DEEP LEARNING OF CLINICAL RELATION IDENTIFICATION IN HEALTH NARRATIVES

by

Ramon Manuel Martinez Maldonado

APPROVED BY SUPERVISORY COMMITTEE:

Sanda M. Harabagiu, Chair

Vincent Ng

Vibhav Gogate

Nicholas Ruozzi

This dissertation is dedicated to my family.

DEEP LEARNING OF CLINICAL RELATION IDENTIFICATION IN HEALTH NARRATIVES

by

RAMON MANUEL MARTINEZ MALDONADO, BS, MS

DISSERTATION

Presented to the Faculty of

The University of Texas at Dallas

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY IN

COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT DALLAS

May 2020

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor, Dr. Sanda Harabagiu. Without her attentiveness, guidance and feedback, this dissertation would not have been possible, nor would the publications that helped comprise the dissertation. Her knowledge, intellect, and grit have been a constant inspiration and motivation to me during my entire PhD. I would also like to thank my committee, Dr. Ng, Dr. Gogate, and Dr. Ruozzi. Their feedback and suggestions have helped improve this dissertation immensely.

I would also like to thank my fellow lab mates who have helped me along this journey: Dr. Travis Goodwin, Dr. Bryan Rink, Stuart Taylor, Omeed Ashtiani, and Maxwell Weinzierl. Working closely with each of you has been an unforgettable experience that in which I have learned most of what I know about programming and research. Most importantly, I would like to thank each of you for making the lab a fun place that I had the joy of spending time in each day.

Finally, I would like to thank my family. Your encouragement and belief in me has allowed me to believe in myself and accomplish my goals, and for that I am forever grateful.

March 2020

DEEP LEARNING OF CLINICAL RELATION IDENTIFICATION IN HEALTH NARRATIVES

Ramon Manuel Martinez Maldonado, PhD
The University of Texas at Dallas, 2020

Supervising Professor: Sanda M. Harabagiu, Chair

The worldwide adoption of electronic health records (EHRs) to document patient data enables the use of big-data methods to harness the medical information contained therein for secondary use. While most medical informatics research focuses on using the structured data found in EHRs, there is a substantial amount of information in the narratives of the records that is inaccessible without processing. This dissertation focuses on extracting this information in the form of medical concepts and relations between them. Specifically, deep learning methods are presented to perform (1) concept detection; and (2) relation extraction. Multiple deep learning methods are explored including recurrent neural networks, convolutional neural networks, memory networks, and attention networks. Methods are presented for performing these tasks in multiple genres of EHRs with differing target concept and relation schemata. Moreover, due to the data-hungry nature of deep learning models and the expertise necessary to annotate medical narratives, this dissertation addresses the problem of training deep learning models using multi-task active learning. These methods can be used to extract data-driven knowledge encoding clinical expertise from practicing clinicians. As such, this dissertation explores methods for representing such knowledge in graphical structures that can be used for question answering and to infer new knowledge. Moreover, these techniques are extended to represent biomedical knowledge from expert-curated ontologies

as embeddings. These embeddings expose otherwise inaccessible biomedical knowledge to deep learning models for relation identification.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

The widespread adoption of electronic health records (EHRs) around the world enables the use of big-data techniques to harness the rich medical information found throughout the records for secondary use. EHRs are patient-centered records generated by clinicians during their practice to document clinically relevant information gleaned during examinations, tests, procedures, and interviews with patients. As such, the information contained within EHRs represents *clinical knowledge* that is relevant to myriad applications throughout medical informatics including clinical trail screening and adverse drug reaction detection (Luo et al., 2016). In clinical trial screening, patients are identified for eligibility for a clinical trial based on certain eligibility criteria that can be found in a patient's EHR. Adverse drug reaction (ADR) detection is the task of identifying unintended adverse reactions caused by a medication and it is one of the leading causes of morbidity and mortality (Onder et al., 2002; Luo et al., 2016). Detecting instances of ADRs in a large corpus of EHRs can support translational research to improve patient outcomes and avoid potential patient injury. This kind of knowledge expressed in EHRs can be represented in relational form, e.g., in a structured database, that is readily accessible to automated systems. Formally, a relation is a triple of the form $\langle A1, R, A2 \rangle$ where the arguments $A1$ and $A2$ are medical concepts linked together in a relationship of type R. For clinical trail screening, eligibility criteria can be expressed in relational form, e.g., $\langle$liver function, status, adequate$\rangle$, which can be matched against relations expressed in EHRs to identify patients that meet the criteria. Likewise, instances of ADR relations can also be represented as relation triples, e.g., $\langle$penicillin, causes, rash$\rangle$, facilitating big-data systems for discovering novel potential ADRs in large corpora of EHR data.

However, much of this relational knowledge is not present in structured form in the EHR directly. Rather, it exists in the unstructured text portions of the EHR which are not readily

accessible for many applications, including those listed above (Chapman et al., 2011). The unstructured text portions of an EHR are referred to as *clinical narratives*. Clinical narratives vary widely in format and content, yet they all contain important clinical knowledge that can be extracted from their text. For example, the relation ⟨liver function, status, adequate⟩ can be extracted from the natural language sentence, "*liver function is adequate*" found in a patient's EHR.

Such relations can be extracted from clinical narratives using machine learning methods, specifically deep learning (Wang et al., 2018). Deep Learning is a sub-field of machine learning that includes computational models with multiple layers of abstraction trained using backpropagation (LeCun et al., 2015). In recent years, deep learning methods have been applied successfully to a wide range of prediction tasks including medical informatics (Ravì et al., 2016) and natural language processing (Young et al., 2018). In order to extract relations from clinical narratives, their arguments must be identified first. Medical concepts can also be extracted from clinical narratives using deep learning learning methods. However, training deep learning models to identify medical concepts and relations between them in clinical narratives requires large amounts of training data. Moreover, considerable expertise is often necessary to effectively create this training data due to the specialized nature of the text and annotation tasks. Therefore, Active Learning (Settles, 2009) is necessary in order to efficiently generate training data for training deep learning methods.

This dissertation focuses on the application of deep learning techniques for identifying medical concepts and relations between them across multiple genres of EHRs. The remainder of this chapter outlines the main contributions of the dissertation followed by a brief overview of planned future work.

1.1 Overview of Contributions

In this dissertation, deep learning methods are applied to medical concept detection and medical relation extraction along with Active Learning systems for training the deep learning methods. Knowledge graph embedding techniques for representing medical knowledge are also described. The knowledge extracted in clinical narratives can be represented in graphical structures, known as *knowledge graphs*. Knowledge graph embeddings distill the relational information of a knowledge graph into a set of discrete embeddings representing concepts and relations between them. Knowledge graphs generated from corpora of clinical narratives can be used to perform inference and identify new knowledge. Moreover, knowledge graphs can be used to expose relational knowledge from curated biomedical ontologies to deep learning models to enhance their representations of medical concepts and relations between them. Embeddings learned from the methods presented in this dissertation are shown to improve relation extraction in clinical narratives. An overview of the contributions of this work is provided below:

Chapter 2 addresses the task of medical concept detection on two types of EHR data: EEG reports and discharge summaries. EEG reports document the results of an electroencephalogram, while discharge summaries describe a patient's hospital stay when they are discharged. Neural models are presented for extracting medical concepts from both types of EHR. An electroencephalogram (EEG) is a medical test performed by neurologists in the treatment of epilepsy and other brain disorders. EEGs measure electrical activity along the scalp as a complex digital signal that can be correlated with brain activity. EEG reports are generated by neurologists to document the important phenomena gleaned from the signal and its clinical relevance, if any. As such, EEG reports contain a wealth of important clinical information in the form of medical concepts that describe the patient's current state, the characteristic signal activity observed during the EEG test (referred to as *EEG activities*),

and the possible medical problems that are indicated by the observed signal activity. In Chapter 2, a schema defining EEG-specific medical concepts and attributes that describe those concepts is defined. Two neural models are introduced: the stacked Long-Short Term Memory network (sLSTM) for identifying medical concepts; and the Deep ReLU Network (DRN) for identifying their attributes. The sLSTM operates on natural language sentences, modeling context to detect mentions of EEG-specific medical concepts. The DRN leverages deep learning to generate a multi-purpose embedding that is used to classify each attribute of a medical concept concurrently. Both neural models are evaluated and shown to be effective in identifying medical concepts and their attributes in EEG reports. In this chapter, we also address the task of identifying three types of medical concepts in discharge summaries: medical problems, tests, and treatments. We rely on the 2010 i2b2/VA challenge dataset (Uzuner et al., 2011) which provides a corpus of discharge summaries with manually annotated medical concepts. Each medical problem is assigned a belief value (referred to as an *assertion*) that describes how that medical problem was believed to have occurred (e.g., present, absent, possible). Chapter 2 presents a multi-task neural model for jointly identifying medical concept mentions and their assertions in discharge summaries. The proposed model relies on a massively pre-trained representation module for generating contextualized embeddings for each token in a sentence. The model is shown to advance the state-of-the-art in both concept detection and assertion classification.

Chapter 3 addresses the task of medical relation extraction from EEG reports. Four relations capturing EEG-specific medical knowledge are defined and two novel neural models for detecting them are presented. Traditionally, relation extraction methods operate at the sentence level, extracting relations between medical concepts mentioned in the same sentence. However, many important medical relations in EEG reports span sentence boundaries, sometimes crossing multiple sentences in an EEG report. Therefore, both neural methods introduced in Chapter 3 operate on the document level, identifying relations between medical

concepts mentioned anywhere in the report. The first neural model, EEG-RelNet, maintains a set of *memory* vectors that it updates recurrently as it reads through each sentence of an EEG report. The memory vectors maintain information pertaining to each medical concept mentioned in the report and each potential relation between each pair of concepts. EEG-RelNet is shown to be able to perform long-distance relation extraction in EEG reports. However, EEG-RelNet requires medical concepts and their attributes to be identified beforehand. The second neural model, the Self-Attention Concept, Attribute, and Relation identifier (SACAR) performs concept detection, attribute classification, and relation extraction jointly in an end-to-end manner. SACAR relies on a powerful neural architecture developed by Dehghani et al. (2018) to generate a contextualized representation of an EEG report. This representation is shared among a series of prediction layers, trained to extract information relevant to each prediction task. In this way, each prediction task helps to inform the others by constraining the shared representation during learning. Evaluations show that SACAR out performs the dedicated neural methods for concept detection and attribute classification introduced in Chapter 2 while performing competitively with EEG-RelNet for relation extraction.

Chapter 4 presents three Active Learning frameworks for training the sLSTM, DRN, EEG-RelNet, and SACAR models introduced in Chapters 2 and 3. Training neural models like these requires significant amounts of labeled data. Due to the substantial expertise involved in annotating EEG reports, we sought to generate the fewest number of manually annotated EEG reports necessary to train our neural models by employing active learning. The first framework, Memory-Augmented Active Deep Learning (MAADL) is presented for training EEG-RelNet to perform relation extraction. The second framework, Multi-Task Active Deep Learning (MTADL) is presented for training the sLSTM and DRN to identify the medical concepts and their attributes that function as the arguments of the relations extracted by

EEG-RelNet. The third framework, Improved Multi-Task Active Deep Learning with the Active Learning Policy Neural Network (MTADL+), identifies concepts, attributes, and relations jointly using the SACAR learner presented in Chapter 3. All three frameworks are used to perform an active learning loop whereby unlabeled EEG reports are selected from a pool for manual annotation. Both MAADL and MTADL adhere to the same five-step active learning paradigm whereby unlabeled EEG reports are selected for manual annotation using the uncertainty of their respective learners. MTADL+ leverages the Active Learning Policy Network (ALPNN) to *learn* how to select unlabeled EEG reports using the internal representation of the reports produced by SACAR.

Chapter 5 addresses the task of knowledge graph embedding. Three knowledge graph embedding frameworks are presented. The first framework encodes biomedical knowledge discovered from a large corpus of EEG reports such that it can be used for probabilistic inference. Specifically, we explore the application of knowledge graph embedding methods to noisy relations extracted from the narrative of EEG reports in order to discover new data-driven knowledge from clinical practice. To our knowledge, this is the first exploration of this application of knowledge graph embedding methods. The second framework presents a novel knowledge graph embedding method for representing a large biomedical ontology, namely the Unified Medical Language System (UMLS) (Lindberg et al., 1993). The UMLS is a comprehensive resource providing millions or relation instances between medical concepts. The framework leverages the rich representation of the medical concepts in the UMLS to produce embeddings for both concepts and relations. The UMLS embeddings are demonstrated to be useful in improving the results of an existing clinical prediction model. The third framework extends the second framework to the task of ontology alignment through the use of *alignment-oriented* embeddings. Alignment-oriented embeddings are learned from two disparate ontologies such that they reside in a single embedding space. Concepts and

relations from either ontology can be aligned using this embedding space. This framework is applied to the task of biomedical ontology alignment, producing alignments between three large biomedical ontologies.

Chapter 6 presents deep learning methods leveraging knowledge graph embeddings presented in Chapter 5 for extracting relations between the medical concepts in discharge summaries. Discharge summaries contain medical concepts that capture important information about a patient's care and status during a hospital stay. and are discussed in more detail in Section 2.3. Moreover, methods for identifying such concepts and their attributes are presented in Section 2.6. In Chapter 6, we investigate the efficacy of incorporating outside medical knowledge into relation extraction decisions through the use of UMLS knowledge embeddings derived by the system described in Section 5.3. The biomedical knowledge contained in the UMLS is particularly relevant to many of the relations present in discharge summaries and can be leveraged to inform concept representations in the model. Experimental results indicate that UMLS knowledge embeddings significantly improve the recall of a state-of-the-art relation extraction system, resulting in an increase in overall performance. Moreover, Chapter 7 demonstrates that deep learning methods for relation extraction are robust across genres of medical text.

Chapter 7 presents methods for extracting a class of relations known as Drug-Drug Interactions. Drug-Drug Interactions (DDIs) indicate when a prescription drug interacts with another substance causing an adverse reaction or event. Drug-drug interactions are of particular importance in medical informatics as they represent a preventable cause of adverse events, the eighth leading cause of death in the United States (Demner-Fushman et al., 2018). Each prescription drug certified by the Food and Drug Administration (FDA) is described by a medical reference document called a *Structured Product Labeling documents* (SPLs). SPLs

contain descriptions of the essential scientific information needed for the safe and effective use of a drug, including known drug-drug interactions. Therefore, the 2019 Text Analysis Conference (TAC) DDI Extraction from Drug Labels track was designed by the FDA and National Library of Medicine to facilitate research in DDI extraction from SPLs. The goal of the 2019 TAC DDI Extraction track is to extract the unique set of DDIs from each SPL and link each interaction to existing ontologies in order to facilitate interoperability with downstream systems. In Chapter 7 we present an end-to-end pipeline for identifying DDIs in SPLs including the medical concepts that participate in DDI relations and the type of DDI relation between them. At the heart of this pipeline is the Multi-Task Transformer for Drug-Drug Interaction (MTTDD) identifier. MTTDDI is a multi-task neural network based on the state-of-the-art Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019). Chapter 7 demonstrates that deep learning methods for relation extraction are robust across genres of medical text.

Finally, Chapter 8 presents possible directions for future work and Chapter 9 summarizes each chapter and concludes the dissertation.

CHAPTER 2

MEDICAL CONCEPTS IN ELECTRONIC HEALTH RECORDS

Medical concept extraction is the task of identifying medical concepts that occur in the free text of medical narratives and is a crucial first step in harnessing the unstructured data in the EHR. The accurate identification of medical concepts enables or enhances a bevy of crucial medical tasks, including patient cohort retrieval (Hersh, 2008), relation extraction (Uzuner et al., 2011), clinical decision support (Demner-Fushman et al., 2009), medical question answering (Goodwin, 2018), knowledge discovery (Maldonado et al., 2017), and predictive modeling (Stubbs et al., 2015). The language describing medical concepts in the free text of medical records can be complex. As such, identifying mentions of medical concepts in an EHR is often insufficient, e.g., in cases where a medical concept is mentioned in a hypothetical context – "*the patient may be unconscious*". This sort of characteristic information can be encoded in the form of *attributes* that describe a medical concept more fully. Given a schema of attribute types and values, *attribute classification* is the task of assigning a value to each attribute type for a medical concept. Trained on EHR data, machine learning techniques are capable of automatically extracting medical concepts and their attributes from medical text.

In this chapter[1][2], deep learning systems for performing automatic medical concept detection and attribute classification in two datasets of electronic health records are presented. The annotation schema for each dataset is presented, defining the medical concepts to be detected along with their attributes. The first dataset is comprised of electroencephalogram (EEG) reports while the second dataset is comprised of discharge summaries. Three

[1]©2019 Elsevier. Reprinted, with permission, from Ramon Maldonado and Sanda M. Harabagiu, *Active Deep Learning for the Identification of Concepts and Relations in Electroencephalography Reports*. Journal of Biomedical Informatics, Vol. 98 (2019): 103265.

[2]This chapter contains excerpts from Maldonado et al. (2017).

deep learning architectures for medical concept detection and attribute classification are presented in this chapter. The **stacked Long Short-Term Memory Network (sLSTM)** is presented for detecting medical concepts in EEG Reports. The **Deep ReLU Network (DRN)** is presented for classifying a rich set of attributes of the medical concepts identified in EEG Reports. The **Multi-Task BERT Graph Convolution Network (MT-BGCN)** is presented for jointly detecting three types of medical concepts in discharge summaries and classifying their attributes with a single neural model. The sLSTM uses deep learning to model the context of a medical narrative to detect medical concepts mentions. Likewise, the DRN uses deep learning to perform joint prediction on a set of 18 attributes of medical concepts found in EEG reports. Both the sLSTM and DRN operate on hand-crafted feature representations of the text in EEG reports. In contrast, the MT-BGCN is an end-to-end model that jointly performs medical concept detection and attribute classification using only the text of a clinical narrative and a dependency parse – without the need for hand-crafted features. Based on the pre-trained Transformer model BERT (Devlin et al., 2019), MT-BGCN leverages massive pre-training to learn abstract representations of text, first using general text data, then using clinical text data. Moreover, the MT-BGCN employs graph convolution to encode syntactic information useful for determining the boundaries of medical concept mentions in text.

The sLSTM and DRN are used in tandem to extract concept and their attributes: first the sLSTM is used to detect medical concepts mentioned in the free text of EEG reports, then the DRN is used to classify the attributes of each medical concepts that was detected. The MT-BGCN is used similarly, first to detect medical concepts, then to classify their attributes. However, while the sLSTM and DRN are two separate models, trained in isolation, the MT-BGCN is comprised of a single model that is trained to perform both tasks jointly. To this end, two forward passes of MT-BGCN are required during prediction – one for concept detection and one for attribute classification.

This chapter is organized as follows: Section 2.1 provides a brief background for the chapter, then Sections 2.2–2.5 describe the medical concepts and attributes found in EEG reports and discharge summaries, respectively. Section 2.4 presents the deep learning method for identifying medical concepts in EEG reports, while Section 2.5 presents the DRN for classifying their attributes. Section 2.6 presents the MT-BGCN for jointly identifying medical concepts and attributes in discharge summaries. Finally, Section 2.7 presents experimental results and discussions and Section 2.8 concludes the chapter.

2.1 Background

Clinical named entity recognition (NER) is the task of extracting medical concepts (named entities) from the free-text of clinical EHRs. Traditional methods for performing clinical NER were rule-based including Aronson (2001) and Friedman (1997). Machine learning methods, usually based on Conditional Random Fields (CRF) (Lafferty et al., 2001), were later introduced, improving performance. Hybrid methods combine traditional rule-based systems with machine learning (Demner-Fushman et al., 2017; Savova et al., 2010). More recently, neural methods for clinical NER have been introduced advancing the state-of-the-art in several common clinical NER datasets. These include Convolutional networks (Wu et al., 2017), Recurrent Networks (Huang et al., 2015), and Transformer networks (Verga et al., 2018) operating on word embeddings. Even more recent work in NLP has shown that methods that rely on word embeddings can be improved by replacing the word embeddings with *contextualized* word embeddings using a pre-trained representation layer (Peters et al., 2018; Devlin et al., 2019; Radford et al., 2019; Dai et al., 2019). In this chapter, we perform clinical NER on two datasets of EHRs: discharge summaries and EEG reports.

