1016/j.jbi.2015.07.020.

[25] Özlem Uzuner, Brett R South, Shuying Shen, and Scott L DuVall. 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association*, 18(5):552–556, 2011. doi:10.1136/amiajnl-2011-000203.

[26] Long Chen, Yu Gu, Xin Ji, Zhiyong Sun, Haodan Li, Yuan Gao, and Yang Huang. Extracting medications and associated adverse drug events using a natural language processing system combining knowledge base and deep learning. *Journal of the American Medical Informatics Association*, 27(1):56–64, 10 2019. doi:10.1093/jamia/ocz141.

[27] Fenia Christopoulou, Thy Thy Tran, Sunil Kumar Sahu, Makoto Miwa, and Sophia Ananiadou. Adverse drug events and medication relation extraction in electronic health records with ensemble deep learning methods. *Journal of the American Medical Informatics Association*, 27(1):39–46, 2020. doi:10.1093/jamia/ocz101.

[28] Yanshan Wang, Liwei Wang, Majid Rastegar-Mojarad, Sungrim Moon, Feichen Shen, Naveed Afzal, Sijia Liu, Yuqun Zeng, Saeed Mehrabi, Sunghwan Sohn, et al. Clinical information extraction applications: a literature review. *Journal of Biomedical Informatics*, 77:34–49, 2018. doi:10.1016/j.jbi.2017.11.011.

[29] Laura Chiticariu, Yunyao Li, and Frederick R. Reiss. Rule-based information extraction is dead! long live rule-based information extraction systems! In *Empirical Methods in*

Natural Language Processing, pages 827–832, 2013. URL https://www.aclweb.org/anthology/D13-1079.

[30] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence Magazine*, 2018. doi:10.1109/MCI.2018.2840738.

[31] Stephen Wu, Kirk Roberts, Surabhi Datta, Jingcheng Du, Zongcheng Ji, Yuqi Si, Sarvesh Soni, Qiong Wang, Qiang Wei, Yang Xiang, et al. Deep learning in clinical natural language processing: a methodical review. *Journal of the American Medical Informatics Association*, 27(3):457–470, 2020. doi:10.1093/jamia/ocz200.

[32] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *International Conference on Learned Representations*, 2013. URL https://arxiv.org/abs/1301.3781.

[33] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543, 2014. doi:10.3115/v1/D14-1162.

[34] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *North American Chapter of the Association for Computational Linguistics*, pages 2227–2237, 2018. doi:10.18653/v1/N18-1202.

[35] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*, pages 4171–4186, 2019. doi:10.18653/v1/N19-1423.

[36] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. XLNet: Generalized autoregressive pretraining for lan-

guage understanding. In *Advances in Neural Information Processing Systems*, pages 5753–5763, 2019. URL http://papers.nips.cc/paper/8812-xlnet-generalized-autoregressive-pretraining-for-language-understanding.pdf.

[37] Weipeng Huang, Xingyi Cheng, Taifeng Wang, and Wei Chu. Bert-based multi-head selection for joint entity-relation extraction. In *International Conference on Natural Language Processing and Chinese Computing*, pages 713–723, 2019. doi:10.1007/978-3-030-32236-6_65.

[38] Haoyu Wang, Ming Tan, Mo Yu, Shiyu Chang, Dakuo Wang, Kun Xu, Xiaoxiao Guo, and Saloni Potdar. Extracting multiple-relations in one-pass with pre-trained transformers. In *Association for Computational Linguistics*, pages 1371–1377, 2019. doi:10.18653/v1/P19-1132.

[39] Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. A general framework for information extraction using dynamic span graphs. In *North American Chapter of the Association for Computational Linguistics*, pages 3036–3046, June 2019. doi:10.18653/v1/N19-1308.

[40] David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. Entity, relation, and event extraction with contextualized span representations. In *Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing*, pages 5788–5793, 2019. doi:10.18653/v1/D19-1585.

[41] Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. Publicly available clinical BERT embeddings. In *Clinical Natural Language Processing Workshop*, pages 72–78, 2019. doi:10.18653/v1/W19-1909.

[42] Ronan Collobert and Jason Weston. A unified architecture for natural language pro-

cessing: Deep neural networks with multitask learning. In *International Conference on Machine Learning*, page 160–167, 2008. doi:10.1145/1390156.1390177.

[43] Aaron Jaech, Larry P. Heck, and Mari Ostendorf. Domain adaptation of recurrent neural networks for natural language understanding. In *Interspeech*, volume 8, pages 690–694, 2016. doi:10.21437/Interspeech.2016-1598.

[44] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. Recurrent neural network for text classification with multi-task learning. In *International Joint Conference on Artificial Intelligence*, page 2873–2879, 2016. URL `https://www.ijcai.org/Proceedings/16/Papers/408.pdf`.