Prior work on clinical narratives related to EEG tests consider either clinical discharge summaries or EEG reports. The Epilepsy Data Extraction and Annotation (EpiDEA) (Cui

et al., 2012) system operates on discharge summaries originating from the Epilepsy Monitoring Unit (EMU) which collects clinical data about patients with potential of Sudden Unexpected Death in Epilepsy (SUDEP) at four centers: University Hospitals Case Medical Center (UH CMC Cleveland), Ronald Reagan University of California Los Angeles Medical Center (RRUMC-Los Angeles), The National Hospital for Neurology and Neurosurgery (NHNN, London, UK), and The Northwestern Memorial Hospital (NMH Chicago). These discharge summaries contain interleaving unstructured free text and semi-structured "attribute-value" text. The EpiDEA system applied regular expressions, concept identification provided by cTakes (Savova et al., 2010) and negation detection delivered by the Negex algorithm (Chapman et al., 2001) to map the clinical narratives into the concept classes provided by the Epilepsy and Seizure Ontology (EpSO) (Sahoo et al., 2013). EpSO has more than 1000 classes modeling the etiology of epilepsy. Cui et al (Cui et al., 2014) describes natural language processing (NLP) methods aiming at extracting epilepsy-related phenotypes from the same discharge summaries as those reported in Cui et al (Cui et al., 2012). The epilepsy phenotypes were related to (a) anatomical locations for the identification Epileptogenic Zone; (b) the Seizure Semiology; and (c) Lateralizing Signs, Interictal EEG Patterns, and Ictal EEG Patterns, which is a sub-set of the EEG activities we targeted in the research presented in this paper. The NLP methodology for phenotype extraction was also centered on the semantics of the EpSO ontology, but it incorporated the detection of anatomical locations available from MetaMap (Aronson, 2001) into the rule-based concept detector described in Cui et al (Cui et al., 2012). The methods introduced in this chapter operate on a clinical narrative dataset which is two orders of magnitude larger than the dataset of discharge summaries on which the methods from Cui et al. (2012, 2014) operate. Furthermore, we incorporate a rich set of attributes, including spatial (which consider anatomical locations) and temporal, and it covers the full set of EEG activities and events (including seizures) that are collected and interpreted during the EEG test. The NLP methods described in this chapter provide an information extraction framework that operates

on EEG-specific clinical narratives and takes advantage of Big Data technologies (including Deep Learning) in contrast with the methods described in Cui et al. (2012, 2014), which rely on the EpSO ontology (Sahoo et al., 2013).

This chapter also presents methods for detecting medical concepts in discharge summaries using the 2010 i2b2/VA challenge dataset (Uzuner et al., 2011). The medical concepts identified in the dataset include medical problems, tests, and treatments. Each medical problem has an associated *assertion* attribute that characterizes the mention of that medical problem. In 2010, a Hidden semi-Markov Model (De Bruijn et al., 2011) and a Support Vector Machine model (Roberts and Harabagiu, 2011) performed best at concept detection and assertion classification, respectively. Neural models have since advanced the state-of-the-art in concept detection on this dataset. Bhatia et al. (2019) introduce a multi-task sequence-to-sequence (Cho et al., 2014) model for jointly detecting medical concepts and their negations. More recently, several methods using contextualized word representations have set new top scores for concept detection (Zhu et al., 2018; Si et al., 2019; Fraser et al., 2019). These methods feed the contextualized word embeddings generated by pre-trained representation layers to simple dense or recurrent layers for prediction. Later in the chapter, we demonstrate the efficacy of incorporating syntactic information through the use of a more sophisticated model built atop the pre-trained representation layer for the concept detection in the 2010 i2b2/VA challenge dataset.

2.2 Medical Concepts and Attributes in EEG Reports

This section presents the medical concepts found in EEG reports and attributes that describe those concepts. Clinical electroencephalography (EEG) is the main investigation tool used for the diagnosis and management of epilepsy. It is also used to evaluate other types of brain disorders (Smith, 2005), including encephalopathies, neurological infections, Creutzfeldt-Jacob disease, and even the progression of Alzheimer's disease. An EEG records electrical

activity along the scalp and measures spontaneous electrical activity of the brain. The signals measured along the scalp can be correlated with brain activity, which makes it a primary tool for diagnosis of brain-related illnesses (Tatum IV, 2014). But as noted in Beniczky et al. (2013), the complexity of the EEG signal, interpreted and documented in EEG reports, produces inter-observer agreement known to be moderate. As more clinical EEG signals and reports become available, the interpretation of EEG signals can be improved by providing neurologists with results of search for patients that exhibit similar EEG characteristics.

CLINICAL HISTORY: This is a 55-year-old gentleman with [right leg swelling]$_{PROB}$, [ESRD]$_{PROB}$, [history of seizures]$_{PROB}$, and [hip fracture]$_{PROB}$.
MEDICATIONS: [Dilantin]$_{TR}$, [Haldol]$_{TR}$, many others.
INTRODUCTION: [Digital video EEG]$_{TEST}$ is performed at the bedside using standard 10-20 system of electrode placement with one channel of [EKG]$_{TEST}$. The patient is described as drowsy.
DESCRIPTION OF THE RECORD: The background EEG is diffusely [slow]$_{ACT}$ with primarily rhythmic [theta frequency activity]$_{ACT}$ of 5 to 7 Hz. There are frontally predominant, relatively synchronous [triphasic waves]$_{ACT}$ seen throughout the record. On one occasion, there may be some [asymmetries]$_{ACT}$, somewhat more remarkable on the left than the right. [Stimulation]$_{EV}$ of the patient produces cessation of the [triphasic waves]$_{ACT}$.
IMPRESSION: Abnormal EEG due to:
1. Generalized background [slowing]$_{ACT}$.
2. [Triphasic waves]$_{ACT}$.
CLINICAL CORRELATION: No [seizures]$_{PROB}$ were recorded. The [triphasic waves]$_{ACT}$ are typically a manifestation of underlying [metabolic encephalopathy]$_{PROB}$ including [hepatic encephalopathy]$_{PROB}$, [renal insufficiency]$_{PROB}$, or medication exposure. The [asymmetry]$_{ACT}$ of the triphasic waves, with prominence on the left, may be due to preexisting [history of epilepsy]$_{PROB}$ and/or [structural brain disease]$_{PROB}$. No [previous EEGs]$_{TEST}$ were available for comparison.

Figure 2.1. Synthetic Example EEG Report.

Recently, Goodwin and M. Harabagiu (2016) have described the MERCuRY (Multimodal EncephalogRam patient Cohort discoveRY) system that uses deep learning to represent the EEG signal. MERCuRY operates on a multi-modal EEG index resulting from the automatic processing of both the EEG signal and the EEG reports that document and interpret them. The MERCuRY system allows neurologist to search a vast data archive of clinical electroencephalography (EEG) signals and EEG reports, enabling them to discover patient populations relevant to queries like *Q: Patients with triphasic waves suspected of*

encephalopathy. The discovery of a relevant patient cohort satisfying the characteristics expressed in queries such as Q relies on the ability to automatically and accurately recognize various medical concepts and their attributes, both in the queries and throughout the EEG reports. In Q we could recognize that *triphasic waves* represents an EEG activity, while *encephalopathy* is a medical problem. To find relevant patients for Q based on their EEG reports, the relevance models and the index implemented in the MERCuRY system consider the concepts identified in the query as well as the concepts identified in the EEG reports. For example, a patient from the cohort relevant to Q has the EEG report illustrated in Figure 2.1, where identified concepts have annotations indicating medical problems [PROB], treatments [TR], tests [TEST], EEG activities [ACT], and EEG events [EV]. This EEG report is relevant to Q because, as indicated in its *impression* section, the patient's *triphasic waves* represent one of the explanations for the abnormal EEG, while in the report's *clinical correlation* section, it is mentioned that the EEG activity identified by *triphasic waves* is a manifestation of the medical problem *encephalopathy*. Thus the EEG report illustrated in Figure 2.1 is relevant to the query Q because of the identified medical concepts and their attributes.

The Temple University Hospital (TUH) EEG corpus is a publicly available collection of EEG reports comprising over 25,000 EEG reports from over 15,000 patients collected over 12 years. Figure 2.1 depicts a synthetic record exemplifying a typical EEG report from the TUH EEG corpus. EEG reports are designed to convey a written impression of the visual analysis of the EEG along with an interpretation of its clinical significance. In accordance with the American Clinical Neurophysiology Society Guidelines for writing EEG reports, the reports from the TUH EEG Corpus start with a *clinical history* of the patient including information about the patient's age, gender, current medical conditions (e.g., *"right leg swelling"*), and relevant past medical conditions (e.g., *"history of seizures"*) followed by a list of *medications* the patient is currently taking (e.g., *"Dilantin"*), described in a separate section. Together,

Table 2.1. Medical concept types and their attribute types.

Concept Type	Polarity	Modality	EEG Activity-Specific
EEG Activity	✓	✓	✓
EEG Event	✓	✓	
Problem	✓	✓	
Test	✓	✓	
Treatment	✓	✓	

these two initial sections depict *the clinical picture and therapy* of the patient, containing a wealth of medical concepts including *medical problems* (e.g., *"stroke"*), symptoms (e.g., *"facial droop"*), signs (e.g., *"twitching"*) and treatments (e.g., *"Haldol"*, *"gastrocnemius surgery"*). After the clinical picture and therapy of the patient is established, the *introduction* section of the EEG report describes the techniques used for the current EEG (e.g., *"digital video EEG using standard 10-20 system of electrode placement with one channel of EKG"*), the patient's condition at the time of the record (e.g., *drowsy*), and possible activating procedures carried out (e.g., *"stimulation of the patient"*). The *description* section is the mandatory part of the report, meant to provide a complete and objective description of the EEG, noting all observed *EEG activities* (e.g., *"triphasic waves"*, *"PLEDs"*), and *EEG events* (e.g., *"stimulation"*, *"hyperventilation"*). The *impression* section indicates whether or not the EEG test is abnormal and, if so, lists the abnormalities in decreasing order of importance. These abnormalities usually describe EEG activities (e.g., *"triphasic waves"*), but can also describe EEG Events (e.g., *"myoclonic seizures"*). Finally, the *clinical correlation* section explains what the EEG findings mean in terms of clinical interpretation, (e.g., *"asymmetry of the triphasic waves ... may be due to preexisting history of epilepsy and/or structural brain disease"*).

From the narratives of each section from every EEG report, we extract five types of concepts: (1) EEG activities, (2) EEG events, (3) medical problems, (4) medical treatments, and (5) medical tests, because they represent the predominant types of concepts in the EEG

reports (as illustrated in Figure 2.1). We were able to take advantage of the definitions of three types of medical concepts, which were used in the 2010 i2b2 challenge (Uzuner et al., 2011), namely *medical problems* (e.g., disease, injury), *tests* (e.g., diagnostic procedure, lab test), and *treatments* described in the previous section. For the EEG-specific medical concepts (i.e., EEG activities and EEG events) we created our own definitions. When deciding on the attributes associated with the five types of medical concepts from EEG reports, as illustrated in Table 2.1, we distinguished between attributes that apply to all types of concepts, e.g., *polarity* and *modality*, and attributes that are specific only to EEG activities. For identifying the polarity of medical concepts in EEG reports, we relied on the definition used in the 2012 i2b2 challenge (Sun et al., 2013), considering that each concept can have either a "positive" or a "negative" polarity, depending on the absence or presence of negation of its finding. When we considered the recognition of the modality of concepts, we took advantage of the definitions used in the same i2b2 challenge, where modality was used to capture whether a medical event discerned from a medical record actually happens, is merely proposed, mentioned as conditional, or described as possible. We extended this definition such that the possible modality values of "factual", "possible", and "proposed" indicate that medical concepts mentioned in the EEGs are actual findings, possible findings and findings that may be true at some point in the future, respectively. Through the identification of polarity and modality of the medical concepts, we aimed to capture the neurologist's beliefs about the medical concepts mentioned in the EEG report. For example, in the EEG report illustrated in Figure 2.1, we identified the medical problem *"right leg swelling"* with a "factual" modality and a "positive" polarity in the *clinical history* section, whereas the medical problem *"structural brain disease"* in the *clinical correlation* section was found to have the modality "possible" and the polarity "positive". In the same section, the medical problem *"seizures"* had the modality "factual" and the polarity "negative".

An EEG activity is defined as "an EEG wave or sequence of waves", while an EEG event is defined as "a stimulus that activates the EEG" by the International Federation of

Table 2.2. Attributes specific to EEG activities.

<table>
<tr>
<td rowspan="4">
Attribute 1: Morphology ::= represents the type or “form” of EEG waves.
• Rhythm: continuous rhythmic activity
• Transient
  • Single Wave:
    • Vertex wave
    • Wicket spikes
    • Spike
    • Sharp wave
    • Slow wave
  • Complex: A sequence of two or more waves recurring with a fairly consistent form, distinguished from background activity
    • K-complex
    • Sleep spindles
    • Spike-and-sharp-wave complex
    • Spike-and-slow-wave complex
    • Sharp-and-slow-wave complex
    • Triphasic wave
    • Polyspike complex
    • Polyspike-and-slow-wave-complex
  • Pattern: any characteristic EEG Activity
    • Suppression
    • Amplitude Gradient
    • Slowing
    • Breach Rhythm
    • Benign Epileptic Transients of Sleep (BETS)
    • Photic Driving (response)
    • Periodic Lateralized Epileptiform Discharges (PLEDs)
    • Generalized Periodic Epileptiform Discharges (GPEDs)
    • Epileptiform discharge (unspecified)
    • Disorganization
    • Positive Occipital Sharp Transients of Sleep (POSTS)
    • Unspecified: the default attribute value used if no morphological information is given
</td>
<td>
Attribute 2: Frequency Band
• Alpha (8 – 13 Hz)
• Beta (13 – 32 Hz)
• Delta (< 4 Hz)
• Theta (4 – 8 Hz)
• Gamma (> 32 Hz)
• N/A: the default value
</td>
</tr>
<tr>
<td>
Attribute 3: Background
• Yes
• No: the default value
</td>
</tr>
<tr>
<td>
Attribute 4: Magnitude ::= describes the amplitude of the EEG activity if it is emphasized in the EEG report
• Low: e.g., subtle (spike); small (polyspike discharges)
• High: e.g., high amplitude (spike); excess (theta)
• Normal: the default value
</td>
</tr>
<tr>
<td>
Attribute 5: Recurrence ::= describes how often the EEG activity occurs
• Continuous: the activity repeats in a continuous, uninterrupted manner
• Repeated: the activity repeats intermittently
• None: the activity occurs once; default value
</td>
</tr>
<tr>
<td>
Attribute 6: Dispersal ::= describes the spread of the activity over regions of the brain
• Localized (focal): limited to a small area of the brain
• Generalized (diffuse): occurring over a large area of the brain or both sides of the head
• N/A: the default value
</td>
<td>
Attribute 7: Hemisphere ::= describes which hemisphere of the brain the activity occurs in
• Right
• Left
• Both
• N/A: the default value
</td>
</tr>
</table>

Clinical Neurophysiology (Noachtar et al., 2004). Although previous efforts of identifying medical concepts in clinical narratives assumed that it is sufficient to automatically discover (a) the boundary of each mention of a concept; (b) the concept type; (c) its modality and

Table 2.3. Location attributes for EEG activities.

Location Attributes: **Brain Location** ::= describes the region of the brain in which the EEG activity occurs. The BRAIN LOCATION attribute of the EEG Activity indicates the location/area of the activity (corresponding to the electrode placement under the standard 10-20 system).
• *Attribute 8*: Frontal (i.e., Anterior): The frontal region of the brain including all F*, Fp* and AF* electrodes
• *Attribute 9*: Occipital (i.e., Posterior): The occipital region of the brain including all O* electrodes
• *Attribute 10*: Temporal: The temporal region of the brain including all T* electrodes
• *Attribute 11*: Central: The central region of the brain including all C* electrodes
• *Attribute 12*: Parietal: The parietal region of the brain including all P* electrodes
• *Attribute 13*: Frontocentral: The area between the frontal and central regions of the brain including all FC* electrodes
• *Attribute 14*: Frontotemporal: The area between the frontal and temporal regions of the brain including all FT* electrodes
• *Attribute 15*: Centroparietal: The area between the central and parietal regions of the brain including all CP* electrodes
• *Attribute 16*: Parieto-occipital: The area between the parietal and occipital regions of the brain including all PO* electrodes

(d) its polarity, the EEG activities could not be recognized in the same way. First, we noticed that EEG activities are not mentioned in a continuous expression. For example, in the narrative fragment: *"there are also bursts of irregular, frontally predominant [sharply contoured delta activity]$_{ACT}$, some of which seem to have an underlying [spike complex]$_{ACT}$ from the left mid-temporal region"* we can recognize one EEG activity that is mentioned by distant expressions in the narrative. To address this problem, we considered (a) the *anchors* of EEG activities and (b) their attributes. This allows us to automatically identify the anchors of EEG activities, annotate them in EEG reports while also recognizing the attributes of EEG activities and attaching them to the anchors, without needing to annotate all text spans referring to EEG activities in the reports. For this purpose, we defined 16 attributes which are specific to EEG activities, listed and defined in Tables 2.2 and 2.3. Because MORPHOLOGY best defines the EEG activities, we decided to use it as the anchor of each EEG activity, but its values also expressed the attributes of the EEG activities. When considering the MORPHOLOGY of EEG activities, we relied on a hierarchy of values, distinguishing first two types: (1) Rhythm and (2) Transient. In addition, the Transient

type contains three subtypes: Single Wave, Complex and Pattern. Each of these sub-types can take multiple possible values, illustrated in Table 2.2. In addition to MORPHOLOGY, we considered three classes of attributes for EEG activities, namely (a) general attributes of the waves, e.g., the FREQUENCY BAND , the BACKGROUND - which asserts whether the EEG activity occurs in the background or not; and MAGNITUDE; (b) temporal attributes and (c) spatial attributes. The only temporal attribute considered is RECURRENCE, which describes how often the EEG activity occurs. As spacial attributes, we considered the DISPERSAL, the HEMISPHERE and eight additional attributes for the BRAIN LOCATION where the EEG activity is observed, since an activity can simultaneously occur in more than one brain location. The BRAIN LOCATION attributes are enumerated in Table 2.3. All attributes specific to EEG activities have multiple possible values associated with them. Table 2.2 defines each of the 16 attributes of EEG activities and illustrates the possible values each of these attributes. In contrast, EEG events, which are frequently mentioned in EEG reports as well, can be recognized only by identifying the text span where they are mentioned.

2.3 Medical Concepts and Attributes in Discharge Summaries

Discharge summaries are clinical reports generated when a patient is discharged from a hospital stay. These reports, prepared by medical professionals, are the primary mode of communication between the hospital care team and subsequent care providers (Kind and Smith, 2008). As mandated by the Joint Commission[3], a discharge summary is to be comprised of six components in accordance with Standard IM.6.10, EP7 (Kind and Smith, 2008):

1. Reason for hospitalization;

2. Significant findings;

[3]The Joint Commission is a healthcare accreditation organization that accredits more than 21,000 organizations in the United States.

3. Procedures and treatment provided;

4. Patient's discharge condition;

5. Patient and family instructions (where applicable);

6. Attending physician's signature.

While each of these components is mandatory, the format of the report is not consistent between institutions and is largely realized in unstructured natural language. As such, NLP methods are necessary to extract useful information from discharge summaries in the form of medical concepts and their attributes.

Discharge summaries are rich with medical concepts including medical problems, treatments, and tests. Medical problems are defined as observations made by patients or clinicians about the patient's body or mind that are thought to be abnormal or caused by a disease (Uzuner et al., 2011). Medical problems can be mentioned throughout a discharge summary, for instance as the reason for hospitalization, a significant finding, and as a part of the patient's discharge condition. Treatments are defined as phrases that describe procedures, interventions, and substances given to a patient in an effort to resolve a medical problem (Uzuner et al., 2011). Treatments can be mentioned as part of a reason for hospitalization, a procedure/treatment provided during hospitalization, and patient/family instructions. Tests are phrases that describe procedures performed on a patient to discover, characterize, or rule out a medical problem and can be mentioned as leading to a significant finding or as administered during hospitalization. Consider the synthetic example: "*[An echocardiogram] revealed [a pericardial effusion] which was relieved by [a pericardiocentesis].*" Here three spans identified by brackets indicate a test ("*An echocardiogram*") a medical problem ("*a pericardial effusion*") and a treatment ("*a pericardiocentesis*"). Together these three types of medical concepts characterize the hospital visit and can be used for downstream automatic processing.

However, simply identifying the medical concepts mentioned in discharge summaries can lead to mischaracterization of a patient's hospital stay. While most of the medical concepts mentioned in a discharge summary indicate problems, tests, and treatments that are present or have happened, others may be mentioned in a hypothetical context, or as concepts that have been ruled out. For example, consider the medical problems in the following examples: EX1:"*The patient denies [pain].*"; EX2:"*[pneumonia] is possible*"; EX3:"*admitted for [stroke]*". While the "*stroke*" from EX3 has indeed occurred, the "*pain*" from EX1 has explicitly not occurred, and the "*pneumonia*" from EX2 is described as "*possible*", having not yet been identified as occurring or not. Each of these medical problems differ in the *belief status* of the physician that wrote the discharge summary. To describe the belief status of each medical problem, the 2010 i2b2/VA Shared-Task on Challenges in NLP for Clinical Data defined a set of 6 *assertions* enumerated below (Uzuner et al., 2011):

1. **Present**: the medical problem has occurred;
2. **Absent**: the medical problem has not occurred;
3. **Possible**: the medical problem may have occurred, but there is uncertainty;
4. **Conditional**: the medical problem occurs, but only under certain conditions;
5. **Hypothetical**: the medical problem may occur in the future;
6. **Associated with Someone Else**: the medical problem is mentioned, but not associated with the patient being discharged.