[45] Yi Luan, Chris Brockett, Bill Dolan, Jianfeng Gao, and Michel Galley. Multi-task learning for speaker-role adaptation in neural conversation models. In *International Joint Conference on Natural Language Processing*, pages 605–614, 2017. URL `https://www.aclweb.org/anthology/I17-1061`.

[46] Isabelle Augenstein and Anders Søgaard. Multi-task learning of keyphrase boundary classification. In *Association for Computational Linguistics*, pages 341–346, 2017. doi:10.18653/v1/P17-2054.

[47] Matthew E Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. Semi-supervised sequence tagging with bidirectional language models. In *Applied Computational Linguistics*, 2017. doi:10.18653/v1/P17-1161.

[48] Ramon Maldonado, Travis R Goodwin, and Sanda M Harabagiu. Active deep learning-based annotation of electroencephalography reports for cohort identification. In *AMIA Summits on Translational Science Proceedings*, page 229, 2017. URL `https://pubmed.ncbi.nlm.nih.gov/28815135/`.

[49] Hrayr Harutyunyan, Hrant Khachatrian, David C Kale, Greg Ver Steeg, and Aram

Galstyan. Multitask learning and benchmarking with clinical time series data. *Scientific Data*, 6(1):1–18, 2019. doi:10.1038/s41597-019-0103-9.

[50] Natasha Jaques, Sara Taylor, Ehimwenma Nosakhare, Akane Sano, and Rosalind Picard. Multi-task learning for predicting health, stress, and happiness. In *NIPS Workshop on Machine Learning for Healthcare*, 2016. URL https://www.media.mit.edu/publications/multi-task-learning-for-predicting-health-stress-and-happiness/.

[51] Rie Johnson and Tong Zhang. Effective use of word order for text categorization with convolutional neural networks. In *North American Chapter of the Association for Computational Linguistics*, pages 103–112, 2015. doi:10.3115/v1/N15-1011.

[52] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. Recurrent convolutional neural networks for text classification. In *Association for the Advancement of Artificial Intelligence*, page 2267–2273, 2015. URL https://dl.acm.org/doi/10.5555/2886521.2886636.

[53] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. In *International Conference on Learning Representations*, 2017. URL https://arxiv.org/abs/1703.03130.

[54] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning*, pages 282–289, 2001. URL https://dl.acm.org/doi/10.5555/645530.655813.

[55] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by Gibbs sam-

pling. In *Association for Computational Linguistics*, pages 363–370, 2005. doi:10.3115/1219840.1219885.

[56] Fuchun Peng and Andrew McCallum. Information extraction from research papers using conditional random fields. *Information Processing and Management*, 42(4):963–979, 2006. doi:10.1016/j.ipm.2005.09.002.

[57] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In *North American Chapter of the Association for Computational Linguistics*, pages 260–270, 2016. doi:10.18653/v1/N16-1030.

[58] Yi Luan, Mari Ostendorf, and Hannaneh Hajishirzi. Scientific information extraction with semi-supervised neural tagging. In *Empirical Methods in Natural Language Processing*, pages 2641–2651, 2017. doi:10.18653/v1/D17-1279.

[59] Nils Reimers and Iryna Gurevych. Optimal hyperparameters for deep lstm-networks for sequence labeling tasks. In *Empirical Methods in Natural Language Processing*, 2017. URL `https://arxiv.org/abs/1707.06799`.

[60] Suncong Zheng, Yuexing Hao, Dongyuan Lu, Hongyun Bao, Jiaming Xu, Hongwei Hao, and Bo Xu. Joint entity and relation extraction based on a hybrid neural network. *Neurocomputing*, 257:59 – 66, 2017. doi:10.1016/j.neucom.2016.12.075.

[61] Walker Orr, Prasad Tadepalli, and Xiaoli Fern. Event detection with neural networks: A rigorous empirical evaluation. In *Conference on Empirical Methods in Natural Language Processing*, pages 999–1004, 2018. doi:10.18653/v1/D18-1122.

[62] Yihe Pang, Jie Liu, Lizhen Liu, Zhengtao Yu, and Kai Zhang. A deep neural network model for joint entity and relation extraction. *IEEE Access*, 7:179143–179150, 2019. doi:10.1109/ACCESS.2019.2949086.

[63] Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. End-to-end neural coreference resolution. In *Empirical Methods in Natural Language Processing*, pages 188–197, September 2017. doi:10.18653/v1/D17-1018.

[64] Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Empirical Methods in Natural Language Processing*, pages 3219–3232, 2018. doi:10.18653/v1/D18-1360.