While any of the three medical concepts targeted by the shared task can be mentioned with one of these belief status assertions, only medical problems are annotated with assertions in the 2010 i2b2/VA challenge dataset.

In order to facilitate research in extracting medical concepts, their assertions, and relations between them in clinical free-text, the 2010 i2b2/VA challenge on concepts, assertions,

and relations in clinical text provided the research community with a set of 871 discharge summaries with medical concepts and their assertions manually annotated by medical professionals (Uzuner et al., 2011). The data contain 72,846 medical concepts with assertions for 30,518 medical problems (Roberts and Harabagiu, 2011). The data is split into a training set of 349 discharge summaries with 27k concepts and a test set of 477 discharge summaries with 45k concepts (Roberts and Harabagiu, 2011).

2.4 Deep Learning for Identifying Medical Concepts in EEG Reports

In this section, the stacked Long Short-Term Memory network (sLSTM) for detecting medical concepts in EEG reports is presented. The sLSTM is an initial application of deep learning to medical concept detection, operating on hand-crafted features, that leverages the expressive power of deep learning to improve prediction. The sLSTM architecture operates at the token level, representing each token in an EEG report as a feature vector and uses a series of LSTM layers to contextualize the token representations before using them to detect mentions of medical concepts in the text.

More specifically, the sLSTM identifies spans of text in medical narratives that correspond to medical concept mentions by assigning a label to each token in a sentence indicating if that token is a part of a medical concept mention. This process is known as *boundary detection* since we are identifying which tokens occur in the *boundaries* of a medical concept mention. Specifically, we represent a sentence as a sequence of tokens, $[w_1, w_2, ..., w_N]$, and train the models to assign a label $b_i \in \{I, O, B\}$ to each token w_i such that the token will receive a label $b_i = B$ if it is the beginning of a mention of a medical concept, a label $b_i = I$ if the token is inside of a mention of a medical concept, and a label $b_i = O$ otherwise. For example, the token sequence "*occasional left anterior temporal* $\lfloor$*sharp and slow wave complexes*$\rfloor_{ACT}$ *were seen*" would correspond to the label sequence $[O, O, O, O, B, I, I, I, I, O, O]$, where tokens {*occasional, left, anterior, temporal, were, seen*} are all assigned labels of O, the token

{*sharp*} is assigned a label of B, and the tokens {*and, slow, wave, complexes*} are all assigned labels of I.

2.4.1 The Stacked Long Short-Term Memory Network for Medical Concept Boundary Detection

The stacked Long Short-Term Memory Network (sLSTM) for medical concept boundary detection is a recurrent deep neural network that processes each token in a sentence, updating a shared memory state to use previous context to inform its predictions. The sLSTM is defined by a recurrent LSTM *cell* (Hochreiter and Schmidhuber, 1997) where the same cell is applied to each token sequentially such that the memory output of a cell for token t_i is taken as the memory input for the cell for token t_{i+1}. This is referred to as *recurrence*. As such the sLSTM model belongs to a class of neural network models known as *recurrent neural networks*. Using this shared memory, the sLSTM is able to incorporate information from each of its previous predictions to inform the current prediction, making it a well-suited for medical concept boundary detection.

The sLSTM described in this section represents each token as a *feature vector* encoding several linguistic phenomena listed in Table 2.4. The GENIA tagger (Tsuruoka and Tsujii, 2005) was used for tokenization, lemmatization, part-of-speech (PoS) tagging, and phrase chunking. Brown Clustering (Brown et al., 1992) is an unsupervised learning method that discovers hierarchical clusters of words based on their contexts. The Unified Medical Lan-

Table 2.4. Features for the stacked LSTM for Medical Concept Boundary Detection.

1.	The lemmas of the current token and the previous/next tokens
2.	The PoS of the token and the previous/next tokens
3.	The phrase chunk of the token and previous/next tokens
4.	The Brown cluster of the token
5.	The UMLS Concept Unique Identifier (CUI) of the UMLS concepts containing the token
6.	The title of the section containing the token

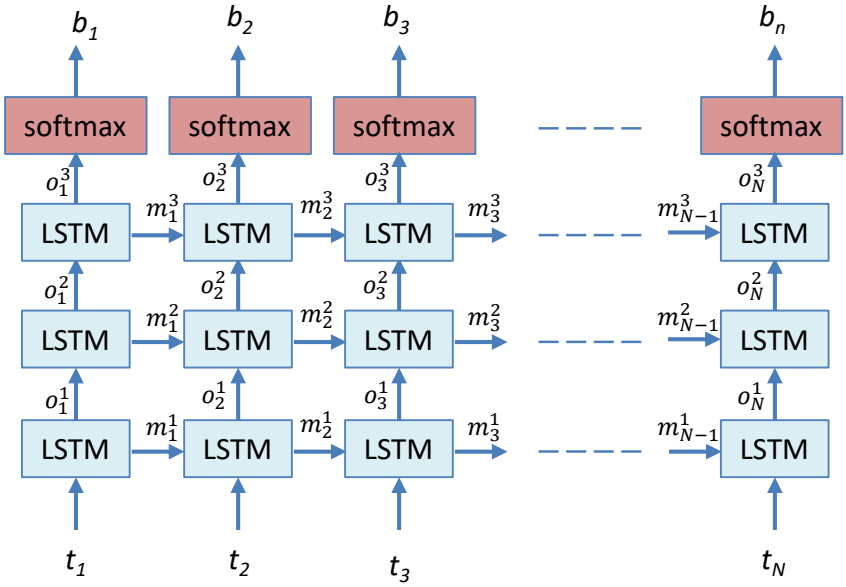

Figure 2.2. The Stacked Long Short-Term Memory Network for Medical Concept Boundary Detection

guage System (UMLS) (Lindberg et al., 1993) is a large biomedical ontology that enumerates most medical concepts of interest, assigning each concept a unique identifier (CUI).

The stacked LSTM is illustrated in Figure 2.2. The sLSTM operates on sentences from the narrative of clinical text. Each token in a sentence $w_i, i \in [1, ..., N]$ is represented by a feature vector t_i, provided as input to the sLSTM, which predicts a label b_i for each token. To predict each label b_i, the sLSTM considers (1) the vector representation of the token, t_i; as well as (2) the vector representation of all previous tokens from the sentence via a shared *memory* vector that is updated as the sLSTM processes each token. In order to update the shared memory vector, the LSTM cell maintains a series of gating mechanisms defined by the following equations:

$$\tilde{m}_i = \tanh\left(W_m[o_{i-1}, t_i] + b_m\right) \tag{2.1}$$

$$g_i = \sigma\left(W_g[o_{i-1}, t_i] + b_g\right) \tag{2.2}$$

$$\tag{2.3}$$

$$f_i = \sigma\left(W_f[o_{i-1}, t_i] + b_f\right) \tag{2.4}$$

$$m_i = f_t \cdot m_{i-1} + g_i \cdot \tilde{m}_i \tag{2.5}$$

where $W_m, W_g, W_f \in \mathcal{R}^{d\times d}$ are weight matrices, $b_m, b_g, b_f \in \mathcal{R}^d$ are bias vectors, σ is the sigmoid function, and $\cdot$ denotes element-wise multiplication. The vector $\tilde{m}_i$ is referred to as the *candidate memory* and is combined with the previous memory state, m_{i-1} to form the new memory state, m_i in equation 2.5. The vector f_i is referred to as the *forget gate* since it modulates how much of the old memory state should be forgotten. Likewise, the gating vector g_i modulates how much of the candidate memory, $\tilde{m}_i$ is incorporated into the new memory state. The output of each LSTM cell is then calculated as follows, given the updated memory state:

$$c_i = \sigma\left(W_c[o_{i-1}, t_i] + b_c\right) \tag{2.6}$$

$$o_i = c_i \cdot \tanh(m_i) \tag{2.7}$$

where $W_c \in \mathcal{R}^{d\times d}$ is a weight matrix and $b_c \in \mathcal{R}^d$ is a bias vector.

LSTM cells have the property that they can be *stacked* such that the outputs of cells on level l are used as the input to the cells on level $l+1$. The sLSTM has three levels where the input to the first level is a sequence of token feature vectors and the output of the top level is used to determine the IOB labels for each token. To this end, the output from the top level, o_i^3, is passed through a softmax layer to produce a probability distribution over possible IOB labels. This is accomplished by computing a vector of probabilities, q_i using the softmax function such that $q_i, 0$ is the probability of label I, $q_i, 1$ is the probability of label O, and $q_i, 2$ is the probability of label B. Specifically,

$$r_i = W_r o_i^3 + b_r \tag{2.8}$$

$$q_{ij} = \frac{e^{r_{ij}}}{\sum_j e^{r_{ij}}} \tag{2.9}$$

where $W_r \in \mathcal{R}^{d \times 3}$ is a weight matrix, d is the hidden size of o_i^3, and $r \in \mathcal{R}^3$ is a bias vector. The predicted IOB label is then chosen as the label with the highest probability, $\hat{y_i} = \text{argmax}_j q_{ij}$.

The weights of the sLSTM are learned by minimizing the categorical cross entropy between the predicted label distribution and the actual label, y_i, for each token:

$$\mathcal{L} = -\sum_i \sum_j \text{I}[y_i = j] \log\left(q_{ij}\right) \tag{2.10}$$

where $\text{I}[y_i = j]$ is an indicator function testing for the true label of token w_i.

Two sLSTM models are trained to detect concept mentions in EEG reports: one for identifying EEG activity anchors, and one for identifying the spans of the other four medical concept types. As described in Section 2.2, EEG activities have complex surface forms, often spanning entire sentences As such, mentions of EEG activities are identified by the text that indicates their morphology attribute, referred to as the *anchor* of the EEG activity. Since the other medical concepts do not have anchors, we identify them by the full span that describes the concept – usually limited to a single phrase. Therefore, it is beneficial to have an entirely two separate models is for identifying EEG activity anchors and spans of the other medical concepts.

2.5 Deep Learning for Classifying Attributes in EEG Reports

In this section, the Deep ReLU Network (DRN) for classifying attributes of medical concepts in EEG reports is presented. The DRN is a deep neural network that performs joint classification of a set of inter-related attributes of the same medical concept (i.e., polarity and modality or the 16 EEG activity-specific attributes described in Section 2.2). After mentions of medical concepts have been automatically identified using methods presented in Section 2.4, the DRN is used to identify the value of each attribute that characterizes the detected medical concepts. Traditional approaches to attribute classification require training

separate classifiers for each attribute, e.g., SVMs. However, by leveraging deep learning, we are able to model a set of attribute classification tasks jointly and perform them with the same network via multi-task learning.

The DRN is a deep feed-forward network operating on a feature vector representing a medical concept that learns a multi-purpose embedding used to predict each attribute. In this way, the DRN is able to use training signals from each attribute task to inform the others.

Formally, the task of attribute classification is defined as a set of traditional multi-class classification problems, one for each attribute type. For attribute $a \in A$ that characterizes a medical concept, a model is trained to produce a probability distribution over the possible values that attribute a can take, q_a. In this way, an attribute classification system should produce a series of distributions q_a for each attribute $s \in A$.

2.5.1 The Deep ReLU Network for Attribute Classification

The Deep ReLU Network (DRN) for attribute classification is a multi-task neural network for jointly performing attribute classification on a set of inter-related attributes that characterize a medical concept in an EEG report. The DRN operates on a feature vector describing a medical concept and the context in which it is found in a medical narrative, distilling from the feature vector a *multi-task embedding* which is used to perform concurrently a set of attribute classification tasks. Using a shared embedding allows important information to be shared between individual tasks. For example, when classifying the frequency of an EEG activity, it is beneficial to have information about the the activity's morphology since several morphologies preclude certain frequency ranges.

The DRN represents a medical concept as a feature vector encoding the medical concept mention itself, and the context in which it occurs in the free text of a medical narrative. The features are listed in Table 2.5 below:

Table 2.5. Feature representation of medical concepts used by the Deep ReLU Network for Attribute Classification.

1.	The text of the medical concept mention
2.	The lemmas of each token in the medical concept mention
3.	The PoS of each token in the medical concept mention
4.	The lemmas of three tokens before/after the medical concept mention
5.	The title of the section containing the medical concept mention
6.	The syntactic dependency path to t
7.	The number of words between the medical concept mention and t
8.	The number of "hops" in the syntactic dependency path from the head of the medical concept mention to t
9.	The number of medical concepts between the medical concept mention and t

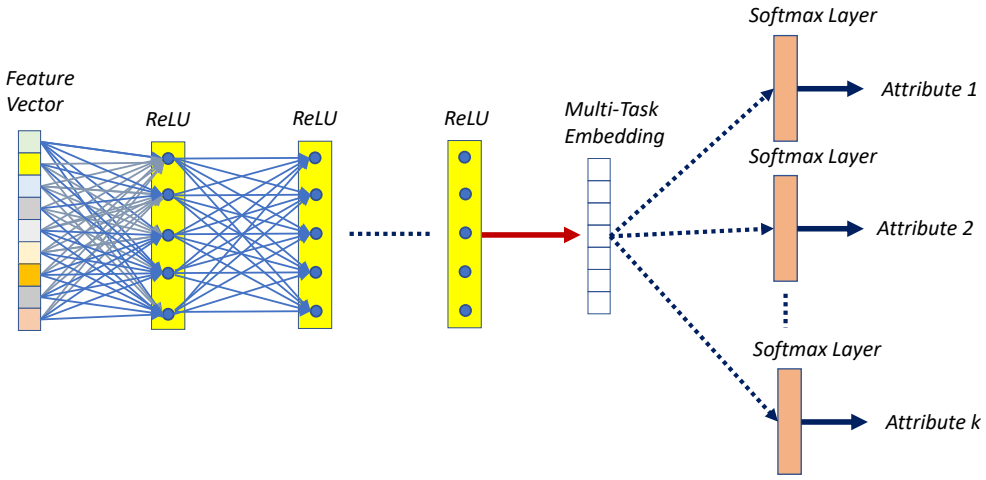

Figure 2.3. The Deep ReLU Network (DRN) for Attribute Classification shown performing joint prediction over k attribute classes.

As with the stacked LSTM of Section 2.4.1, the GENIA tagger (Tsuruoka and Tsujii, 2005) is used for tokenization and lemmatization, while Stanford CoreNLP (Chen and Manning, 2014) is used for dependency parsing. The *context features* 6-9 help the DRN capture long-distance context information which is helpful if the textual cues indicating the value of an attribute are far away. For example, due to the complex nature of EEG activities, it is not uncommon for entire sentences to describe a single activity, with attribute values being indicated by text throughout the sentence.

The DRN is presented in Figure 2.3. Given a feature vector x_c representing a medical concept c from a medical narrative, the DRN learns a d-dimensional multi-task embedding of the concept, $e_c \in \mathcal{R}^d$. The embedding e_c is produced by passing the feature vector through five fully-connected Rectified Linear Unit (ReLU) (Nair and Hinton, 2010) layers. Specifically, for layer $l \in \{1, ..., 5\}$,

$$r_c^l = \max\left\{0, W_l r_c^{l-1} + b_l\right\} \tag{2.11}$$

where $W_l \in \mathcal{R}^{d \times d}$ is a weight matrix for layer l, $b_l \in \mathcal{R}^d$ is a bias vector for layer l, and $r_c^0 = x_c$ is the feature vector representing medical concept c. The ReLU layers provide two benefits that allow the network to function properly at depth: (1) ReLUs allow for a deep network configuration and (2) they learn sparse representations, allowing them to perform de facto internal feature selection (Glorot et al., 2011). ReLU's allow for deep network configurations by avoiding the vanishing gradient problem. The vanishing gradient problem effects deep networks by causing them to lose information used to update weights in the network as the network gains depth (Bengio et al., 1994). The gradient with respect to a traditional unit (e.g., tanh, sigmoid) can cause gradient values to saturate because the gradient is scaled up or down at each layer. If the gradients are scaled down, this can cause the gradient values back-propagated to the input layers to trend towards zero, leading to stagnation. Because the ReLU activation effectively acts as a linear unit during back-propagation, the gradients are passed through the ReLU layers without scaling, effectively avoiding the problem of vanishing gradients.

As illustrated in Figure 2.3, the multi-task embedding produced by the DRN is used to perform prediction on a set of k attributes using a fully-connected softmax layer for each attribute. Specifically, for attribute a of concept c in a medical narrative, the DRN produces

a probability distribution q_c^a as follows:

$$r_c^a = W_r^a e_c^a + b_r^a \tag{2.12}$$

$$q_{cj}^a = \frac{e^{r_{cj}^a}}{\sum_j e^{r_{cj}^a}} \tag{2.13}$$

where $W_r^a \in \mathcal{R}^{d \times |a|}$ is a weight matrix, $|a|$ is the number of possible values for attribute a, and $b_r^a \in \mathcal{R}^{|a|}$ is a bias vector. The predicted attribute value is then chosen as the value with the highest probability, $\hat{y}_c^a = \text{argmax}_j q_{cj}^a$.

The weights of the DRN are learned using cross-entropy for each attribute, a, of each concept, c in the training corpus:

$$\mathcal{L} = -\sum_c \sum_a \sum_j \text{I}[y_c^a = j] \log\left(q_{cj}^a\right) \tag{2.14}$$

where $\text{I}y_c^a = j]$ is an indicator function testing for the true label of attribute a of concept c.

As the the sLSTMS for boundary detection, we train two DRM models classifying the attributes of medical concepts in EEG reports: one for EEG activities and one for the other four types of medical concepts. Two models are necessary since EEG activities have 16 attributes that only pertain the activities. Specifically we have a DRN model that is applied to EEG activities that predicts the 16 EEG activity specific attributes along with their polarity and modality. A second DRN model is trained to predict the polarity, modality, and type of the other medical concepts identified by EEG reports. Since the spans of the other four types of medical concepts are identified together by a single sLSTM model, their type needs to be predicted as well, so it is incorporated as an additional attribute.

2.6 Deep Learning for Jointly Identifying Medical Concepts and Assertions in Discharge Summaries

This section describes the Multi-Task BERT Graph Convolution Network (MT-BGCN) for jointly identifying medical concepts and assertions in discharge summaries. Sections 2.4 and

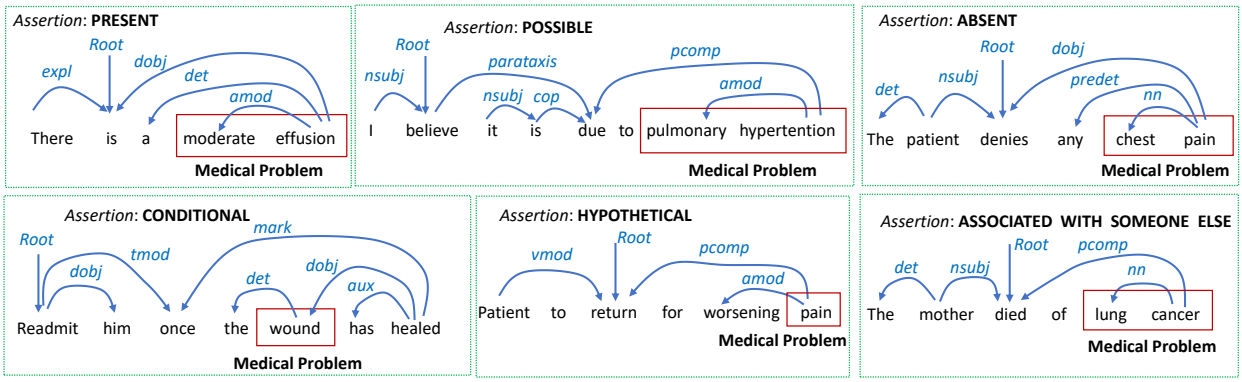

Figure 2.4. Syntactic dependencies between medical concepts and their narrative context indicative of the assertions.

2.5 introduced two distinct systems for medical concept boundary detection and attribute classification, respectively. However, as these two tasks are closely related, MT-BGCN is designed to perform both tasks jointly via multi-task learning, leveraging the training signals of each task to inform the other. As such, MT-BGCN leverages a sophisticated pre-trained representation layer to perform medical concept boundary detection (as in Section 2.4) and attribute classification (as in Section 2.5) jointly, with the same network. As described in Section 2.3, the we focus on a single attribute of medical problems that occur in discharge summaries called the *assertion*. Figure 2.4 illustrates examples of assertions regarding medical concepts mentioned in the context of clinical narratives. Moreover, Figure 2.4 illustrates the syntactic dependency parse for each example sentence to demonstrate the interactions between medical problems and their assertions.

Figure 2.4 illustrates how a PRESENT assertion of the medical concept *"moderate effusion"* can be inferred because the mention of the medical problem is a direct object (*dobj*) of the expletive (*exp*) used in the narrative context- a dependency pattern that is highly representative of medical problems which are PRESENT. Medical problems that are believed to be POSSIBLE may form a prepositional complement (*pcomp*) to verbs involved in a *parataxis* relation with verbs of judgement, such as "believe", *"think"* or *"consider"*. As shown in Figure2.4, the medical problem *"pulbmonary hypertention"* is a complement of the phrasal verb

"is due", which is involved in a parataxis relation with the judgement verb *"believe"*. Alternatively, when the mention of a medical concept is a direct object (*dobj*) is a verb that has negation connotations, the assertion receives a value of ABSENT. As illustrated in Figure 2.4, the medical problem *"chest pain"* is believed to be ABSENT because it is a direct object of the verb *"denies"*. Sometimes the assertion value related to a medical concept is inferred by a chain of dependency relations. For example, as illustrated in Figure 2.4, the medical problem *"wound"* is believed to be CONDITIONAL on its healing such that the readmission discussed in the context of the clinical narrative can occur. In this case, the value of the assertion is informed not only by the fact that the medical problem is a direct object (*dboj*) of the healing event, but also by temporal relationship (*tmod*) established in the dependency parse between the verb *"readmit"* and the temporal signal *"once"*, which is also a marker to the dependency relating it to the verb *"healed"*. A HYPOTHETICAL assertion of a medical concept can be inferred, as illustrated in Figure 2.4, when a dependency relation to a verb in infinitive (e.g., *"to return"*) which has the medical problem as a complement is headed by a preposition (*pcomp*) (e.g., *for ... pain"*). Figure 2.4 also shows how a *pcomp* dependency from a medical problem to a verb that has a subject mentioning a person in some relation (e.g., kinship) to the patient, the assertion that the medical problem is ASSOCIATED WITH SOMEONE ELSE can be inferred. Syntactic dependencies similar to those illustrated in Figure 2.4 can also be used to identify the span of words that cover the mention of a medical concept, as these words participate in specific relations that inform the assertion inference. MT-BGCN employs a *graph convolution network* to leverage the syntactic information made available by the dependency parse to more accurately identify medical concepts and their assertions.

MT-BGCN, depicted in Figure 2.5, is comprised of five modules operating at the sentence level:

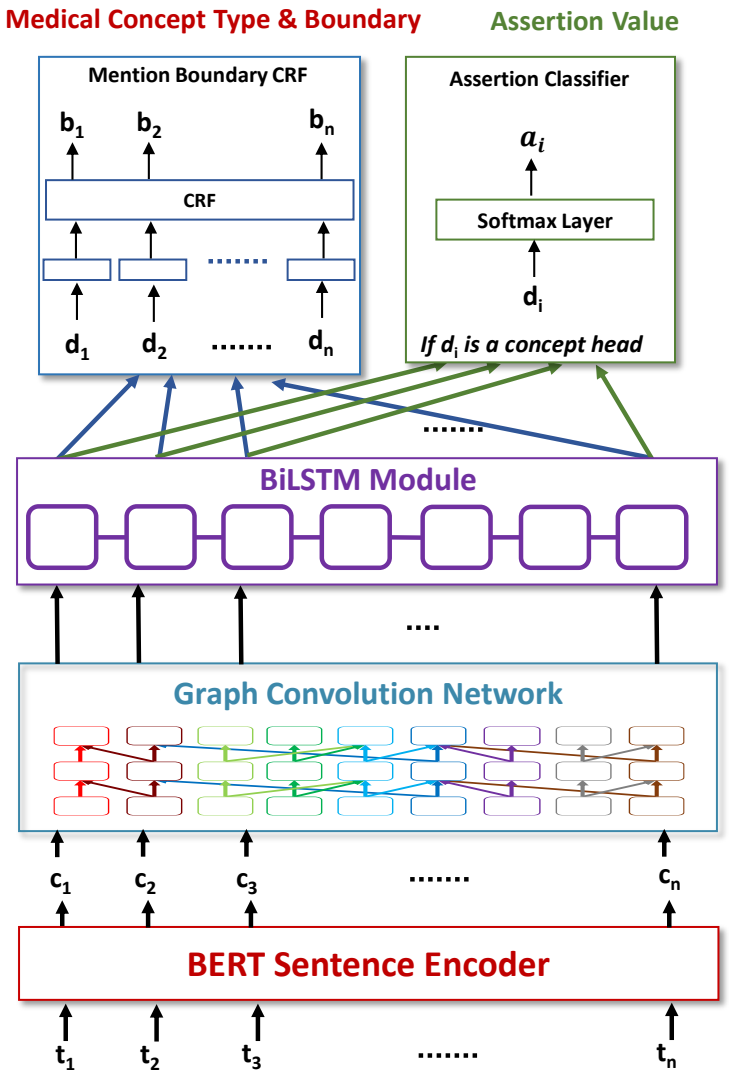

Figure 2.5. The Multi-Task BERT Graph Convolution Network for jointly identifying medical concepts and assertions in discharge summaries.

1. **The BERT Sentence Encoder** is the massively pre-trained transformer model of Devlin et al. (2019). It was pre-trained on a general text corpus using language modeling tasks, then fine-tuned on a clinical text corpus by Peng et al. (2019).

2. **The Graph Convolution Network** incorporates syntactic information in the form of Part-of-Speech (PoS) tags and contextualizes each token using its neighbors in the dependency graph using graph convolution.

3. **The BiLSTM Module** further contextualizes each token in the sentence via forward and backward connections in the BiLSTM.

4. **The Mention Boundary CRF** is comprised of a fully connected layer and a Conditional Random Field (CRF) (Lafferty et al., 2001) used to perform token-level medical concept boundary prediction.

5. **The Assertion Classifier** is a softmax layer for performing assertion classification.

The first three modules are used to extract contextualized representations of each token in a sentence which are shared among the two prediction modules, the Mention Boundary CRF and the Assertion Classifier. Each module is described in detail in the remainder of the section.

2.6.1 The BERT Sentence Encoder

The Bidirectional Encoder Representations from Transformers (BERT) model developed by Devlin et al. (2019) is a pre-trained Transformer encoder model that constructs contextualized token embeddings from a text sequence. BERT has been shown to be widely applicable in natural language processing, achieving state of the art results in several tasks including named entity recognition, question answering, and natural language inference (Devlin et al., 2019). Moreover, BERT has also been successfully applied to clinical NLP tasks as well (Peng et al., 2019; Fraser et al., 2019). BERT is part of a wider trend in NLP emphasizing the *contextualization* of word embeddings (Peters et al., 2018; Radford et al., 2019; Dai et al., 2019).

Traditional word embeddings (e.g., the GloVe embeddings used in Section 2.4) are used to represent a word in isolation whereas contextualized word embedding systems will produce different embeddings for the same word based on the context in which that word occurs. For example, the word "*cardiovascular*" would have a different representation if it occurred in the phrase "*cardiovascular distress*" vs. "*cardiovascular surgery*" in a contextualized embedding system, but the same representation in a static embedding system like GloVe or Word2Vec. Contextualized word embeddings produced by BERT are used in place of traditional word embeddings (e.g., GloVe (Pennington et al., 2014), Word2Vec (Mikolov et al., 2013)) to provide increased representational power, capturing for each word its deep connections to other words from the same context.

The BERT Sentence Encoder, illustrated in Figure 2.5 is a multi-layer bidirectional Transformer (Vaswani et al., 2017) encoder which operates at the sentence level, considering each sentence as a sequence of word-piece tokens. WordPiece tokenization (Wu et al., 2016) is completely data-driven and guaranteed to generate a deterministic segmentation for any possible sequence of characters of each sentence. This is especially important for clinical narratives where rare words, that are not available in common word embedding collections, are prevalent. For this reason, we used the word-piece token vocabulary of Devlin et al. (2019), as required by the BERT model. Given a sequence of word-piece tokens, $t_1, ..., t_n$, the BERT Sentence Encoder produces a sequence of contextualized token embeddings $c_1, ..., c_n$ using a stack of twelve Transformer layers. The Transformer (Vaswani et al., 2017) is a self-attentive model that contextualizes each token in a sentence using every other token in the same sentence, determining token-token relevance using a process called attention. In this way, the contextualized token representations, c_i contain bidirectional context information. The learning framework of the BERT Sentence Encoder has two steps: (1) *pre-training* and (2) *fine-tuning* During pre-training, BERT is trained to perform language modeling on a large collection of unlabeled free text. We have used the BLUE-BERT model which is pre-trained in two steps. First BLUE-BERT is trained on the BooksCorpus (800M words) (Zhu

et al., 2015) and the English Wikipedia (2,500M words) by Devlin et al.(Devlin et al., 2019) In order to adapt BLUE-BERT to the clinical domain, it was further pre-trained on PubMed abstracts (4,000M words) and MIMIC-III (500M words) by Peng et al. (Peng et al., 2019) Fine-tuning was straightforward since the self-attention mechanism in the Transformer allows BERT to model medical concept identification by swapping out the appropriate inputs and outputs. For fine-tuning, the BERT Sentence Encoder was first initialized with the pre-trained parameters, enabling all of the parameters to be fine-tuned.

2.6.2 The Graph Convolution Network for Incorporating Syntactic Information.

Because medical concepts are annotated on discharge summaries available for the 2010 i2b2/VA Challenge by considering only complete noun and adjectival phrases, while assertions are informed by dependency relations, it is imperative to incorporate in MT-BGCN syntactic information in two forms: (1) Part-of-Speech (PoS) information; and (2) information provided by dependency parsing. PoS information is incorporated at the token level in the form of a PoS tag embedding, while the dependency parse is used to further contextualize each token embedding, c_s, using graph convolution on the dependency graph. PoS tags and dependency parses are obtained using SciSpacy (Neumann et al., 2019), an NLP pipeline made for scientific and biomedical text. To incorporate PoS information, we concatenated the contextualized token embedding produced by the BERT Sentence Encoder with PoS embeddings:

$$s_i = [c_i, p_i] \tag{2.15}$$

where [] is the concatenation operation and p_i is the PoS embedding of token i. The PoS embeddings were initialized randomly and learned along with the rest of the parameters of the MT-BGCN model. While PoS tags representing lexico-syntactic information was

incorporated through embedding concatenation, the syntactic dependency information was incorporated using Graph Convolution Networks (GCNs).

GCNs are neural networks operating on graphs and inducing embeddings of nodes based on properties of their neighborhoods in the graph. In the MT-BGCN architecture, we consider the graph induced by the syntactic dependency parse where the tokens from a sentence comprise the nodes of the graph and the dependency relations constitute the edges. GCNs operating on the syntactic dependency parse have been shown to be effective in several NLP tasks including semantic role labeling (Marcheggiani and Titov, 2017) and relation extraction (Guo et al., 2019). An adjacency matrix A informed by the syntactic dependency of a sentence assigns $A_{ij} = 1$ if the token i is directly connected to token j in the dependency parse and 0 otherwise. For a sequence of token embeddings, $[s_1, ...s_n]$, a GCN computes a sequence of syntactically-informed embeddings $[g_1, ..., g_n]$ using the dependency graph as follows:

$$\text{GCN}(c_i) = \phi\left(\sum_j A_{ij}\left(W_g c_j + b_g\right)\right) \tag{2.16}$$

where W_g and b_g are a weight matrix and bias vector and ϕ is an activation function. We used the Gaussian Error Linear Unit (GELU) (Hendrycks and Gimpel, 2016) activation function in this work. By using the adjacency matrix A, the computation of $\text{GCN}(c_i)$ only considers those tokens which are connected to token i in the dependency parse. It should be noted that self-edges are added to the adjacency matrix (i.e., $A_{ii} = 1$ for each i) for each token such that each token, c_i, is used to inform it's own induced representation, $\text{GCN}(c_i)$.

Figure 2.6 illustrates how the GCN is processing the sentence: "*Complaining of low-grade fever and nasal congestion*", showcasing the PoS tags and dependency parse produced by ScispaCy along with three GCN layers having edges informed by the dependency parse. The GCN module combines each token representation with its part of speech tag and uses a series of graph convolution layers to further contextualize each token embedding using its neighbors in the dependency graph. Each token and its representations throughout the

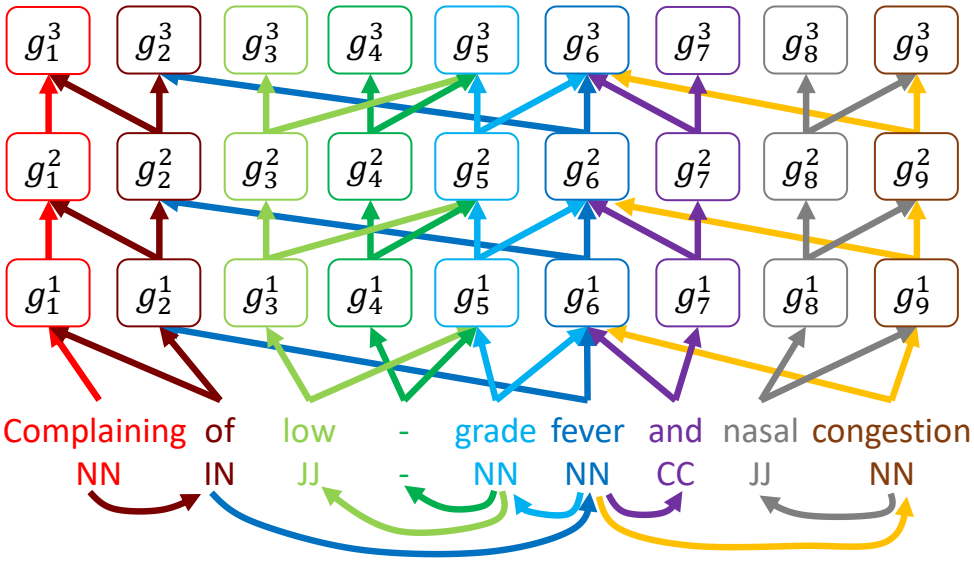

Figure 2.6. The Graph Convolution module for incorporating syntactic information.

GCN are highlighted in a different color in Figure 2.6. At each layer of the GCN, each token representation is a function of the tokens connected to it in the dependency graph. It is to be noted that at each layer of the GCN, each token representation is a function of the tokens connected to it in the dependency graph. We can see, for example, that the phrase "*low-grade fever*" is represented by its head token, "*fever*", since each token in the phrase has a connection to "*fever*" in the dependency graph. By stacking GCN layers, we allow for influence to spread along the graph. For example, the representation of the token, "*complaining*", depends on the token "*congestion*" in the third layer though the path (g_6^1, g_2^2, g_1^3).

In the GCN of MT-BGCN we used three GCN layers augmented with residual connections (He et al., 2016). Formally, this enabled the representation of token i in layer l to be calculated as:

$$g_i^l = g_i^{l-1} + \text{GCN}(g_i^{l-1}) \tag{2.17}$$

The residual connections combine the output of each GCN node with its input to cause each node in the GCN to *refine* its input instead of replacing it, thus leading to faster convergence,

especially in larger networks (He et al., 2016). The input of the first GCN layer, $g_i^0 = s_i$ as computed in Equation 2.15. It is to be noted that PoS tags and dependency parses are defined on the word level, not the word-piece level. Therefore, in MT-BGCN, word-pieces from the same word inherit that word's PoS tag. Moreover, edges are added to the adjacency matrix of the dependency graph connecting word-piece tokens of the same word.

2.6.3 The BiLSTM Module

The BiLSTM module is a simple 1-layer bidirectional LSTM that performs the final contextualization step. While the GCN incorporates important syntactic information into the model, the BiLSTM layer performs two functions the GCN does not: (1) it encodes local sequential information; and (2) it provides robustness in the face of erroneous dependency parses. Local sequential information is privileged in a BiLSTM architecture due to the connections between adjacent BiLSTM cells. This is also important for medical concept recognition since medical concept mentions are defined by contiguous token spans. Moreover, the BiLSTM module provides a fallback for contextualization in the event of errors in the dependency parse produced by SciSpacy. Due to the nature of discharge summaries, based on dictation, and including semi-structured text, this is not an uncommon occurrence.

The BiLSTM module produces an embedding d_i for each token by passing the embeddings g_i^3 produced by the GCN module through a bidirectional LSTM: $d_i = BiLSTM(g_i^3; g_:^3)$.

2.6.4 Prediction Modules in the MT-BGCN

There are two prediction modules in the MT-BGCN: the Medical Concept Type and Boundary CRF (MCTB-CRF) and the Assertion Classifier. The MCTB-CRF uses a Conditional Random Field (Lafferty et al., 2001) to generate the most likely boundary tag sequence for a sentence in order to identify the spans of medical concept mentions. Because it uses different boundary tags for each type of medical concept, it also identifies the concept type. The

Assertion Classifier uses a softmax layer to calculate a probability distribution over possible assertion values for a given medical concept using its token embedding.

Given a sequence of token embeddings from the BiLSTM module, $d_1, ..., d_n$, the MCTB-CRF uses a fully connected layer to produce a vector of potentials[4], $\tilde{d}_i \in \mathcal{R}^3$ with one potential for each possible IOB tag. A CRF is trained to predict the highest likelihood tag sequence given a sequence of vectors of potentials, $\tilde{d}_1, ..., \tilde{d}_n$. The CRF uses the likelihood of transitioning between IOB tags measured during training to enforce consistency in its predictions during testing (e.g., it will be very unlikely to predict an 'I' tag after an 'O' tag since this is never seen during training). Since there are three medical concept types, the MCTB-CRF has three fully connected layers that produce three sequences of vectors of potentials, one for each concept type. However, the same CRF is shared for each of the three concept types since they share the output label space and their label transition probabilities should align. The MCTB-CRF is trained to minimize the negative log likelihood of the correct tag sequence s:

$$\mathcal{L}_B = -\log P(s|\tilde{d}_1, ..., \tilde{d}_n) \tag{2.18}$$

In fact, the MCTB-CRF uses three loss functions, one for each concept type, $\mathcal{L}_{BP}$, $\mathcal{L}_{BTe}$, $\mathcal{L}_{BTr}$ for problems, test, and treatments, respectively.

The Assertion Classifier is comprised of a single softmax layer that produces a probability distribution over assertion values: $\alpha_i = \text{softmax}(W_\alpha d_i + b_\alpha)$ where $W_\alpha \in \mathcal{R}^{d \times 6}$ is a weight matrix and $b_\alpha \in \mathcal{R}^6$ is a bias vector. Only tokens d_i representing the start of a medical problem mention are considered by the Assertion Classifier. The Assertion Classifier is trained considering the minimization of a cross-entropy loss, defined as:

$$\mathcal{L}_A = -\sum_a \text{I}[y_i = a] \log(\alpha_{ia}) \tag{2.19}$$

[4]A potential is simply a score associated with an IOB tag indicating the likelihood of that tag.

where I$y = a$] is an indicator function that returns 1 if the assertion associated with the medical problem mentioned at token i has value $a \in [1, ..., 6]$ and 0 otherwise. A value of $a = 1$ indicates that the assertion of the medical problem has the value of PRESENT, a value of $a = 2$ corresponds to POSSIBLE, while $a = 3$ indicates ABSENT and $a = 4$ indicates CONDITIONAL; $a = 5$ indicates HYPOTHETICAL; $a = 6$ indicates ASSOCIATED WITH SOMEONE ELSE. MT-BGCN is trained to minimize the multi-task loss function defined as a linear interpolation of each loss:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{BP} + \lambda_2 \mathcal{L}_{BTe} + \lambda_3 \mathcal{L}_{BTr} + \lambda_4 \mathcal{L}_A \tag{2.20}$$

2.7 Experimental Results and Discussions

In order to evaluate the deep learning methods for identifying concepts and their attributes in electronic health records introduced in this chapter, we conduct experiments using the TUH EEG corpus and the 2010 i2b2/VA challenge dataset (Uzuner et al., 2011). We adopt the metrics used in the 2010 i2b2/VA challenge for both medical concept detection and attribute classification: precision (P), recall (R), and F_1-score (F_1) defined below:

$$P = \frac{tp}{tp + fp} \tag{2.21}$$

$$R = \frac{tp}{tp + fn} \tag{2.22}$$

$$F_1 = 2\frac{P * R}{P + R} \tag{2.23}$$

where tp, fp, and fn indicate true positives (prediction matches gold annotation), false positives (prediction does not match gold annotation) and false negatives (gold annotation does not match any prediction), respectively. For medical concept detection, we report exact and partial matching evaluations. For partial matching, overlapping medical concept spans are counted as true positives even if their boundaries do not exactly align, while for exact matching, their spans must align completely.