[65] Zhongliang Yang, Yongfeng Huang, Yiran Jiang, Yuxi Sun, Yu-Jin Zhang, and Pengcheng Luo. Clinical assistant diagnosis for electronic medical record based on convolutional neural network. *Scientific Reports*, 8(1):1–9, 2018. doi:10.1038/s41598-018-24389-w.

[66] Yuan Luo, Yu Cheng, Özlem Uzuner, Peter Szolovits, and Justin Starren. Segment convolutional neural networks (Seg-CNNs) for classifying relations in clinical notes. *Journal of the American Medical Informatics Association*, 25(1):93–98, 08 2017. doi:10.1093/jamia/ocx090.

[67] Xinbo Lv, Yi Guan, Jinfeng Yang, and Jiawei Wu. Clinical relation extraction with deep learning. *International Journal of Hybrid Information Technology*, 9(7):237–248, 2016. doi:10.14257/ijhit.2016.9.7.22.

[68] Franck Dernoncourt, Ji Young Lee, Ozlem Uzuner, and Peter Szolovits. De-identification of patient notes with recurrent neural networks. *Journal of the American Medical Informatics Association*, 24(3):596–606, 2017. doi:10.1093/jamia/ocw156.

[69] Abhyuday N Jagannatha and Hong Yu. Bidirectional RNN for medical event detection in electronic health records. In *North American Chapter of the Association for Computational Linguistics*, pages 473–482, 2016. doi:10.18653/v1/N16-1056.

[70] Xue Shi, Yingping Yi, Ying Xiong, Buzhou Tang, Qingcai Chen, Xiaolong Wang, Zongcheng Ji, Yaoyun Zhang, and Hua Xu. Extracting entities with attributes in clinical text via joint deep learning. *Journal of the American Medical Informatics Association*, 26(12):1584–1591, 09 2019. doi:10.1093/jamia/ocz158.

[71] Shang Gao, Michael T Young, John X Qiu, Hong-Jun Yoon, James B Christian, Paul A Fearn, Georgia D Tourassi, and Arvind Ramanthan. Hierarchical attention networks for information extraction from cancer pathology reports. *Journal of the American Medical Informatics Association*, 25(3):321–330, 2018. doi:10.1093/jamia/ocx131.

[72] Özlem Uzuner, Ira Goldstein, Yuan Luo, and Isaac Kohane. Identifying patient smoking status from medical discharge records. *Journal of the American Medical Informatics Association*, 15(1):14–24, 2008. doi:10.1197/jamia.M2408.

[73] Sebastian Gehrmann, Franck Dernoncourt, Yeran Li, Eric T Carlson, Joy T Wu, Jonathan Welt, John Foote Jr, Edward T Moseley, David W Grant, Patrick D Tyler, et al. Comparing deep learning and concept extraction based methods for patient phenotyping from clinical narratives. *PloS One*, 13(2), 2018. doi:10.1371/journal.pone.0192360.

[74] Daniel J Feller, Jason Zucker, et al. Towards the inference of social and behavioral determinants of sexual health: Development of a gold-standard corpus with semi-supervised learning. In *AMIA Annual Symposium Proceedings*, page 422, 2018. URL `https://pubmed.ncbi.nlm.nih.gov/30815082/`.

[75] Genevieve B Melton, Sharad Manaktala, Indra Neil Sarkar, and Elizabeth S Chen. Social and behavioral history information in public health datasets. In *AMIA Annual Symposium Proceedings*, volume 2012, pages 625–34, 2012. URL `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3540479/`.

[76] Elizabeth S Chen, Elizabeth W Carter, Indra Neil Sarkar, Tamara J Winden, and

Genevieve B Melton. Examining the use, contents, and quality of free-text tobacco use documentation in the electronic health record. In *AMIA Annual Symposium Proceedings*, pages 366–374, 2014. URL https://pubmed.ncbi.nlm.nih.gov/25954340/.

[77] Elizabeth W Carter, Indra Neil Sarkar, Genevieve B Melton, and Elizabeth S Chen. Representation of drug use in biomedical standards, clinical text, and research measures. In *AMIA Annual Symposium Proceedings*, pages 376–385, 2015. URL https://pubmed.ncbi.nlm.nih.gov/26958169/.

[78] Yan Wang, Elizabeth S Chen, Serguei Pakhomov, Elizabeth Lindemann, and Genevieve B Melton. Investigating longitudinal tobacco use information from social history and clinical notes in the electronic health record. In *AMIA Annual Symposium Proceedings*, pages 1209–1218, 2016. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5333299/.

[79] Yan Wang, Elizabeth S Chen, Serguei Pakhomov, Elliot Arsoniadis, Elizabeth W Carter, Elizabeth Lindemann, Indra Neil Sarkar, and Genevieve B Melton. Automated extraction of substance use information from clinical texts. In *AMIA Annual Symposium Proceedings*, volume 2015, pages 2121–30, 2015. URL https://pubmed.ncbi.nlm.nih.gov/26958312/.