Table 2.6. Evaluation results for stacked LSTM models for concept boundary recognition evaluated with exact and partial matching against a CRF baseline.

	EEG Activity					
		Exact			Partial	
Model	Precision	Recall	F_1-Score	Precision	Recall	F_1-Score
CRF	0.8721	0.7650	0.8364	0.9489	0.8308	0.9095
sLSTM	**0.8949**	**0.8125**	**0.8517**	**0.9650**	**0.8968**	**0.9297**
	Other Medical Concepts					
		Exact			Partial	
Model	Precision	Recall	F_1-Score	Precision	Recall	F_1-Score
CRF	0.9047	0.8501	0.8683	0.9474	0.8773	0.9110
sLSTM	**0.9207**	**0.8538**	**0.8860**	**0.9469**	**0.9263**	**0.9332**

2.7.1 Evaluation of Deep Learning Systems for Identifying Concepts and Their Attributes in EEG Reports

In this subsection, we evaluate (1) the ability of the stacked LSTM (sLSTM) to detect anchors of EEG activities and the boundaries of medical problems, tests, and treatments; and (2) the ability of the Deep ReLU Network (DRN) to determine attributes of each medical concept. The evaluations are conducted on a set of 140 EEG reports in which concepts and attributes were manually annotated by three graduate students after extensive consultation with practicing neurologists. Average inter-annotator agreement, measured using Jaccard Score (Levandowsky and Winter, 1971), was 0.9658 and 0.9518 for concept boundaries and attributes, respectively. The evaluations were performed using a test set of 30 reports and a training set of 110 reports, with 10 held out for validation. There are 1184 EEG activities, 419 EEG events, 747 medical problems, 669 tests, and 500 treatments manually identified in the EEG reports.

Table 2.6 presents the results of the two sLSTM models for detecting (a) the anchors of EEG activities and (b) the boundaries of all other medical concepts (EEG events, medical problems, test, and treatments). The sLSTM models are compared with a linear chain Conditional Random Field (Lafferty et al., 2001) trained on the same features using an im-

Table 2.7. Performance of the DRN when automatically detecting attributes of EEG events, medical problems, tests, and treatments.

Attributes and Value	#	Precision	Recall	F_1-Score
Concept Type	**2335**	–	–	**0.939**
EEG Event	419	0.926	0.928	0.927
Medical Problem	747	0.901	0.960	0.929
Test	669	0.982	0.958	0.970
Treatment	500	0.964	0.898	0.930
Modality	**2318**	–	–	**0.659**
Factual	2199	0.971	0.990	0.980
Possible	64	0.634	0.406	0.495
Proposed	55	0.622	0.418	0.500
Polarity	**121**	–	–	**0.770**

plementation provided by CRFSuite (Okazaki, 2007). As described in Section 2.4, we train two separate sLSTM models due to the syntactic differences in the surface forms indicating EEG activity anchors versus the other medical concept boundaries. The results show that the performance of predicting EEG activity anchors was slightly lower than predicting the other medical concept boundaries. This is not surprising given the fragmented and complex language used in EEG activity descriptions in EEG reports, as noted in Section 2.2. However, with F_1 scores of 0.8517 and 0.8860, it is clear that the sLSTM model is able to accurately identify medical concept mentions in EEG reports. Recall from Section 2.5 that in order to differentiate medical problems, treatments, tests, and EEG activities, the concept type is encoded as an attribute of the medical concepts identified by the sLSTM along with polarity and modality. The results of the DRN for concept type, polarity and modality are presented in Table 2.7 along with the counts of each concept type and attribute value. When identifying medical concepts using the sLSTM for boundary detection and predicting their types using the DRN, we achieve F_1 scores of 0.8563, 0.8118, 0.8963, and 0.9074 for recognizing EEG events, medical problems, tests, and treatments, respectively. Table 2.8 presents the performance of the DRN for predicting the attributes of EEG activities. Aggregated metrics are presented for each attribute type using micro-average, however precision and recall are

Table 2.8. Performance of the DRN when automatically detecting attributes of EEG activities evaluated with precision (P), recall (R), and F_1-score (F_1).

Attributes and Values	#	P	R	F_1	Attributes and Values	#	P	R	F_1
Morphology	1438	–	–	0.7716	**Frequency Band**	1438	–	–	0.8754
Rhythm	388	0.6087	0.8660	0.7149	Alpha	128	0.8632	0.7891	0.8245
Vertex Wave	37	0.6977	0.8108	0.7500	Beta	91	0.6881	0.8242	0.7500
Wicket Spikes	11	0.5000	0.1818	0.2667	Delta	129	0.8416	0.6589	0.7391
Spike	43	0.8000	0.5581	0.6575	Theta	118	0.7755	06441.	0.7037
Sharp Wave	107	0.9195	0.7577	0.8247	N/A	972	0.9067	0.9410	0.9235
Slow wave	64	0.8929	0.7813	0.8333	**Magnitude**	1438	–	–	0.8148
K-complex	11	0.8889	0.7273	0.8000	Low	184	0.7548	0.5939	0.6648
Sleep spindles	28	0.5526	0.7500	0.6364	High	175	0.6273	0.6900	0.6571
Spike-and-slow wave	106	1.0000	0.6132	0.7602	Normal	1079	0.9210	0.9396	0.9302
Triphasic wave	16	0.8333	0.6250	0.7143	**Background**	1438	0.8904	0.8197	0.8543
Polyspike complex	29	0.7500	0.5172	0.6122	**Recurrence**	1438	–	–	0.7174
Suppression	48	0.5814	0.5208	0.5495	Continuous	224	0.7244	0.6546	0.6878
Slowing	174	0.9371	0.7701	0.8454	Repeated	262	0.7974	0.6461	0.7138
Breach rhythm	12	0.8000	0.6667	0.7273	None	952	0.6607	0.8163	0.7303
Photic driving	51	0.8750	0.6863	0.7692	**Location**	692	0.7450	0.5958	0.6618
PLEDs	20	0.5625	0.4500	0.5000	Frontal	189	0.7302	0.4868	0.5841
Epileptiform Discharge	136	0.7934	0.7059	0.7471	Occipital	268	0.7871	0.7313	0.7582
Disorganization	98	0.8000	0.6122	0.6936	Temporal	128	0.6786	0.5938	0.6333
Unspecified	59	0.3855	0.5424	0.4507	Central	30	0.7500	0.4109	0.5309
Hemisphere	1438	–	–	0.8148	Parietal	10	0.5714	0.4000	0.4706
Right	96	0.7634	0.7396	0.7513	Frontocentral	50	0.8824	0.6000	0.7143
Left	159	0.8257	0.5660	0.6716	Frontotemporal	17	0.5618	0.4963	0.5270
Both	246	0.6027	0.5488	0.5745	**Modality**	1438	–	–	0.9337
N/A	937	0.8603	0.9218	0.8900	Factual	1366	0.9696	0.9939	0.9816
Dispersal	1438	–	–	0.7283	Possible	70	0.4308	0.4000	0.4148
Localized	330	0.5955	0.6424	0.6181	Proposed	76	0.2318	0.4605	0.3084
Generalized	246	0.5162	0.5813	0.5468	**Polarity**	1438	0.9100	0.7389	0.8156
N/A	862	0.8441	0.7947	0.8186					

omitted for multi-class classification tasks since they are equivalent to F_1. Table 2.9 presents a comparison of the DRN against a Support Vector Machine (Vapnik, 1999) trained on the same features.

In general, it is clear that the DRN was able to accurately determine the attributes of EEG, obtaining an overall macro-averaged F_1 score of 0.7133. While the DRN out performs the SVM in micro-averaged F_1 score, it is also clear that the DRN struggles to predict certain attribute values, for example MODALITY=Possible, MORPHOLOGY=Polyspike_and_wave,

Table 2.9. Deep ReLU Network compared with Support Vector Machine for detecting attributes of medical concepts in EEG Reports measured using micro-average F_1 score.

Attribute	SVM	DRN
Morphology	0.7521	**0.7716**
Hemisphere	0.7897	**0.8148**
Dispersal	0.6734	**0.7283**
Frequency Band	0.8506	**0.8754**
Magnitude	0.8062	**0.8148**
Background	0.8483	**0.8543**
Recurrence	**0.7380**	0.7174
Location	**0.6741**	0.6618
Modality	0.7275	**0.7500**
Polarity	0.7384	**0.7852**
Concept Type	0.9109	**0.9392**

and BRAIN_LOCATION=Central. The degraded performance for these values is unsurprising as they are some of the least frequently annotated attributes in our data set (with 41, 5, 29 instances respectively). However, we believe that the performance of our model when detecting rare attributes could be improved in future work by incorporating knowledge from neurological ontologies (Sahoo et al., 2014) as well as other sources of general medical knowledge (Lindberg et al., 1993). We found that the performance of our DRN for determining attribute of other medical concepts was highly promising, with an overall accuracy of 97.4%. However, we observed the same correlation between the number of annotations for an attribute's value and the DRN's ability to predict that value. In the TUH EEG corpus, we found that nearly all mentions of EEG Events and medical problems, test, or treatments had a factual modality (96%). The lowest performance of the DRN was observed when determining the polarity attribute. The main source of errors for determining polarity was due to frequent ungrammatical sentences in the EEG Reports, e.g., "*There are rare sharp transients noted in the record but without after going slow waves as would be expected in epileptiform sharp waves*". We believe these errors could be overcome in future work by relying on parsers trained on medical data. The medical concepts and their attributes extracted from EEG

Table 2.10. Annotation distribution for concepts and assertions in the 2010 i2b2/VA challenge. The AwSE assertion represents an assertion value of "Associated with Someone Else".

	Concepts					
	Problem	Test	Treatment			
Training Set	11,968	7,369	8,500			
Test Set	18,550	12,899	13,560			
	Assertions					
	Present	Absent	Possible	Conditional	Hypothetical	AwSE
Training Set	8,052	2,535	535	103	651	92
Test Set	13,025	3,609	883	171	717	145

reports it enables us to generate EEG-specific qualified medical knowledge (Goodwin and Harabagiu, 2013). We believe this knowledge can be enhanced by incorporating information from the EEG signals, creating a multi-modal medical knowledge representation. Such a knowledge representation is needed for reasoning mechanisms operating on big medical data.

2.7.2 Evaluation of Deep Learning Systems for Identifying Concepts and Their Attributes in Discharge Summaries

In this subsection, we evaluate the ability of the MT-BGCN system to detect medical concepts and their attributes in discharge summaries. To evaluate MT-BGCN, we use the 2010 i2b2/VA Challenge dataset described in Section 2.3 comprised of 349 discharge summaries for training and 477 for testing. The discharge summaries have manually annotated spans of medical concept mentions including medical problems, tests, and treatments. Moreover, each medical problem has an *assertion* associated with it taking one of six values defined in Section 2.3. Table 2.10 describes the distribution of annotations in the dataset.

For boundary detection MT-BGCN is compared against several baselines:

- **Semi-Markov HMM** (De Bruijn et al., 2011) is the system that won the 2010 i2b2/VA challenge. It relies on a Semi-Markov HMM operating on feature vectors.

- **MT-Seq2Seq** is a Multi-task Sequence-to-Sequence model of (Bhatia et al., 2019) trained to jointly perform concept detection and negation.

- **ELMo+BiLSTM-CRF** is a model from Zhu et al. (2018) using contextualized ELMo embeddings (Peters et al., 2018) fed into a BiLSTM-CRF. This is equivalent to MT-BGCN with the BERT sentence encoder replaced by ELMo and the GCN module removed. The ELMo model used in this system was pre-trained first on general purpose text, then on medical text, similar to the BERT model in MT-BGCN.

- **BERT** is a fine-tuned BERT model (Devlin et al., 2019) feeding directly into a single softmax layer for prediction. This was reported in Fraser et al. (2019).

- **MIMIC-BERT-large** is a model developed by Si et al. (2019) that takes the pre-trained BERT-large model, conducts a second pre-training regimen using clinical notes in MIMIC, then fine-tunes it for the 2010 i2b2/VA challenge using a BiLSTM. The BERT-large model contains roughly three times as many parameters as the BERT-base model used in MT-BGCN and the rest of the baselines.

- **NCBI-BERT** is a model developed by Fraser et al. (2019) that uses two pre-trained BERT models concatenated together to form its contextualized word embeddings. The first BERT model is the BERT-base model trained on Wikipedia and the BooksCorpus and the second BERT model is the same pre-trained BERT model as MT-BGCN. The concatenated token embeddings are passed to a BiLSTM as in MT-BGCN.

MT-BGCN is implemented in TensorFlow (Abadi et al., 2016) and trained using the entire training set for 50 epochs using a learning rate of 1e-6 with exponential decay with rate of 0.95. The PoS embedding size is 256, the GCN and BiLSTM hidden size is 200. A dropout rate of 0.4 is used in all layers other than the BiLSTM in which no dropout is used. λ_1, λ_2,

Table 2.11. Evaluation results for medical concept detection for the 2010 i2b2/VA challenge. Best results are in bold. *Results marked with an asterisk were reported with only two significant figures. The developers of MIMIC-BERT-large only report F_1 score.

System	Precision	Recall	F_1-Score
Semi-Markov HMM	0.869	0.836	0.852
MT-Seq2Seq	0.854	0.858	0.855
ELMo+BiLSTM-CRF	0.893	0.879	0.886
BERT	0.85*	0.87*	0.86*
MIMIC-BERT-large	–	–	0.903
NCBI-BERT	0.89*	0.90*	0.90*
MT-BGCN	**0.901**	**0.917**	**0.909**

λ_3, and λ_4 are set to 0.1, 0.6, 0.3, and 0.7, respectively. Hyper-parameters are tuned using a reserved set of 10% of the training data.

As in the previous subsection, boundary detection is evaluated in terms of precision, recall, and F_1. However we only report exact matching scores since partial matching metrics are only available for one baseline. The results are presented in Table 2.11. MT-BGCN achieves to top score in each metric. We can see from the baselines that contextual embedding is important for this task, since the ELMo and BERT models all outperform the MT-Seq2Seq model. However, the results also indicate that it is important to adapt the contextual embedding system to medical data before fine-tuning, as MT-BGCN, ELMo+BiLSTM-CRF, MIMIC-BERT-large, and NCBI-BERT all outperform the general purpose BERT model. Most importantly, the results indicate that the representation layer is not the only important part of the model and that syntactic information is beneficial. The representation layers of NCBI-BERT and MIMIC-BERT-large contain roughly 2x and 3x more parameters than the BERT sentence encoder of MT-BGCN, yet MT-BGCN outperforms them both. Moreover, MT-BGCN jointly performs assertion classification in addition to medical concept boundary detection, soundly outperforming the other multi-task network, MT-Seq2Seq.

For assertion classification, MT-BGCN is compared against the best performing system from the 2010 i2b2/VA challenge as well as MT-Seq2Seq. However, MT-Seq2Seq only per-

Table 2.12. Evaluation results for assertion classification. The AwSE assertion value represents "Associated with Someone Else". Best results are in bold. MT-Seq2Seq model is trained to only classify the "Absent" assertion.

	Present			Possible			Absent		
	P	R	F_1	P	R	F_1	P	R	F_1
SVM	0.944	**0.980**	0.962	**0.816**	0.589	0.684	0.959	0.934	0.947
MT-BGCN	**0.970**	0.974	**0.972**	0.740	**0.818**	**0.777**	**0.968**	**0.969**	**0.969**
MT-Seq2Seq	–	–	–	–	–	–	0.919	0.891	0.905
	Conditional			Hypothetical			AwSE		
	P	R	F_1	P	R	F_1	P	R	F_1
SVM	0.729	0.298	0.423	0.922	0.870	0.895	0.915	**0.812**	**0.861**
MT-BGCN	**0.759**	**0.386**	**0.512**	**0.940**	**0.874**	**0.906**	**0.934**	0.779	0.850
	Micro-Average								
	F_1								
SVM	0.939								
MT-BGCN	**0.955**								

forms negation detection – identifying only those medical problems whose assertions have the "absent" value. The top system for assertion classification from the 2010 i2b2/VA challenge was a Support Vector Machine (SVM) model from Roberts and Harabagiu (2011).

The results for assertion classification are presented in Table 2.12. MT-BGCN attains top performance in F_1 score for five out of six assertion classes. MT-BGCN is out-performed by the SVM model in recall and F_1 for the "Associated with Someone Else" assertion value which is, unsurprisingly, the assertion value with the least amount of training data, as per Table 2.10. MT-BGCN also out-performs the MT-Seq2Seq model in negation detection by a wide margin.

By advancing the state-of-the-art in both medical concept detection and assertion classification on the 2010 i2b2/VA challenge, MT-BGCN shows the importance of pre-trained contextualized embedding methods and incorporating syntactic information using graph convolution. Moreover, MT-BGCN demonstrates the efficacy of multi-task learning by achieving top results in both tasks with the same network. However, MT-BGCN is still prone to certain

classes of errors. In spite of the incorporation of syntactic information in the form of PoS tags and the dependency parse, MT-BGCN still makes mistakes identifying phrasal boundaries. When the boundary detection evaluation is relaxed to allow for partial matching, the F_1 score increased by 0.4 points to 0.949. This could be due to errors in PoS tagging and dependency parsing by SciSpacy. In future work, it may be beneficial to allow the network to model this information implicitly in the form of auxiliary training tasks. This way, the network will develop its own internal representations of syntactic information and not reliant on third party tools. Another class of errors pertains to falsely predicted medical problems. For example, MT-BGCN falsely identifies phrases like "*somewhat uncontrolled*" and "*acute onset*" as medical problems. These phrases are indeed indicative of medical problems, but on their own do not indicate a specific problem. In future work, incorporating medical knowledge like that found in biomedical ontologies could prove to be beneficial since it could allow the model to better characterize medical concepts. Biomedical knowledge would also help the model determine the types of medical concepts it identifies. For example, the model identifies the span "*a net fluid balance*" as a medical concept, but erroneously assigns it the type of treatment instead of test. Biomedical ontologies arrange medical concepts in ontologies with type information explicitly encoded.

2.8 Summary Lessons Learned

In this chapter we described medical concept detection and attribute classification, focusing on two datasets of clinical free-text documents. Both EEG reports and discharge summaries contain important medical concepts characterized by attributes that require identification to inform downstream tasks. In particular both EHR datasets contain mentions of medical problems, tests, and treatments that help paint a clinical picture for the patient described in the record. Moreover, EEG reports are characterized by two EEG-specific concepts: EEG activities and EEG events. EEG activities are particularly complex and as such are described

by intricate and specialized language. Therefore, characterizing EEG activities requires a schema of eighteen EEG Activity attributes that are defined in the chapter.

We presented three deep learning architectures for performing medical concept detection and attribute classification. The stacked Long Short-Term Memory network (sLSTM) was shown to be capable of identifying medical concepts in EEG reports while the Deep ReLU Network (DRN) was able to classify a complex set of attributes for EEG activities and the other medical concepts that occur in EEG reports. Together, the sLSTM and DRN show that deep learning can be effectively used to process EEG reports. Likewise the Multi-Task BERT Graph Convolution Network (MT-BGCN) was introduced for performing both medical concept detection and attribute classification in discharge summaries with the same model. MT-BGCN relies on the pre-trained contextualized embedding model, BERT to represent text from discharge summaries. Moreover, it incorporates syntactic information in the form of graph convolution on the dependency graph of a sentence to improve prediction. MT-BGCN advances the state-of-the-art in concept detection and assertion classification in the 2010 i2b2/VA challenge.

CHAPTER 3

IDENTIFYING RELATIONS BETWEEN MEDICAL CONCEPTS IN EHRS

In this chapter[1][2] deep learning methods are proposed for identifying relations between medical concepts in EHRs. In order to leverage big medical data increasingly available in large collections of EHRs, medical concept detection and attribute classification methods like those described in Chapter 2 provide a useful first step. However, the information encoded in medical narratives is not limited to concepts and their attributes. *Relations* between medical concepts encode important information that is also widely useful in many of the same medical informatics tasks as concept detection, including patient cohort retrieval (Hersh, 2008), relation extraction (Uzuner et al., 2011), clinical decision support (Demner-Fushman et al., 2009), medical question answering (Goodwin, 2018), and knowledge discovery (Maldonado et al., 2017). Moreover, accurately identified relations between medical concepts in clinical narratives can help drive secondary use of EHR data for clinical and translational research (Wang et al., 2018). A relation between a pair of medical concepts can encode many forms of clinically relevant information, e.g., a drug is a treatment for a medical problem, one medical problem causes another, or a physiological structure is located in an anatomical region.

This chapter address the task of extracting relations from EEG reports. As described in Chapter 2, EEG reports are clinical documents generated by neurologists during an EEG exam that document the EEG signal. Since inter-observer agreement is known to be moderate (Beniczky et al., 2013), we seek to develop methods to aid in the interpretation of the EEG signal. Patient cohort retrieval can help improve EEG signal interpretation by providing neurologists with the results of search from a vast archive of EEG reports. The medical

[1]©2019 Elsevier. Reprinted, with permission, from Ramon Maldonado and Sanda M. Harabagiu, *Active Deep Learning for the Identification of Concepts and Relations in Electroencephalography Reports*. Journal of Biomedical Informatics, Vol. 98 (2019): 103265.

[2]This chapter contains excerpts from Maldonado et al. (2018).

concepts and attributes identified in the TUH-EEG corpus by methods like those introduced in Chapter 2 can help inform cohort retrieval systems, such as MERCuRY (Goodwin and Harabagiu, 2016). However, identifying relations between medical concepts can also provide important information to cohort retrieval systems. Consider the example record from Section 2.2, reproduced in Figure 3.1. In Section 2.2, we saw that this record can be deemed

CLINICAL HISTORY: This is a 55-year-old gentleman with [right leg swelling]$_{PROB}$, [ESRD]$_{PROB}$, [history of seizures]$_{PROB}$, and [hip fracture]$_{PROB}$.
MEDICATIONS: [Dilantin]$_{TR}$, [Haldol]$_{TR}$, many others.
INTRODUCTION: [Digital video EEG]$_{TEST}$ is performed at the bedside using standard 10-20 system of electrode placement with one channel of [EKG]$_{TEST}$. The patient is described as drowsy.
DESCRIPTION OF THE RECORD: The background EEG is diffusely [slow]$_{ACT}$ with primarily rhythmic [theta frequency activity]$_{ACT}$ of 5 to 7 Hz. There are frontally predominant, relatively synchronous [triphasic waves]$_{ACT}$ seen throughout the record. On one occasion, there may be some [asymmetries]$_{ACT}$, somewhat more remarkable on the left than the right. [Stimulation]$_{EV}$ of the patient produces cessation of the [triphasic waves]$_{ACT}$.
IMPRESSION: Abnormal EEG due to:
1. Generalized background [slowing]$_{ACT}$.
2. [Triphasic waves]$_{ACT}$.
CLINICAL CORRELATION: No [seizures]$_{PROB}$ were recorded. The [triphasic waves]$_{ACT}$ are typically a manifestation of underlying [metabolic encephalopathy]$_{PROB}$ including [hepatic encephalopathy]$_{PROB}$, [renal insufficiency]$_{PROB}$, or medication exposure. The [asymmetry]$_{ACT}$ of the triphasic waves, with prominence on the left, may be due to preexisting [history of epilepsy]$_{PROB}$ and/or [structural brain disease]$_{PROB}$. No [previous EEGs]$_{TEST}$ were available for comparison.

Figure 3.1. Synthetic Example EEG Report.

relevant to an example query *Q: Patients with triphasic waves suspected of encephalopathy* by identifying the medical concepts in the query and throughout the corpus of EEG reports.

However, the identification of the medical concepts from the query and in the EEG reports is not sufficient, as many false positives can be produced. For example, the query Q does not only ask about the concepts it mentions, but it also implicitly asks about the relation between the concepts $[triphasic\ waves]_{ACT}$ and $[encephalopathy]_{PROB}$. The EEG report illustrated in Figure 3.1 contains two relations between the medical concepts mentioned in Q, namely: (R1):$[triphasic\ waves]_{ACT}$ $-$EVIDENCES$\rightarrow$ $[metabolic\_\ encephalopathy]_{PROB}$; (R2):$[triphasic\ waves]_{ACT}$ $-$EVIDENCES$\rightarrow$ $[hepatic\ encephalopathy]_{PROB}$. Therefore, the EEG report is

judged relevant to the query Q. However, if only the query concepts would be considered to infer relevance of an EEG report, the EEG report illustrated in Figure 3.1 would be deemed relevant also for the query *Q': Patients with theta waves suspected of encephalopathy.* Both concepts from *Q'* are mentioned in the EEG report, but no relation between those concepts can be inferred from the EEG report, which correctly indicates that it should not be judged as relevant to *Q'*.

Two deep learning methods for identifying relations in EEG reports are presented in this chapter: (1) **EEG-RelNet** and (2) the **Self-Attention Concept, Attribute, Relation (SACAR)** identifier. EEG-RelNet is a recurrent neural model that processes entire EEG reports, one sentence at a time, in order to identify potentially long-distance relations. SACAR is a multi-task model that is trained to jointly extract concepts, their attributes and relations between them.

Relations between medical concepts in EEG reports can span sentences and even sections. Therefore, both EEG-RelNet and SACAR operate at the report level, identifying relations between concepts mentioned anywhere in an EEG report. EEG-RelNet reads through an EEG report, recurrently updating a set of *memory vectors* that store information about medical concepts and potetntial relations between them. SACAR relies on a powerful encoding mechanism based on the Universal Transformer neural architecture (Dehghani et al., 2018) to derive a shared representation of the text of an EEG report. This shared representation is then fed to a series of prediction modules that perform concept detection, attribute classification, and relation extraction. EEG-RelNet is used only for relation extraction, requiring previously identified medical concepts with attributes. In contrast, SACAR is an end-to-end model that does not require pre-identification of concepts or attributes.

This chapter is organized as follows: Section 3.1 provides background for medical relation extraction from EEG reports; Section 3.2 describes the relations of interest in EEG reports; Section 3.3 presents the EEG-RelNet model; Section 3.4 presents the SACAR model; Sec-

tion 3.5 presents experimental evaluations and discussion; and Section 3.6 concludes the chapter.

3.1 Background

Clinical relation extraction is the task of identifying relations between medical concepts in clinical narratives. Clinical relation extraction along with medical concept detection and attribute classification, discussed in Chapter 2, comprise the field known as *Clinical information extraction (IE)* (Wang et al., 2018). The 2010 i2b2/VA challenge (Uzuner et al., 2011) has proven to be a useful benchmark dataset for research in clinical IE since it provides a relation extraction task in addition to the concept detection and assertion classification tasks discussed in Chapter 2. The 2010 i2b2/VA challenge includes relations from the following three categories: medical problem−treatment (TrP) relations, medical problem−test (TeP) relations, and medical problem−medical problem (PP) relations.

The winner of the 2010 i2b2/VA challenge proposed a Support Vector Machine model operating on sentence-level features (Rink et al., 2011). Later, D'Souza and Ng (2014) introduced an ensemble-based method operating in an ILP framework leveraging human-supplied knowledge. Neural methods have shown promise for this task including Convolutional Neural Networks (Luo et al., 2017), and BiLSTM-CRF (Li et al., 2019). More recently, massively pre-trained contextualized embedding models have been employed to advance the state-of-the-art further still (Peng et al., 2019).

Although it is now well established that automated extraction of knowledge from clinical notes involves accurately identifying not only the medical concepts, but also the various relationships in which they are involved (Cimino, 1998), the automatic identification of relations between medical concepts in EEG reports is hindered by two major obstacles. First, the *types* of relations between medical concepts in EEG reports are not well represented in existing clinical IE corpora. As illustrated in the exemplified EEG report, TrP from the

2010 i2b2/VA challenge relations would be useful, but other types of relations relevant for the knowledge expressed in the report would be missed. The second hurdle arises from the constraint that only relations between medical concepts observed in the same sentence can be identified with methods produced for the 2010 i2b2/VA challenge, even those using deep learning methods capable of processing large corpora of clinical documents (Luo, 2017). We found the solution of both these limitations by employing neural models that operate at the report-level instead of the sentence-level. By considering entire EEG reports, the systems are able to model interactions between medical concepts across sentences, from anywhere in the report. First, we consider and extend RelNet (Bansal et al., 2017), a memory-augmented neural network in which medical concepts can be processed in abstract memory cells while relations between medical concepts are processed in separate relation memory cells. The memories implicitly model the *current knowledge* about medical concepts and the relations they share. We also consider and extend Bi-Affine Relation Attention Networks (Verga et al., 2018) which simultaneously predicts relations between all concept mention pairs in a document using a Transformer encoder (Vaswani et al., 2017) and a Bi-Affine prediction layer.

3.2 Relations Between Medical Concepts in EEG Reports

This section describes the relations between medical concepts that can be extracted from the narrative of EEG reports. In the rest of this chapter, we focus on the publicly available corpus of EEG reports from Temple University Hospital, the TUH-EEG corpus, described in Section 2.2.

In this chapter, we focus on four binary relations between medical concepts: (1) EVIDENCES; (2) EVOKES; (3) CLINICAL-CORRELATION; and (4) TREATMENT-FOR. The decision to focus on these four relations was motivated by discussions with practicing neurologists, as the relations represent implicit knowledge gleaned from the EEG reports which informs

their reading of the Impression and Clinical Correlation sections of the EEG report. The EVIDENCES relation considers EEG activities or medical problems as providing evidence for medical problems mentioned in the EEG report. For example, the excerpt "*The triphasic waves are typically a manifestation of underlying metabolic encephalopathy*" from the example report in Figure 3.1 indicates the following EVIDENCES relation: $[triphasic\ waves]_{ACT}$ $-$EVIDENCES$\rightarrow$ $[metabolic\_\ encephalopathy]_{PROB}$. The EVOKES relation represents the relationship where a medical concept evokes an EEG activity. EEG events, medical problems and treatments can all evoke EEG activities. For example, the excerpt "*Stimulation of the patient produces cessation of the triphasic waves*" indicates the following EVOKES relation: $[stimulation]_{EV}$ $-$EVOKES$\rightarrow$ $[triphasic\ waves]_{ACT}$. It should be noted that the polarity attribute for this triphasic waves entity is negative.

The CLINICAL-CORRELATION relation connects the EEG activities and medical problems mentioned in the Clinical Correlation section of the EEG report if the activity *clinically correlates* with the medical problem. Examples of CLINICAL-CORRELATION relations in the example record include: $[triphasic\ waves]_{ACT}$ $-$CLINICAL-CORRELATION$\rightarrow$ $[metabolic\ encephalopathy]_{PROB}$, $[triphasic\ waves]_{ACT}$ $-$CLINICAL-CORRELATION$\rightarrow$ $[renal\ insufficiency]_{PROB}$, and $[asymmetry]_{ACT}$ $-$CLINICAL-CORRELATION$\rightarrow$ $[structural\ brain\ disease]_{PROB}$. The TREATMENT-FOR relation links treatments to the medical problems for which they are prescribed. A common pattern for TREATMENT-FOR relations is to link a medication from the "*MEDICATIONS*" section to a medical problem from the "*CLINICAL HISTORY*" section, e.g., $[Dilantin]_{TR}$ $-$TREATMENT-FOR$\rightarrow$ $[history\ of\ seizures]_{PROB}$. It should be noted that this relation spans across sentence boundaries. While this phenomenon is common for TREATMENT-FOR relations, it is not limited to TREATMENT-FOR relations as arguments of any of the four relation types can cross sentences and even sections.

Because in the same EEG report, the same entity corresponding to a unique medical concept may be mentioned several times, we distinguish between *concept mentions* and

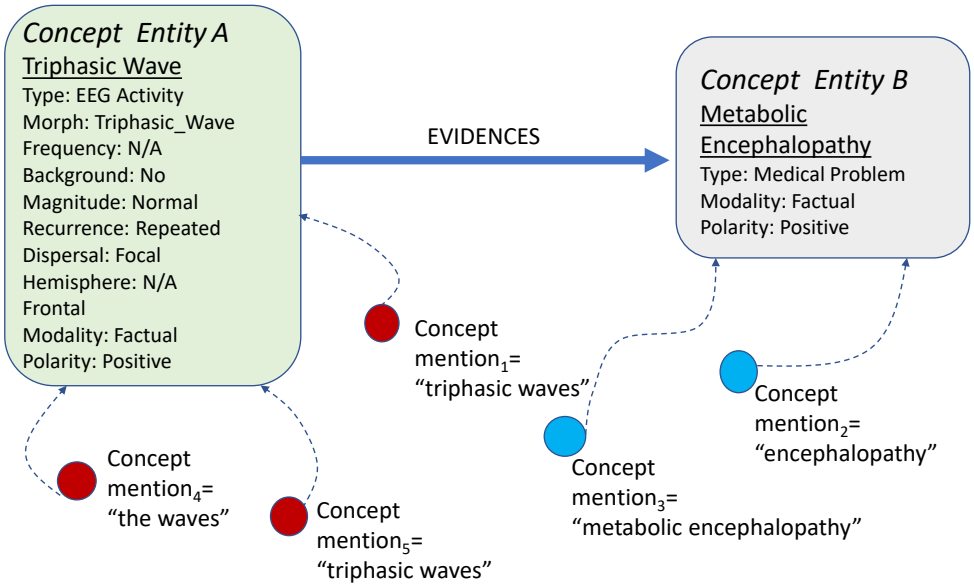

Figure 3.2. Concept entities, their mentions and possible relations between them.

concept entities. Figure 3.2 illustrates two concept entities and their corresponding concept mentions. As it can be seen, the concept entities are illustrated through (1) their normalized name; (2) their type; and (3) their identified attributes. We normalized each concept mention into a canonical form (referred to as *normalized name*) using the (i) the morphology attribute for EEG activities and (ii) the United Medical Language System (UMLS) (Lindberg et al., 1993) *preferred name* of the concepts of other types. To identify the concept entities, we assumed that concept mentions from the same EEG report that (1) have the same normalized name, (2) the same type, and (3) the same values of their attributes *co-refer* to the same concept entity. We defined relations between concept entities, not between concept mentions. The remainder of this chapter details two neural networks for detecting relations between concept entities in EEG reports.

3.3 Memory-Augmented Deep Learning for Recognizing Long-Distance Relations in EEG Reports

The automatic identification of relations between pairs of medical concepts in EEG reports, regardless of their presence in the same sentence or sentence of the report, has been made possible by EEG-RelNet. The EEG-RelNet deep learning architecture provides end-to-end detection of relations between medical concepts in EEG reports by using a neural network augmented with two types of memory cells: (i) a memory for each medical concept mentioned in the report; and (ii) a memory for each relation between each pair of medical concepts mentioned in the report. Moreover, the relational memory is dynamic as it changes to model the specific concepts and relations observed in each EEG report. By processing entire reports at once, EEG-RelNet is able to model both inter- and intra-sentential relations with the same network.

Given a corpus of EEG reports with pre-identified medical concepts (EEG activities, EEG events, medical problems, tests and treatments), EEG-RelNet provides inference of the EVIDENCES, EVOKES, CLINICAL-CORRELATION, and TREATMENT-FOR relations between pairs of such concepts. Inspired by RelNet, a model reported by Bansal et al. (2017), we developed the EEG-RelNet, a deep neural network architecture that operates on the *full text* of an EEG report considering all medical concepts identified in the report to detect relations. More specifically, given the full text of an EEG report and the set of medical concepts identified in that report, EEG-RelNet can predict whether there is relation, $\hat{R}_{ij}$, of type t between any pair of medical concepts c_i and c_j recognized in the report. To do so, EEG-RelNet processes the EEG report, one sentence at a time, reading its words, encoding the information from the sentence, processing the sentence information in the dynamic relational memory, and predicting each type of relation based on the dynamic memories after they have processed each sentence in the EEG report. EEG-RelNet is comprised of three modules:

- the **Input Encoding Module** which *encodes* information from the report in concept- and sentence-level embedding vectors, which are used throughout the deep learning architecture;
- the **Dynamic Relational Memory Module** which maintains and updates a set of hidden states called *memories* to capture accumulated information about each medical concept and potential relation in the EEG report;
- the **Output Module** which uses the updated memories to determine the most likely relations (and their types) between medical concepts in the EEG report.

In the remainder of this section, we provide a detailed description of each module of EEG-RelNet.

3.3.1 The Input Encoding Module

The role of this module is to learn (1) an embedding encoding each medical concept as well as each of its attributes and (2) an embedding encoding the information from each sentence in the EEG report. Formally, we represent an EEG report as a set of medical concepts, $C = \{c_1, \cdots, c_d\}$, and a sequence of sentences, $[s_1, \cdots, s_n]$. Each medical concept, c_i, is associated with several N-dimensional vectors called embeddings: (a) an embedding for the normalized concept name, $\mathbf{\tilde{c}_i} \in \mathbb{R}^N$ and (b) separate embeddings for each of its A attributes values $\{\mathbf{a_1^{c_i}}, \cdots, \mathbf{a_A^{c_i}}\} \in \mathbb{R}^N$. Thus, the embedding $\vec{\boldsymbol{c_i}}$ for a medical concept is created by (1) concatenating the embedding for the name of the medical concept with the embedding for each of its attributes and (2) projecting this concatenated vector using a learned weight matrix $W_C \in \mathbb{R}^{N \times N(|A|+1)}$, i.e., $\vec{\boldsymbol{c_i}} = W_C \times [\mathbf{\tilde{c}_i}, \mathbf{a_1^{c_i}}, \cdots, \mathbf{a_A^{c_i}}]$. In this way, each medical concept is represented by an embedding, $\vec{\boldsymbol{c_i}} \in \mathbb{R}^N$.

Participation of medical concepts in relations is informed by the context of each concept in the text of the EEG report. Contextual information is provided by the words of the sentence where the concept is mentioned, hence a representation of words from each sentence as is also

desirable. Therefore, we learn an embedding e_i for each word w_i in a sentence, enabling us to represent each sentence as a sequence of embeddings $E = [\mathbf{e_1}, \cdots, \mathbf{e_m}]$ such that the elements of E occur in the same order as the words from the sentence [3]. We use the pre-trained word embeddings provided by GloVe (Pennington et al., 2014). While the traditional choice for combining and composing the embeddings in E into a single sentence embedding would be a Recurrent Neural Network (RNN), we instead adopt a more recent and significantly more efficient strategy, namely a *learned positional mask* (Bansal et al., 2017; Sukhbaatar et al., 2015). The k-th sentence from the EEG report is represented as: $\vec{\boldsymbol{s_k}} = \sum_i^m \mathbf{f_i} \odot \boldsymbol{e_i}$ where $\mathbf{f_i}$ is the learned positional mask for word i and $\odot$ is the element-wise product. Given that the sentence had m words, the vectors $[\mathbf{f_1}, \cdots, \mathbf{f_m}]$, represent the learned positional mask for the entire sentence. It is important to note that the same vectors $[\mathbf{f_1}, \cdots, \mathbf{f_m}]$ are used when each new sentence is encoded and they are learned jointly with the other parameters of the deep learning model.

3.3.2 The Dynamic Relational Memory Module

Because EEG reports often contain long-distance relations between concepts we relied on a Dynamic Relational Memory (Bansal et al., 2017) (DRM) Module to keep track of the interactions between medical concepts in each report. The DRM, depicted in Figure 3.3, accumulates information about medical concepts and the relations between them by processing each sentence encoded by the Input Module and recurrently updating a set of hidden states, called *memories*. The DRM maintains one *concept memory*, $\vec{\boldsymbol{h_i}}$, for each medical concept, c_i, mentioned in the EEG report being processed and one *relation memory*, $\vec{\boldsymbol{r_{ij}}}$, for each pair of concepts, (c_i, c_j), mentioned. Specifically, given a set of d concept embeddings $\{\vec{\boldsymbol{c_1}}, \cdots, \vec{\boldsymbol{c_d}}\}$, one for each concept mentioned in the report, the DRM constructs a set of d

[3]Embeddings, e_w, corresponding to words contained within a concept mention, c_i are replaced with the embedding for that concept instead of the word, i.e., $e_w = \vec{\boldsymbol{c_i}}$. This is required to enable the Key-Value memory structures described in the next subsection

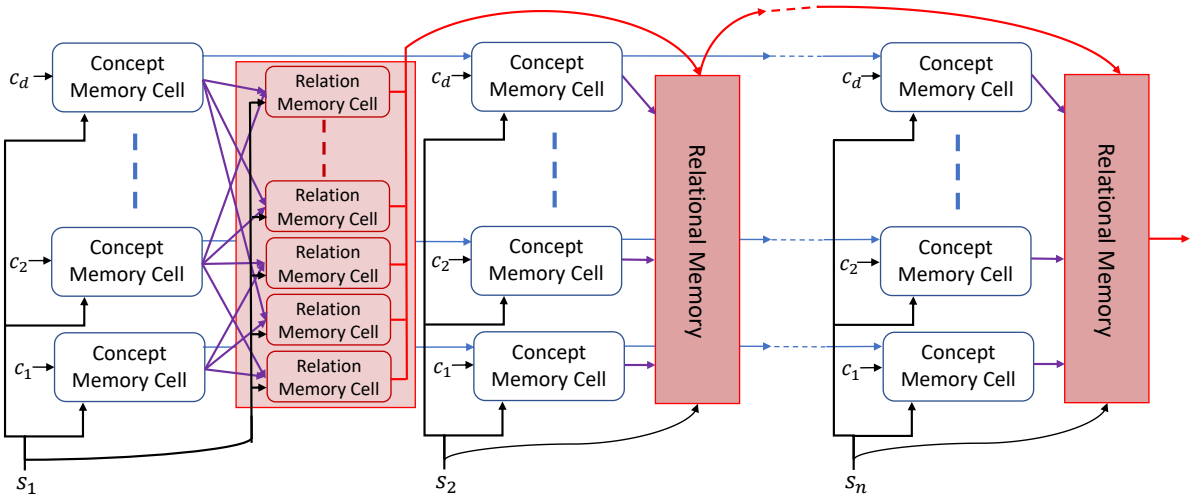

Figure 3.3. The Dynamic Relational Memory Module of EEG-RelNet. The Dynamic Relational Memory Module processes n sentences, updating a set of d Concept Memories and $d\,(d-1)$ Relation Memories for each sentence.