[80] Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, Russell Reas, Jiangjiang Yang, Darrin Eide, et al. Cord-19: The covid-19 open research dataset. *arXiv*, 2020. URL https://arxiv.org/abs/2004.10706.

[81] World Health Organization. Global literature on coronavirus disease, 2020. URL https://search.bvsalud.org/global-literature-on-novel-coronavirus-2019-ncov/.

[82] Xuan Wang, Xiangchen Song, Yingjun Guan, Bangzheng Li, and Jiawei Han. Compre-

hensive named entity recognition on cord-19 with distant or weak supervision. *arXiv*, 2020. URL `https://arxiv.org/abs/2003.12218`.

[83] Brett R South, Shuying Shen, Makoto Jones, Jennifer Garvin, Matthew H Samore, Wendy W Chapman, and Adi V Gundlapalli. Developing a manually annotated clinical document corpus to identify phenotypic information for inflammatory bowel disease. In *BMC Bioinformatics*, volume 10, 2009. doi:10.1186/1471-2105-10-s9-s12.

[84] Rob Koeling, John Carroll, Rosemary Tate, and Amanda Nicholson. Annotating a corpus of clinical text records for learning to recognize symptoms automatically. In *International Workshop on Health Text Mining and Information Analysis*, pages 43–50, 2011. URL `http://sro.sussex.ac.uk/id/eprint/22351`.

[85] David Cohn, Les Atlas, and Richard Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994. doi:10.1007/BF00993277.

[86] David A Cohn, Zoubin Ghahramani, and Michael I Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996. doi:10.1613/jair.295.

[87] Dan Shen, Jie Zhang, Jian Su, Guodong Zhou, and Chew-Lim Tan. Multi-criteria-based active learning for named entity recognition. In *Association for Computational Linguistics*, page 589–596, 2004. doi:10.3115/1218955.1219030.

[88] Yi Yang, Zhigang Ma, Feiping Nie, Xiaojun Chang, and Alexander G Hauptmann. Multi-class active learning by uncertainty sampling with diversity maximization. *International Journal of Computer Vision*, 113(2):113–127, 2015. doi:10.1007/s11263-014-0781-x.

[89] Bo Du, Zengmao Wang, Lefei Zhang, Liangpei Zhang, Wei Liu, Jialie Shen, and Dacheng Tao. Exploring representativeness and informativeness

for active learning. *IEEE Transactions on Cybernetics*, 47(1):14–26, 2017. doi:10.1109/TCYB.2015.2496974.

[90] W. Wu and M. Ostendorf. Graph-based query strategies for active learning. *IEEE/ACM Transactions on Audio, Speech, and Lang. Processing*, 21(2):260–269, 2013. doi:10.1109/TASL.2012.2219525.

[91] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *Journal Machine Learning Research*, 2:45–66, 2002. doi:10.1162/153244302760185243.

[92] Sungrae Park, Wonsung Lee, and Il-Chul Moon. Efficient extraction of domain specific sentiment lexicon with active learning. *Pattern Recognition Letters*, 56:38 – 44, 2015. doi:10.1016/j.patrec.2015.01.004.

[93] Yukun Chen, Thomas A Lasko, Qiaozhu Mei, Joshua C Denny, and Hua Xu. A study of active learning methods for named entity recognition in clinical text. *Journal of Biomedical Informatics*, 58:11–18, 2015. doi:10.1016/j.jbi.2015.09.010.

[94] Yukun Chen, Thomas A Lask, Qiaozhu Mei, Qingxia Chen, Sungrim Moon, Jingqi Wang, Ky Nguyen, Tolulola Dawodu, Trevor Cohen, Joshua C Denny, et al. An active learning-enabled annotation system for clinical named entity recognition. *BMC Medical Informatics and Decision Making*, 17(2):82, 2017. doi:10.1186/s12911-017-0466-9.

[95] Mahnoosh Kholghi, Lance De Vine, Laurianne Sitbon, Guido Zuccon, and Anthony Nguyen. Clinical information extraction using small data: An active learning approach based on sequence representations and word embeddings. *Journal of the Association for Information Science and Technology*, 68(11):2543–2556, 2017. doi:10.1002/asi.23936.

[96] Muqun Li, Martin Scaiano, Khaled El Emam, and Bradley A Malin. Efficient active learning for electronic medical record de-identification. *AMIA Summits on Transla-*

tional Science Proceedings, 2019:462, 2019. URL `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6568071/`.