Concept Memory cells, maintaining a set of d concept memories, $\{\vec{\boldsymbol{h_1}}, \cdots, \vec{\boldsymbol{h_d}}\}$. Likewise, the DRM constructs a set of $d \times (d-1)$ Relation Memory cells maintaining a set of relation memories, $\{\vec{\boldsymbol{r_{ij}}} | \forall i, j \in [1, ..., d], i \neq j\}$. The *Dynamic Relational Memory* consists of the entire set of concept and relation memories in an EEG report. This set of memories is recurrently updated for each sentence in the report. Given a sentence embedding, $\vec{\boldsymbol{s_k}}$, the DRM sends that sentence embedding to each Concept Memory cell to determine if that sentence is relevant to that concept. If it is deemed relevant, the Concept and Relation Memory cells update the concept and relation memories associated with that concept. The shared concept and relation memories are recurrently updated by their respective cells for each sentence in the EEG report. In this way, after processing each sentence in the EEG report, the concept and memory cells will contain information from the entire report that can be used to identify relations between concepts.

The Concept Memories are organized as a Key-Value Memory Network (Miller et al., 2016). Key-Value memory networks function in two stages: the lookup (addressing) stage is based on a *key* vector while the reading stage (giving the returned result) returns the *value*

vector. Consequently, in EEG-RelNet, memory vectors are tied to so-called key vectors enabling the model to selectively update a memory vector when the input sentence has context that is *relevant* to the memory's associated key vector. Henaff et al. (2016) have shown that when concept embeddings are used as key vectors, the associated memory vectors will accumulate information about those concepts. Consequently, in EEG-RelNet, concept embeddings are used as key vectors allowing the network to selectively update each Concept Memory, $\mathbf{h_i}$, if an input sentence is relevant to the concept, $\mathbf{c_i}$. The Concept Memory Cell, illustrated in Figure 3.4, is used to update a Concept Memory, $\mathbf{h_i}$, given a medical concept embedding, $\mathbf{c_i}$, and a sentence encoding, $\vec{\boldsymbol{s_k}}$, via the following equations:

$$g_i^c = \sigma\left(\langle \vec{\boldsymbol{s_k}}, \mathbf{h_i} + \mathbf{c_i} \rangle\right) \tag{3.1}$$

$$\tilde{\mathbf{h}}_\mathbf{i} = \phi\left(W_u \mathbf{h_i} + W_v \mathbf{c_i} + W_s \vec{\boldsymbol{s_k}}\right) \tag{3.2}$$

$$\mathbf{h_i} \leftarrow \mathbf{h_i} + g_i^c \odot \tilde{\mathbf{h}}_\mathbf{i} \tag{3.3}$$

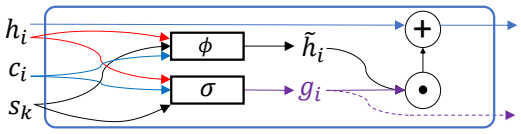

Figure 3.4. Concept Memory Cell

where W_u, W_v and W_s are trainable weight matrices in $\mathbb{R}^{N \times N}$, $\langle \cdot, \cdot \rangle$ is the inner product, σ is the sigmoid function and ϕ is a Parametric Rectified Linear Unit (PReLU) (He et al., 2015). Equation 3.1 is a *gating* function that determines how much the kth input sentence affects the ith Concept Memory such that $g_i^c \in [0, 1]$ values close to 1 indicate sentence s_k is relevant to medical concept c_i and values close to 0 indicate the opposite. Equation 3.2 defines the *candidate* Concept Memory that will be used to update the existing Concept Memory, $\mathbf{h_i}$, after it is scaled by g_i^c as shown in equation 3.3.

As illustrated in Figure 3.3, when each sentence s_i is processed, the DRM uses and updates not only concept memories, but also a much larger set of relation memories. This is explained by the fact that maintaining a single memory vector for each concept is not sufficient for modeling concepts that participate in multiple relations, especially when those relations involve concepts that are mentioned at significant distance in the EEG report. Thus, to model the interactions each concept has with each other concept in the same EEG

report, we maintain a set of Relation Memories corresponding to each pair of concepts from the EEG report, $\{\mathbf{r_{ij}} : i, j \in C, i \neq j\}$, where C is the set of medical concepts in the EEG report. Each Relation Memory is updated using the Relation Memory Cell illustrated in Figure 3.5 via the following equations:

$$g_{ij}^r = g_i^c g_j^c \sigma\left(\langle \vec{\boldsymbol{s_k}}, \mathbf{r_{ij}} \rangle\right) \tag{3.4}$$

$$\tilde{\mathbf{r}}_{\mathbf{ij}} = \phi\left(W_A \mathbf{r_{ij}} + W_B \vec{\boldsymbol{s_k}}\right) \tag{3.5}$$

$$\mathbf{r_{ij}} \leftarrow \mathbf{r_{ij}} + g_{ij}^r \odot \tilde{\mathbf{r}}_{\mathbf{ij}} \tag{3.6}$$

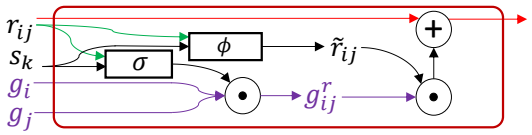

Figure 3.5. Relation Memory Cell

where W_A and W_B are trainable weight matrices in $\mathbb{R}^{N \times N}$. As in the Concept Memory Cell, the Relation Memory Cell uses a gating function (equation 3.4) and a candidate memory (equation 3.5) to update the relation memory in a way that reflects how relevant the input sentence, s_k, is to the concept pair, (c_i, c_j). To compute the gate value g_{ij}^r, the Relation Memory Cell uses the two concept gate values, g_i^c, g_j^c from the Concept Memory Cells for concepts c_i and c_j, ensuring that input sentences that are relevant to either concept can be used to update the Relation Memory. By maintaining a memory vector for each pair of concepts and updating that memory vector as the model accumulates information across each sentence in an EEG report, EEG-RelNet can be interpreted as constructing a local *latent knowledge graph* (Bansal et al., 2017) for each EEG report, where each Relation Memory represents a possible relation in the graph.

3.3.3 The Output Module

The output module makes use of the Dynamic Relational Memory updated after processing the last sentence in the EEG report to identify relations (and their types) between any pair of medical concepts from the report. The relation prediction, $\hat{R}_{ij}$ between medical concepts

c_i and c_j is produced by passing the Concept Memories associated with concepts c_i and c_j along with the Relational Memory r_{ij} to two fully connected PReLU layers followed by a softmax layer: where $W_q \in \mathbb{R}^{N \times 3N}$ and $W_z \in \mathbb{R}^{5 \times N}$ are learned weight matrices, and

$$\mathbf{q_{ij}} = \phi\left(W_q\left[\mathbf{h_i}, \mathbf{h_j}, \mathbf{r_{ij}}\right]\right) \tag{3.7}$$

$$\mathbf{R^{ij}} = \text{softmax}\left(\phi\left(W_z \mathbf{q_{ij}}\right)\right) \tag{3.8}$$

ϕ is a Parametric Rectified Linear Unit. $\mathbf{R^{ij}}$ is a probability distribution over 5 possible relations: the 4 relation types described in the annotation schema and a 5^{th} type indicating no relation. Consequently, the relation (if any) detected between concepts c_i and c_j is given by $\hat{R}_{ij} = \text{argmax}_t \mathbf{R^{ij}}_t$.

EEG-RelNet is trained using cross-entropy:

$$\mathcal{L} = -\sum_i \sum_j \text{I}[y_{ij} = t] \log\left(\boldsymbol{R_t^{ij}}\right) \tag{3.9}$$

where $\text{I}[y_{ij} = t]$ is an indicator function denoting that the relation between concept i and concept j is of type t.

3.4 Joint Learning of Medical Concepts, their Attributes, and Relations Between them in EEG Reports

While EEG-RelNet is able to identify long-distance relations between medical concepts in EEG reports, it requires the medical concepts it relates to be identified a priori. Moreover, EEG-RelNet requires that those concepts also have their attributes identified. This is problematic for two reasons: (1) the performance of EEG-RelNet is reliant upon the performance of the concept and attribute identification systems upon which it runs; and (2) it complicates the efficient use of Active Learning to generate expert annotations on EEG reports. Due to the expertise required to generate concept, attribute, and relation annotations in EEG reports, it is crucial that manual annotation be done as efficiently as possible.

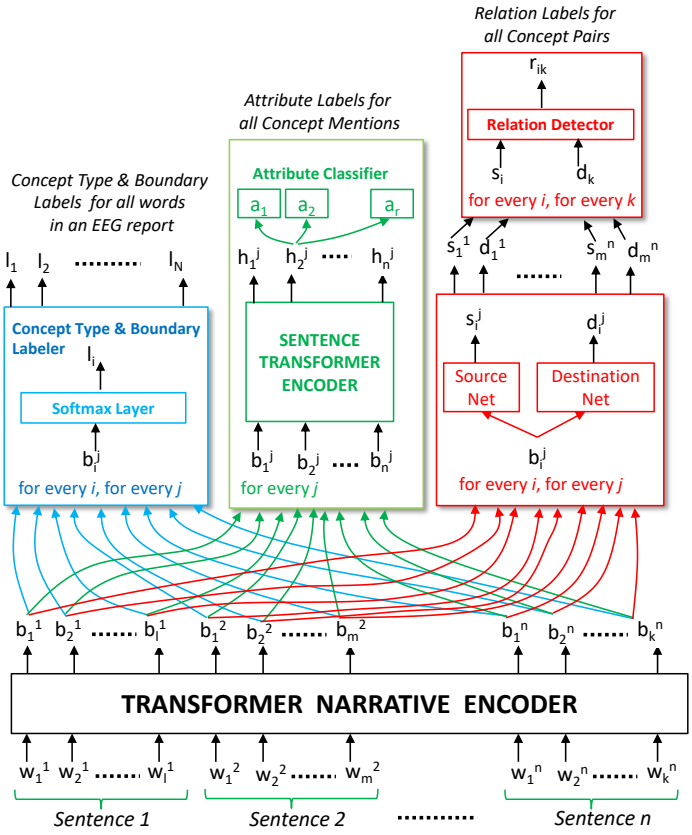

Figure 3.6. Architecture for Self-Attention Concept, Attribute and Relation (SACAR) Identification.

Active Learning (Settles, 2009) provides a solution to this problem, however training EEG-RelNet via active learning entails two rounds of the active learning loop, once for concept and attribute recognition, and another for relation extraction. Therefore, in this section we introduce a method for performing end-to-end information extraction from EEG reports: the Self-Attention Concept, Attribute and Relation (SACAR) identifier for automatically recognizing concepts, their attributes, and relations relations between concepts.

Inspired by the Bi-affine Relation Attention Networks (BRANs) presented in Verga et al. (2018), SACAR operates on entire EEG reports, enabling it to identify long-distance relations across multiple sentences from the EEG report. To accomplish this, SACAR uses *self-attention* to learn a representation of all the words in the EEG report. Self-attention is

an attention mechanism relating different positions of a single sequence (of words) with one another in order to compute a *contextualized* representation of each sequence member. Self-attention has been used successfully before in abstractive summarization (Paulus et al., 2018), textual entailment (Parikh et al., 2016) and learning task-independent sentence (Lin et al., 2017) and token (Devlin et al., 2019) representations.

As shown in Figure 3.6, SACAR first uses a transformer narrative encoder (Vaswani et al., 2017) to generate an encoding, b_i^j, for each word, w_i^j, in each sentence $Sentence_j$ in the narrative of the EEG report[4]. These encodings serve as input to: (1) **the concept type and boundary labeler** that detects the type and the boundaries of each concept mentioned in an EEG report; (2) **the attribute classifier** that recognizes the attributes of each concept mentioned in the EEG report; and (c) **the relation detector** that identifies relations between the concepts in EEG reports. As shown in Figure 3.6, these three modules operate jointly because they share the encoding of the words and sentences produced by the transformer narrative encoder. Moreover, each of these three modules has an associated loss function, namely $\mathcal{L}_C$ for the concept type and boundary labeler, $\mathcal{L}_A$ for the attribute classifier and $\mathcal{L}_R$ for the relation detector, respectively. We shall define the loss functions later in the section, when we detail the functionality of each module. The parameters of each module as well as the transformer narrative encoder are learned jointly by minimizing the combined loss:

$$\mathcal{L} = \gamma_C \mathcal{L}_C + \gamma_A \mathcal{L}_A + \gamma_R \mathcal{L}_R + \gamma_T \mathcal{L}_T \tag{3.10}$$

where $\mathcal{L}_T$, is the Adaptive Computation Time loss used in the encoder for learning the encodings of words from the EEG report, defined later in the section, while γ_C, γ_A, γ_R, and γ_T are hyperparameters (set to 0.85, 0.65, 1.0, and 0.05 in this work, respectively). The combined loss $\mathcal{L}$ is minimized using Adam (Kingma and Ba, 2015), a widely-used stochastic optimization algorithm.

[4]Tokenization was performed using the GENIA tagger (Tsuruoka et al., 2005) and sentence splitting was performed using OpenNLP (opennlp.apache.org).

3.4.1 The Transformer Narrative Encoder

The transformer narrative encoder (TNE) learns a contextualized encoding, $b_i^j \in \mathbb{R}^d$, for each word, w_i^j, in an EEG report given the full context of the EEG report using self-attention. The TNE consists of B recurrent blocks, denoted as TNE_k, with $k \in \{1, \ldots, B\}$. Each block TNE_k shares parameters and is made up of two sub-components: (a) a multi-head attention and (b) a series of convolutions, as in Verga et al. (2018). The output of the k^{th} block for the i^{th} word, denoted as bl_i^k, is connected to its input, $bl_i^{(k-1)}$ through a residual connection (He et al., 2016):

$$bl_i^{(k)} = bl_i^{(k-1)} + TNE_k(bl_i^{(k-1)}) \tag{3.11}$$

The i^{th} input to the first block of the TNE, $bl_i^0 = w_i$, is the word embedding of the i^{th} word in the EEG report. It should be noted that word embeddings are learned jointly along with the other parameters of this model in this work.

Each block TNE_k uses the multi-headed self-attention mechanism introduced in Vaswani et al. (2017), which allows our learning model to jointly attend to information from different representation subspaces at different positions in the sequence of words from each sentence of the EEG report. This amounts to using multiple, parallel self-attention functions, one for each head. The self-attention function maps a sequence of input vectors to a sequence of output vectors, where each output vector is a weighted sum of the input vectors and the weights are computed using *a compatibility function* that compares the sequence of input vectors to itself (hence the name *self*-attention). In this way, for head h using an input vector $v_i \in \mathbb{R}^d$, the self-attention function computes the output vector $o_{ih} \in \mathbb{R}^{d_o}$, as:

$$o_{ih} = \sum_j W_v^h v_j \odot \alpha_{ijh} \tag{3.12}$$

where $W_v^h \in \mathbb{R}^{d_o \times d}$ is a learned weight matrix, d_o is the dimension of the vector o_{ih}, $\odot$ denotes the element-wide multiplication, and α_{ijh} is the attention weight between inputs i and j of

attention head h computed by:

$$\alpha_{ijh} = \text{softmax}\left(\frac{W_q^h v_i \times \left(W_k^h v_j\right)^\intercal}{\sqrt{d}}\right) \tag{3.13}$$

where $W_q^h, W_k^h \in \mathbb{R}^{d \times d}$ are learned weight matrices.

The outputs of each attention-head, o_{ih}, are concatenated and projected back into d dimensions using a linear layer:

$$o_i = [o_{i1}, o_{i2}, \ldots, o_{iH}]W_O \tag{3.14}$$

where H is the number of attention heads, $W_O \in \mathbb{R}^{Hd_o \times d}$ is a learned weight matrix and $[\cdot;\cdot]$ is the concatenation operation. A residual connection combined with Layer Normalization (Ba et al., 2016), denoted as $LN(\cdot)$, is then applied to the output vectors:

$$m_i = LN(bl_i^{(k-1)} + o_i) \tag{3.15}$$

In addition to the multi-head attention, as in Verga et al. (2018), for each block TNE_k we have used a feed-forward network of three convolutional layers cl followed by a final residual connection and Layer Norm:

$$cl_i^{(0)} = ReLU(C_1(m_i^k)) \tag{3.16}$$

$$cl_i^{(1)} = ReLU(C_5(cl_i^{(0)})) \tag{3.17}$$

$$cl_i^{(2)} = C_1(cl_i^{(1)}) \tag{3.18}$$

$$TNE_k(bl_i^{(k-1)}) = LN(m_i^k + cl_i^{(2)}) \tag{3.19}$$

where $C_L(\cdot)$ represents a convolution operator of kernel width L and $TNE_k(bl_i^{(k-1)}) \in \mathbb{R}^d$ is the output of block k for word i.

It is to be noted that in EEG reports, certain words tend to be more ambiguous than others, suggesting that computing the encodings of these words requires additional processing to correctly capture their meaning from the contexts in which they appear. Therefore, some

words will require a larger number of TNE blocks than others. Inspired by Dehghani et al. (2018), we dynamically adjusted the number of blocks used to produce the encoding for each word in the narrative using Adaptive Computation Time (ACT) (Graves, 2016). ACT allows the model to learn how many processing steps (blocks) are necessary to encode each word and to dynamically halt processing for one word, but to continue processing for other words whose encodings still require further refinement. Once ACT has halted for a word in the EEG report narrative, its encoding is simply copied to the next block until processing has halted for all other words in the narrative or until a maximum number of blocks has been reached (12 in this work).

To determine if processing should halt for word i at block k, we used a sigmoidal *halting* unit with weights $W_h \in \mathbb{R}^{d\times 1}$ and bias $b_h \in \mathbb{R}$ (Graves, 2016):

$$h_i^k = \sigma\left(W_h TNE_k(bl_i^{(k-1)}) + b_h\right) \tag{3.20}$$

The *halting score*, h_i^k, for word i at block k is used to calculate (i) the number of recurrent updates for word i, $N(i)$; (ii) the *halting probability* p_i^k; and (iii) the *remainder*, $R(i)$, defined as:

$$N(i) = \min\left\{k' : \sum_{k=1}^{k'} h_i^k \geq 1-\epsilon\right\} \tag{3.21}$$

$$p_i^k = \begin{cases} R(i) & \text{if } k = N(i) \\ h_i^k & \text{otherwise} \end{cases} \tag{3.22}$$

$$R(i) = 1 = \sum_{k=1}^{N(i)-1} h_i^k \tag{3.23}$$

where *epsilon* is a small constant (0.01 in this work). Equation 3.21 bounds the number of recurrent updates for word i by the total halting score for all blocks such that processing halts when $\sum_{k=1} h_i^k \geq 1-\epsilon$. The remainder, $R(i)$, represents the amount of halting score remaining before processing halts. The final encoding for word i produced by the transformer

narrative encoder is the sum of the output at each block weighted by the halting probability for that block:

$$b_i = \sum_{k=1}^{N(i)} p_i^k TNE_k(bl_i^{(k-1)}) \tag{3.24}$$

In order to allow the neural network to learn when to halt or continue processing, we add the differentiable loss term described in Graves (2016) which bounds the total recurrent computation performed by the TNE:

$$\mathcal{L}_T = \sum_{i=1} N(i) + R(i) \tag{3.25}$$

3.4.2 Concept Type and Boundary Annotator

The concept type and boundary annotator first identifies spans of text that correspond to medical concept mentions by assigning a label to each word in the narrative of the EEG report indicating that the word begins a medical concept mention (B), is inside of a medical concept mention but not at the beginning (I), or is outside of a medical concept mention (O). In this way, medical concept mentions can be identified by continuous sequences of words starting with a word labeled B optionally followed by words labeled I.