[97] J. Gao, J. Chen, S. Zhang, X. He, and S. Lin. Recognizing biomedical named entities by integrating domain contextual relevance measurement and active learning. In *IEEE Information Technology, Networking, Electronic and Automation Control Conference*, pages 1495–1499, 2019. doi:10.1109/ITNEC.2019.8728991.

[98] Artem Shelmanov, Vadim Liventsev, Danil Kireev, Nikita Khromov, Alexander Panchenko, Irina Fedulova, and Dmitry V. Dylov. Active learning with deep pre-trained models for sequence tagging of clinical and biomedical texts. In *IEEE International Conference on Bioinformatics and Biomedicine*, pages 482–489, 2019. doi:10.1109/BIBM47256.2019.8983157.

[99] Ramon Maldonado and Sanda M. Harabagiu. Active deep learning for the identification of concepts and relations in electroencephalography reports. *Journal of Biomedical Informatics*, 98:103265, 2019. doi:10.1016/j.jbi.2019.103265.

[100] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint*, 1607.06450, 2016. URL `https://arxiv.org/abs/1607.06450`.

[101] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *International Conference on Machine Learning*, pages 160–167, 2008. doi:10.1145/1390156.1390177.

[102] Bing Liu and Ian Lane. Attention-based recurrent neural network models for joint intent detection and slot filling. In *INTERSPEECH*, pages 685–689, 2016. doi:10.21437/Interspeech.2016-1352.

[103] Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction.

In *Conference on Empirical Methods in Natural Language Processing*, pages 3219–3232, 2018. doi:10.18653/v1/D18-1360.

[104] Hrayr Harutyunyan, Hrant Khachatrian, David C Kale, Greg Ver Steeg, and Aram Galstyan. Multitask learning and benchmarking with clinical time series data. *Scientific Data*, 6(1):96, 2019. doi:10.1038/s41597-019-0103-9.

[105] Nikita Kitaev, Steven Cao, and Dan Klein. Multilingual constituency parsing with self-attention and pre-training. In *Association for Computational Linguistics*, page 3499–3505, 2019. doi:10.18653/v1/P19-1340.

[106] Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. ACE 2005 Multilingual Training Corpus LDC2006T06, 2006. URL `https://catalog.ldc.upenn.edu/LDC2006T06`.

[107] World Health Organization. Global status report on alcohol and health, 2014. URL `http://apps.who.int/iris/bitstream/10665/112736/1/9789240692763_eng.pdf`.

[108] Preetha Anand, Ajaikumar B Kunnumakara, Chitra Sundaram, Kuzhuvelil B Harikumar, Sheeja T Tharakan, Oiki S Lai, Bokyung Sung, and Bharat B Aggarwal. Cancer is a preventable disease that requires major lifestyle changes. *Pharmaceutical Research*, 25(9):2097–2116, 2008. doi:10.1007/s11095-008-9661-9.

[109] Max G Griswold, Nancy Fullman, Caitlin Hawley, Nicholas Arian, Stephanie RM Zimsen, Hayley D Tymeson, Vidhya Venkateswaran, Austin Douglas Tapp, Mohammad H Forouzanfar, Joseph S Salama, et al. Alcohol use and burden for 195 countries and territories, 1990–2016: a systematic analysis for the global burden of disease study 2016. *The Lancet*, 392(10152):1015–1035, 2018. doi:10.1016/S0140-6736(18)31310-2.

[110] Ahmedin Jemal, Michael J Thun, Lynn AG Ries, Holly L Howe, Hannah K Weir, Melissa M Center, Elizabeth Ward, Xiao-Cheng Wu, Christie Eheman, Robert Anderson, et al. Annual report to the nation on the status of cancer, 1975–2005, featuring

trends in lung cancer, tobacco use, and tobacco control. *Journal of the National Cancer Institute*, 100(23):1672–1694, 2008. doi:10.1093/jnci/djn389.

[111] Ozlem Uzuner, Peter Szolovits, and Isaac Kohane. i2b2 workshop on natural language processing challenges for clinical records. In *AMIA Annual Symposium Proceedings*, 2006.

[112] Aaron M. Cohen. Five-way smoking status classification using text hot-spot identification and error-correcting output codes. *Journal of the American Medical Informatics Association*, 15(1):32–5, 2008. doi:10.1197/jamia.M2434.

[113] Cheryl Clark, Kathleen Good, Lesley Jezierny, Melissa Macpherson, Brian Wilson, and Urszula Chajewska. Identifying smokers with a medical extraction system. *Journal of the American Medical Informatics Association*, 15(1):36–39, 2008. doi:10.1197/jamia.M2442.

[114] Jitendra Jonnagaddala, Hong-Jie Dai, Pradeep Ray, and Siaw-Teng Liaw. A preliminary study on automatic identification of patient smoking status in unstructured electronic health records. In *ACL Workshop on Biomedical Natural Language Processing*, volume 2015, pages 147–151, 2015. doi:10.18653/v1/W15-3818.

[115] Steven Bird, Edward Loper, and Ewan Klein. Natural language processing with python, 2009.

[116] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, et al. Scikit-learn: machine learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830, 2011. doi:https://dl.acm.org/doi/pdf/10.5555/1953048.2078195.

[117] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`.

[118] Hilary Daniel, Sue S Bornstein, and Gregory C Kane. Addressing social determinants to improve patient care and promote health equity: An american college of physicians position paper. *Annals of Internal Medicine*, 168(8):577–578, 2018. doi:10.7326/M17-2441.

[119] David U Himmelstein and Steffie Woolhandler. Determined action needed on social determinants. *Annals of Internal Medicine*, 168(8):596–597, 2018. doi:10.7326/M18-0335.

[120] Jane E Clougherty, Kerry Souza, and Mark R Cullen. Work and its role in shaping the social gradient in health. *Annals of the New York Academy of Sciences*, 1186:102–124, 2010. doi:10.1111/j.1749-6632.2009.05338.x.

[121] Joel D Kaufman, Sara D Adar, R Graham Barr, Matthew Budoff, Gregory L Burke, Cynthia L Curl, Martha L Daviglus, Ana V Diez Roux, Amanda J Gassett, David R Jacobs Jr, et al. Association between air pollution and coronary artery calcification within six metropolitan areas in the usa (the multi-ethnic study of atherosclerosis and air pollution): a longitudinal cohort study. *The Lancet*, 388(10045):696–704, 2016. doi:10.1016/S0140-6736(16)00378-0.

[122] Laura G Hooper and Joel D Kaufman. Ambient air pollution and clinical implications for susceptible populations. *Annals of the American Thoracic Society*, 15(Supplement 2):S64–S68, 2018. doi:10.1513/AnnalsATS.201707-574MG.

[123] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960. doi:10.1177/001316446002000104.

[124] Joseph Lilleberg, Yun Zhu, and Yanqing Zhang. Support vector machines and word2vec for text classification with semantic features. In *International Conference on Cognitive Informatics & Cognitive Computing*, pages 136–140, 2015. doi:10.1109/ICCI-CC.2015.7259377.

[125] Bishan Yang, Jian-Tao Sun, Tengjiao Wang, and Zheng Chen. Effective multi-label active learning for text classification. In *International Conference on Knowledge Discovery and Data Mining*, page 917–926, 2009. doi:10.1145/1557019.1557119.

[126] Jian Wu, Victor S Sheng, Jing Zhang, Pengpeng Zhao, and Zhiming Cui. Multi-label active learning for image classification. In *IEEE International Conference on Image Processing*, pages 5227–5231, 2014. doi:10.1109/ICIP.2014.7026058.

[127] Oscar Reyes and Sebastián Ventura. Evolutionary strategy to perform batch-mode active learning on multi-label data. *ACM Transactions on Intelligent Systems and Technology*, 9(4), 2018. doi:10.1145/3161606.

[128] Zunyou Wu and Jennifer M. McGoogan. Characteristics of and important lessons from the coronavirus disease 2019 (covid-19) outbreak in china: Summary of a report of 72314 cases from the chinese center for disease control and prevention. *Journal of the American Medical Association*, 323(13):1239–1242, 04 2020. doi:10.1001/jama.2020.2648.

[129] Jing Yang, Ya Zheng, Xi Gou, Ke Pu, Zhaofeng Chen, Qinghong Guo, Rui Ji, Haojia Wang, Yuping Wang, and Yongning Zhou. Prevalence of comorbidities in the novel wuhan coronavirus (covid-19) infection: a systematic review and meta-analysis. *International Journal of Infectious Diseases*, 2020. doi:10.1016/j.ijid.2020.03.017.

[130] Pauline Vetter, Diem Lan Vu, Arnaud G L'Huillier, Manuel Schibler, Laurent Kaiser, and Frederique Jacquerioz. Clinical features of COVID-19. *British Medical Journal*, 2020. doi:10.1136/bmj.m1470.

[131] Guoqing Qian, Naibin Yang, Ada Hoi Yan Ma, Liping Wang, Guoxiang Li, Xueqin Chen, and Xiaomin Chen. COVID-19 transmission within a family cluster by presymptomatic carriers in China. *Clinical Infectious Diseases*, 2020. doi:10.1093/cid/ciaa316.

[132] Wycliffe E Wei, Zongbin Li, Calvin J Chiew, Sarah E Yong, Matthias P Toh, and Vernon J Lee. Presymptomatic transmission of SARS-CoV-2—singapore, january 23–march 16, 2020. *Morbidity and Mortality Weekly Report*, 69(14):411, 2020. doi:10.15585/mmwr.mm6914e1.

[133] Chaomin Wu, Xiaoyan Chen, Yanping Cai, Xing Zhou, Sha Xu, Hanping Huang, Li Zhang, Xia Zhou, Chunling Du, Yuye Zhang, et al. Risk factors associated with acute respiratory distress syndrome and death in patients with coronavirus disease 2019 pneumonia in Wuhan, China. *JAMA Internal Medicine*, 2020. doi:10.1001/jamainternmed.2020.0994.

[134] Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. BRAT: a web-based tool for NLP-assisted text annotation. In *Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107, 2012. URL `https://www.aclweb.org/anthology/E12-2021`.

[135] S Wollenstein-Betech, CG Cassandras, and IC Paschalidis. Personalized predictive models for symptomatic COVID-19 patients using basic preconditions: Hospitalizations, mortality, and the need for an ICU or ventilator. *Medrxiv preprint*, 2020. doi:10.1101/2020.05.03.20089813.

[136] Dimitris Bertsimas, Léonard Boussioux, Ryan Cory Wright, Arthur Delarue, Vassilis Digalakis Jr., Alexandre Jacquillat, Driss Lahlou Kitane, Galit Lukin, Michael Lingzhi Li, Luca Mingardi, Omid Nohadani, Agni Orfanoudaki, Theodore Papalexopoulos, Ivan Paskov, Jean Pauphilet, Omar Skali Lami, Bartolomeo Stellato, Hamza Tazi Bouardi, Kimberly Villalobos Carballo, Holly Wiberg, and Cynthia Zeng. From predictions to prescriptions: A data-driven response to COVID-19. *arXiv preprint*, 2006.16509, 2020. URL `https://arxiv.org/abs/2006.16509`.

[137] Jose Luis Izquierdo, Julio Ancochea, and Joan B Soriano. Clinical characteristics and prognostic factors for icu admission of patients with COVID-19 us-

ing machine learning and natural language processing. *medRxiv preprint*, 2020. doi:10.1101/2020.05.22.20109959.

[138] Aiyuan Zhou, Yating Peng, David R Price, Hong Peng, Xin Liao, Peng Huang, Wenlong Liu, Zhi Xiang, Qimi Liu, Mingyan Jiang, et al. Symptoms at disease onset predict prognosis in covid-19 disease. *Research Square preprint*, 2020. doi:10.21203/rs.3.rs-25145/v1.

[139] Vageesh Jain and Jin-Min Yuan. Predictive symptoms and comorbidities for severe COVID-19 and intensive care unit admission: a systematic review and meta-analysis. *International Journal of Public Health preprint*, page 1, 2020. doi:10.1007/s00038-020-01390-7.

[140] Yalan Dong, Haifeng Zhou, Mingyue Li, Zili Zhang, Weina Guo, Ting Yu, Yang Gui, Quansheng Wang, Lei Zhao, Shanshan Luo, et al. A novel simple scoring model for predicting severity of patients with sars-cov-2 infection. *Transboundary and Emerging Diseases*, 2020. doi:10.1111/tbed.13651.

[141] Peng Peng Xu, Rong Hua Tian, Song Luo, Zi Yue Zu, Bin Fan, Xi Ming Wang, Kai Xu, Jiang Tao Wang, Juan Zhu, Ji Chan Shi, et al. Risk factors for adverse clinical outcomes with COVID-19 in China: a multicenter, retrospective, observational study. *Theranostics*, 10(14):6372, 2020. doi:10.7150/thno.46833.

[142] Laure Wynants, Ben Van Calster, Gary S Collins, Richard D Riley, Georg Heinze, Ewoud Schuit, Marc M J Bonten, Johanna A A Damen, Thomas P A Debray, Maarten De Vos, Paula Dhiman, Maria C Haller, Michael O Harhay, Liesbet Henckaerts, Nina Kreuzberger, Anna Lohmann, Kim Luijken, Jie Ma, Constanza L Andaur Navarro, Johannes B Reitsma, Jamie C Sergeant, Chunhu Shi, Nicole Skoetz, Luc J M Smits, Kym I E Snell, Matthew Sperrin, René Spijker, Ewout W Steyerberg, Toshihiko Takada, Sander M J van Kuijk, Florien S van Royen, Christine Wallisch, Lotty Hooft, Karel G M Moons, and Maarten van Smeden. Prediction models for diagnosis and

prognosis of COVID-19: systematic review and critical appraisal. *BMJ*, 369, 2020. doi:10.1136/bmj.m1328.

[143] Juan A. Siordia. Epidemiology and clinical features of COVID-19: A review of current literature. *Journal of Clinical Virology*, 127:104357, 2020. ISSN 1386-6532. doi:10.1016/j.jcv.2020.104357.

[144] John JY Zhang, Keng Siang Lee, Li Wei Ang, Yee Sin Leo, and Barnaby Edward Young. Risk factors of severe disease and efficacy of treatment in patients infected with COVID-19: A systematic review, meta-analysis and meta-regression analysis. *Clinical Infectious Diseases*, 2020. doi:10.1093/cid/ciaa576.

[145] Ji-Hyun Kim. Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap. *Computational statistics & data analysis*, 53(11):3735–3745, 2009. doi:10.1016/j.csda.2009.04.009.

[146] Scott M Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence*, 2(1):2522–5839, 2020. doi:10.1038/s42256-019-0138-9.

[147] Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. Classifying relations via long short term memory networks along shortest dependency paths. In *Empirical Methods in Natural Language Processing*, pages 1785–1794, 2015. doi:10.18653/v1/D15-1206.

[148] Zhiheng Li, Zhihao Yang, Chen Shen, Jun Xu, Yaoyun Zhang, and Hua Xu. Integrating shortest dependency path and sentence sequence into a deep learning framework for relation extraction in clinical text. *BMC Medical Informatics and Decision Making*, 19 (1):22, 2019. doi:10.1186/s12911-019-0736-9.

Appendix A

APPENDIX A

A.1 SDOH

Event type, e	**Argument type,** a	**Argument subtypes,** y_l	**Span examples**
Alcohol, Drug, or Tobacco	Trigger*	–	"alcohol"
	Status*	{none, current, past}	"denies"
	Duration	–	"for 8 years"
	History	–	"seven years ago"
	Type	–	"beer," "cocaine"
	Amount	–	"2 packs"
	Frequency	–	"daily," "monthly"
Employment	Trigger*	–	"works," "nurse"
	Status*	{employed, unemployed, retired, on disability, student, homemaker}	"works"
	Duration	–	"for five years"
	History	–	"15 years ago"
	Type	–	"nurse"
Living status	Trigger*	–	"lives"
	Status*	{current, past, future}	"lives," "lived"
	Type*	{alone, with family, with others, homeless}	"with husband"
	Duration	–	"for 6 months"
	History	–	"last month"
Insurance	Status	{yes, no}	"has been off"'
Sexual orientation	Status	{current, past}	"participated in"
	Type	{heterosexual, homosexual, bisexual}	"homosexual"
Gender identity	Status	{current, past}	"identifies as"
	Type	{cisgender, transgender}	"transgender"
Country of origin	Type	–	"England"
Race	Type	–	"African American"
Physical activity	Status	{none, current, past}	"currently jogs"
	Duration	–	"for 2 years"
	History	–	"10 years ago"
	Type	–	"walks"
	Amount	–	"4 miles"
	Frequency	–	"every evening"
Environmental exposure	Status	{none, current, past}	"no history"
	Duration	–	"since 2001"
	History	–	"until last month"
	Type	–	"asbestos"
	Amount	–	"significant"
	Frequency	–	"daily"

Table A.1: Annotation guideline summary for all event types. *indicates the argument is required.

Round	Source	Selection	Active learning training set	Train	Dev	Test	Total
1	MIMIC	Random	–	100	–	–	100
2	MIMIC	Random	–	144	56	–	200
3	MIMIC	Random	–	288	112	–	400
4	UW Dataset	Random	–	84	140	280	504
5	MIMIC	Active	572 samples (Round 3 train + 284 YVnotes)	400	–	–	400
6	UW Dataset	Random	–	168	120	240	528
7	MIMIC	Random	–	–	20	280	300
8	UW Dataset	Random	–	112	–	–	112
9	UW Dataset	Active	1336 samples (Rounds 3-8 *train* + 284 YVnotes)	728	–	–	728
10	UW Dataset	Active	2064 samples (Rounds 3-9 *train* + 284 YVnotes)	728	–	–	728
11	MIMIC	Active	3036 samples (Rounds 1-10 *train* + 284 YVnotes)	384	–	–	384
12	MIMIC	Random	–	–	–	96	96
			TOTAL	**3136**	**448**	**896**	**4480**

Table A.2: Annotation round summary, including selection type (random versus active) and training data used in active selection.

Parameter	Query function selection in Table 5.3	Active learning evaluation in Figure 5.6
batch size	20	100
learning rate	0.001	0.005
maximum gradient L2 norm	1.0	1.0
maximum length	200	200
number of epochs	500	500
LSTM hidden size	100	100
dropout, input to LSTM	0.7	0.4
dropout, output of LSTM	0.0	0.4
dropout, self-attention	0.7	0.4

Table A.3: Surrogate Classifier hyperparameters