In order to also identify the *type* of the medical concepts from the EEG reports and to distinguish EEG activities, EEG events, medical problems, treatments, and tests, we extended the IOB labeling system to include separate B and I labels for each concept type, yielding 11 total medical concept boundary and type labels: $L_C = \{$*B-ACT, I-ACT, B-EV, I-EV, B-PR, I-PR, B-TR, I-TR, B-TE, I-TE, O*$\}$. The concept type and boundary labeler assigns a label $l_i \in L_C$ to each word w_i in the EEG report by passing each word's encoding produced by the TNE through a fully connected softmax layer to produce a distribution p_i^C over the labels:

$$p_i^C = \text{softmax}\left(W_C b_i + b_C\right) \tag{3.26}$$

where $W_C \in \mathbb{R}^{d \times 11}$ is a weight matrix and $b_C \in \mathbb{R}^{11}$ is a bias vector. Then, the predicted label for word w_i is the label with the highest probability $l_i = \text{argmax}_j p_{ij}^C$.

We trained the concept type and boundary labeler to maximize the likelihood of labeling each word correctly, considering that the labels y_i^C are conditionally independent, given the word encoding produced by the transformer narrative encoder, namely b_i. This allowed us to define:

$$\mathcal{L}_C = -\sum_i \log P(y_i^C|b_i) \tag{3.27}$$

where the probability $P(y_i^C|b_i)$ is the probability assigned to the label y_i^C in p_i^C.

3.4.3 Attribute Classifier

Each medical concept automatically identified in an EEG report is associated with several attributes including its modality, and polarity. In addition, EEG activities have 16 specific attributes defined in Table 2.3. After each medical concept is identified in the EEG report, the attribute classifier determines the values of each attribute type for each medical concept using another transformer encoder – this time at the sentence level – and a series of linear classifiers, one for each attribute type.

In each EEG report, the textual cues that signal attribute values for a medical concept are typically found in the same sentence as the concept mention. Therefore, the attribute classifier needs to further refine the encoding for each word using the *sentence transformer encoder* (STE), which is a transformer module similar to the TNE, operating at the sentence level instead of the full narrative. By operating at the sentence level, the STE allows the model to attend to each word in the surrounding sentence, determining which context words are most informative for attribute classification, while ignoring irrelevant out-of-sentence words. The STE is defined exactly the same as the TNE, consisting of a set of B^s identical blocks (8 in this work) where the number of blocks used is dynamically determined for each word using Adaptive Computation Time. For each sentence S_j in the EEG report, the STE produces an encoding $h_i^j \in \mathbb{R}^d$ for each word in S_j using the sequence of encodings b_i^j for the words in sentence S_j produced by the TNE, as illustrated in Figure 3.6. We also

took into account the annotations produced by the concept type and boundary annotator on S_j such that for each concept c_k^j identified in S_j we considered only the encoding h_i^j of the sentence words found within the boundaries of c_k^j. In addition we took into account the type $t(c_k^j)$ of concept c_k^j to decide which of the 18 attributes should be identified for that concept. If $t(c_k^j) = \text{EEGACTIVITY}$ each of the 18 attributes will be considered but if $t(c_k^j) \in \{\text{EEGEVENT}, \text{MEDICALPROBLEM}, \text{TREATMENT}, \text{TEST}\}$, we shall only consider the attributes POLARITY and MODALITY. In this way, given any attribute a selected for concept c_k^j, we computed the distribution p_{kj}^a over the values for attribute a as follows:

$$p_{kj}^a = \text{softmax}\left(W_{A1}^a ReLU(W_{A0}^a c_k^j + b_{A0}^a) + b_{A0}^a\right) \tag{3.28}$$

where $W_{A1}^a \in \mathbb{R}^{\Delta(a)\times d}, W_{A0}^a \in \mathbb{R}^{d\times d}$ are weight matrices, d is the dimension of the encodings produced by the STE, $\Delta(a)$ is the number of distinct attribute values of attribute a, and $b_{A1}^a \in \mathbb{R}^{\Delta(a)}, b_{A0}^a \in \mathbb{R}^{d_A}$ are bias vectors.

Similarly to the concept type and boundary annotator, the attribute classifier is trained to maximize the likelihood of correctly classifying each attribute of every concept identified in an EEG report, where the attribute values, y_{kj}^a are conditionally independent given the concept encodings, c_k^j:

$$\mathcal{L}_A = -\sum_{(j,k)\in\Phi}\sum_a P(y_{kj}^a | c_k^j) \tag{3.29}$$

where Φ is the set of $(\text{sentence}, \text{concept})$ indices (j,k) denoting concept mentions and the probability $P(y_{kj}^a|c_k^j)$ is given by the probability assigned to attribute value y_{kj}^a in the predicted distribution, p_{kj}^a, for attribute a.

3.4.4 Relation Detector

The relation detector automatically identifies relations between pairs of concepts recognized in EEG reports (the definition of relations types was provided in Section 3.2). Recall from the discussion in Section 3.2 that relations are identified between concept *entities*, not concept

mentions. Concept entities are identified by concept mentions from the same EEG report that have the same (1) normalized name, (2) type, and (3) attribute values. It should be noted that the automatic identification of medical concept type and boundaries performed by the SACAR neural system recognizes only concept mentions, requiring this normalization step. Moreover, observing that relations in EEG reports are directed, we consider both directions when classifying potential relations. Specifically, we considered for each pair of concept entities two possible cases, namely (case 1) in which the first concept entity is the the *source concept* of the potential relation and the second concept entity is the *destination* of the relation; or (case 2) in which the first concept entity is the destination whereas the second concept entity is the source.

In this framework relation discovery was cast as a prediction of the most likely relation type between two concept entities. The Relation Detector performs simultaneously prediction of both directions of a potential relation between concepts, considering that the first concept of the pair is the source while the second concept is the destination for one possible relation, and conversely for a second possible relation. As in Verga et al. (2018), to enable the prediction of the most likely relation, the Relation Detector generates for each concept mention c_i^k of every concept entity both a *source encoding*, s_i^k and a destination encoding d_i^k using the Source Net, and the Destination Net, respectively, as illustrated in Figure 3.6, implemented as two-layer neural networks:

$$s_i^j = \text{SourceNet}(c_i^j) = W_S^1 \left(\text{ReLU}\left(W_S^0 c_i^j\right)\right) \tag{3.30}$$

$$d_i^j = \text{DestinationNet}(c_i^j) = W_D^1 \left(\text{ReLU}\left(W_D^0 c_i^j\right)\right) \tag{3.31}$$

where $W_S^0, W_S^1, W_D^0, W_D^1 \in \mathbb{R}^{d \times d}$ are weight matrices. These encodings enabled us to represent all possible relations between each pair of concept entities from an EEG report in terms of their mentions using an $N \times R \times N$ tensor, L, where N is the number of concept mentions discovered in the EEG report and R is the number of possible *relation types*. For each source

concept entity S and each destination concept entity D, we consider (a) the source encodings of all the mentions of S in the EEG report, denoted as s_m; and (b) the destination encodings we produced for all the mentions of D, denoted as d_n. In order to compute each value of L, we used a bi-affine function between a source encoding, s_m, and a destination encoding, d_n, for each relation type $r \in R$:

$$L_{rmn} = (s_m^{\intercal} Q_r)\, d_n \tag{3.32}$$

where $Q \in \mathbb{R}^{R \times d \times d}$ is a learned tensor representing a set of $d \times d$ embedding matrices (one for each relation type $r \in R$). This enabled us to compute the scores of all relation types between a pair of concept entities, in which the source is S and the destination is D using the LogSumExp function as in Verga et al. (2018):

$$scores(S, D) = \log \sum_{\substack{m \in M(S) \\ n \in M(D)}} \exp(L_{m,n}) \tag{3.33}$$

where $M(S)$ are all the mentions of concept entity S and $M(D)$ are all the mentions of the concept entity D in the same EEG report. Note that $scores(S, D) \in \mathbb{R}^R$ is a vector of scalar scores for each relation type between S and D and the LogSumExp function is a smooth approximation to the max function (Das et al., 2017). The probability distribution over the possible relation types between S and D can then be calculated using the softmax function:

$$p_{S,D}^{Rel} = \text{softmax}\left(scores(S, D)\right) \tag{3.34}$$

To train the relation detector, we maximized the likelihood of classifying each relation between each pair of concepts entities correctly:

$$\mathcal{L}_R = -\sum_{(S,D)} \log P\left(y_{S,D}^r | scores(S, D)\right) \tag{3.35}$$

where $y_{S,D}^r$ is the type of the relation from concept entity S to concept entity D and $P\left(y_{S,D}^r | scores(S, D)\right)$ is the probability assigned to relation type $y_{S,D}^r$ in the distribution $p_{S,D}^{Rel}$.

Table 3.1. Data statistics for the evaluation dataset.

Concept Types				
Activity	Event	Problem	Test	Treatment
1438	452	798	716	539
Relation Types				
EVIDENCES	EVOKES	TREATMENT-FOR	CLINICAL-CORRELATION	
397	342	356	195	

3.5 Experimental Results and Discussions

In this section, the deep learning architectures introduced in this chapter are evaluated using a set of 140 EEG reports with concepts, attributes, and relations manually annotated. This is the same set of reports used in Chapter 2 with relation annotations generated by the same annotators with an average inter-annotator agreement of 0.8843, measured using Jaccard Score. Statistics for concept and relation types for the evaluation dataset are provided in Table 3.1. The dataset is split into training/validation/test splits of 100/10/30. First, evaluations of EEG-RelNet for recognizing long-distance relations in EEG reports are presented. Next, because SACAR performs end-to-end clinical information extraction tasks including concept detection and attribute classification, in addition to relation detection, each task is evaluated for SACAR. Performance is measured using Precision (P), Recall (R), and F_1 score (F_1).

3.5.1 Evaluation of EEG-RelNet for Recognizing Long-Distance Relations in EEG Reports

To measure the impact of the EEG-RelNet architecture, we compare our system with two alternate configurations and one baseline:

1. **EEG-RelNet_NRM** is a deep neural network structured similarly to EEG-RelNet but without Relation Memories. Formally, we omit equations [3.4–3.6] and replace equation 3.7 with $\mathbf{q_{ij}} = \phi\left(\mathbf{W_q}\left[\mathbf{h_i}, \mathbf{h_j}\right]\right)$.

Table 3.2. Hyper-parameters of EEG-RelNet. Selected values indicated by an asterisk*.

Hyper-parameter	Values	Hyper-parameter	Values
Hidden size	N=[100*, 300]	Learning Rate	[1e-2, 1e-3*, 1e-4]
Batch size	[8, 16, 32*]	Dropout Probability	[0.1, 0.25, 0.5*]

2. **EEG-RelNet_NA** is a deep neural network structured similarly to EEG-RelNet that ignores the attributes of each medical concept in the Input Module. Formally, EEG-RelNet_NA represents each concept embedding using only the embedding for the name of that concept, $\vec{\boldsymbol{c_i}} = \tilde{\boldsymbol{c_i}}$.
3. **Heuristic** is a simple rule-based baseline from Maldonado et al. (2017) that uses medical concept type and section type to detect relations. EVIDENCES relations are created between any medical concept in an EEG report and medical problems in the clinical correlation section, EVOKES relations are created between any medical concept and an EEG activity, TREATMENT-FOR relations are created between any treatment and medical problems in the history section of the EEG report, and CLINICAL-CORRELATION relations are created between EEG activities in the impression section and medical problems in the clinical correlation section.

Each EEG-RelNet configuration is trained for 10 epochs with the same random initialization with early-stopping using validation F_1 score. Hyper-parameters are set using grid search over the values defined in Table 3.2.

EEG-RelNet is able to successfully detect the four relation types, EVOKES, EVIDENCES, TREATMENT-FOR, and CLINICAL-CORRELATION obtaining F_1 scores of 0.8387, 0.6674, 0.7358, and 0.8487, respectively. Clearly, EEG-RelNet obtains the best performance on each relation type, demonstrating the importance of both the Dynamic Relational Memory and medical concept attributes when detecting relations. EEG-RelNet achieves significantly better performance when recognizing EVOKES and CLINICAL-CORRELATION relations compared to the other two relation types indicating that the network is able to correctly link medical problems

Table 3.3. Evaluation Results EEG-RelNet for relation identification in EEG Reports.

	EVOKES			EVIDENCES		
Model	Precision	Recall	F_1	Precision	Recall	F_1
EEG-RelNet	**0.8601**	0.8183	**0.8387**	0.6754	0.6596	**0.6674**
EEG-RelNet_NRM	0.8185	0.7556	0.7858	**0.6852**	0.5255	0.5948
EEG-RelNet_NA	0.6987	0.6481	0.6725	0.6689	0.5753	0.6186
Heuristic	0.1960	**0.9771**	0.3265	0.1750	**0.8624**	0.2910
	TREATMENT-FOR			CLINICAL-CORRELATION		
Model	Precision	Recall	F_1	Precision	Recall	F_1
EEG-RelNet	**0.6060**	0.9365	**0.7358**	**0.8422**	0.8554	**0.8487**
EEG-RelNet_NRM	0.5946	0.9163	0.7212	0.8041	0.7348	0.7679
EEG-RelNet_NA	0.5523	0.8680	0.6751	0.8022	0.7787	0.7902
Heuristic	0.1715	**0.9852**	0.2921	0.1895	**0.9820**	0.3177
	All Relations (Macro Average)					
Model	Precision	Recall	F_1			
EEG-RelNet	**0.7459**	0.8175	**0.7727**			
EEG-RelNet_NRM	0.7256	0.7331	0.7174			
EEG-RelNet_NA	0.6805	0.6925	0.6891			
Heuristic	0.1830	**0.9517**	0.3068			

with the EEG activities they evoke and the with which they clinically correlate. The effect of the Dynamic Relational Memory is most obvious when considering the EVIDENCES relation type, increasing the F_1 measure by more than 10%. Interestingly, the removal of attribute information from the model drastically reduces performance when detecting the EVOKES relation type, but only slightly reduces performance on the other two types compared to the EEG-RelNet_NRM system. The Heuristic approach is able to achieve the highest recall on each relation type since it was specifically designed for high recall. However, due to the poor precision, the Heuristic baseline achieves by far the worst overall performance.

3.5.2 Evaluation of SACAR for Clinical Information Extraction

This subsection presents evaluations of the SACAR identifier for automatically recognizing concepts, their attributes, and relations spanning them. For training purposes, another subset of 1,000 EEG reports with *silver-standard* annotation produced by the concept detection and attribute classification systems presented in Chapter 2 and relation extraction produced

Table 3.4. Data statistics for the evaluation dataset along with automatically generated silver-standard annotations.

	Concept Types				
Annotations	Activity	Event	Problem	Test	Treatment
Gold	1438	452	798	716	539
Silver	8820	3033	5367	5049	3416

	Relation Types			
Annotations	EVIDENCES	EVOKES	TREATMENT-FOR	CLINICAL-CORRELATION
Gold	397	342	356	195
Silver	2850	2326	2065	1228

by EEG-RelNet are used to augment the training data. The statistics for the silver-standard data are provided in Table 3.4 along with the gold-standard data, for reference. The silver-standard data was used to augment the training data for SACAR during training. SACAR was compared to EEG-RelNet for relation extraction as well as the stacked LSTM (sLSTM) from Section 2.4 for concept boundary detection and the Deep ReLU Network (DRN) from Section 2.5 for attribute classification. Each of the these neural baselines was trained to perform one of the three tasks discussed above, while SACAR is trained to perform all tasks jointly.

When evaluating the results of SACAR, we took also into account the fact that SACAR uses two transformer encoders – the TNE and the STE – to produce internal representations of each EEG report. We were interested to evaluate two important properties of SACAR's transformer encoders: (1) recurrence and (2) Adaptive Computation Time. Inspired by Dehghani et al. (2018) we hypothesized that introducing recurrence to the transformer encoders could help SACAR learn better from our small amount of labeled data and that ACT could further boost performance by allowing SACAR to dynamically allocate more resources to more complicated encodings. In this section, we will refer to the full SACAR model described in Section 3.4 that uses both recurrence and ACT based on the Adaptive Universal Transformer (Dehghani et al., 2018) as SACAR-A. We evaluate SACAR-A against two alternative

Table 3.5. Hyper-parameters of SACAR. Selected values indicated by an asterisk*.

Hyper-parameter	Values	Hyper-parameter	Values
Hidden size	d=[100*, 300]	Learning Rate	[1e-2, 1e-3, 1e-4*]
Batch size	[4, 8, 16*]	Dropout Probability	[0.1, 0.25, 0.5*]
Concept Weight	$\mathcal{L}_C$=[0.65, 0.85*, 1.0]	Attribute Weight	$\mathcal{L}_A$=[0.65*, 0.85, 1.0]
Relation Weight	$\mathcal{L}_R$=[0.65, 0.85, 1.0*]	ACT Weight	$\mathcal{L}_T$=[0.05*, 0.1, 0.2]
TNE stacks	k=[4*, 8]	TNE attention heads	H=[4, 8*, 12]
STE stacks	k_S=[2*]	STE attention heads	H_S=[4, 8*, 12]

configurations: (1) SACAR-V which uses a simple *vanilla* transformer encoder without recurrence or ACT as described in Vaswani et al. (2017); and (2) SACAR-U based on the Universal Transformer described in Dehghani et al. (2018) which uses recurrence, but not ACT. Formally, SACAR-V is equivalent to SACAR-A where each TNE and STE block has its own parameters and Equation 3.24 is replaced by $b_i = TNE_B(bl_i^{(B-1)})$ given B blocks. Similarly, SACAR-U is equivalent to SACAR-V where parameters are shared between TNE blocks and between STE blocks (i.e., the transformer encoders are *recurrent*). Each SACAR configuration is trained for 10 epochs with the same random initialization with early-stopping using validation F_1 score averaged over the concept detection, attribute classification, and relation extraction tasks. Hyper-parameters are set using grid search over the values defined in Table 3.5.

The results for concept type and boundary detection are presented in Table 3.6 in terms of precision, recall, and F_1 score, where predicted concept boundaries are considered correct if they exactly match a manually annotated boundary using the exact match protocol as in Section 2.7. We compare SACAR and its alternate configurations to two sLSTM baselines. Recall from Section 2.4 that two sLSTM models are required to detect medical concepts in EEG reports, one for detecting EEG activities and one for detecting the other types medical concepts. As we can see from the Table, the SACAR-A model outperforms the sLSTM baselines as well as the other alternative implementations of the transformers for all concept types, except for Treatment, where it is slightly outperformed by SACAR-U. Interestingly,

Table 3.6. Evaluation Results for Concept Type and Boundary Recognition.

	EEG Activity			**EEG Event**			**Medical Problem**		
Model	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1
LSTM	0.8949	0.8125	0.8517	0.8842	0.8301	0.8563	0.8391	0.7863	0.8118
SACAR-V	0.8662	0.8287	0.8522	0.8378	0.8076	0.8224	0.6967	0.6121	0.6516
SACAR-U	0.8734	**0.9422**	0.9065	0.8375	0.8761	0.8564	0.8028	0.7923	0.7975
SACAR-A	**0.9327**	0.9153	**0.9239**	**0.9048**	**0.8800**	**0.8922**	**0.8985**	**0.8766**	**0.8874**

	Treatment			**Test**			**All Types (Macro Average)**		
Model	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1
LSTM	0.9257	0.8687	0.8963	0.8904	0.9250	0.9074	0.8869	0.8645	0.8756
SACAR-V	0.8079	0.7940	0.8009	0.8457	0.8822	0.8636	0.8109	0.7851	0.7981
SACAR-U	0.9186	**0.8967**	0.9075	0.8666	0.8564	0.9092	0.8798	0.8727	0.8755
SACAR-A	**0.9687**	0.8747	**0.9193**	**0.8945**	**0.9296**	**0.9117**	**0.9198**	**0.8917**	**0.9069**

the only model that does not make use of recurrence, SACAR-T, performs worst – obtaining results that are outperformed even by the LSTM baseline.

When evaluating the results for attribute classification, we used as baseline the Deep ReLU Network (DRN) described in Section 2.5. As was the case for boundary detection, we used two DRN models: one for EEG activity attributes, and one for the attributes of all other medical concepts. It should be noted that both baselines used for concept type and boundary detection and for attribute classification, namely the LSTM and DRN baselines, rely on hand-crafted features while the SACAR models are trained end-to-end with no feature extraction needed. The results of the baseline used for attribute classification as well as the results of all three SACAR model are listed in Table 3.7, evaluated with Precision, Recall, and F_1 score. Table 3.7 presents the results for each value of each attribute along with their prevalence in the gold annotated data (indicated by the '#' symbol). Attribute values with no examples in the training data are omitted. Aggregated metrics are presented for each attribute type using micro-average, however precision and recall are omitted for multi-class classification tasks since they are equivalent to F_1. All three SACAR models tend to outperform the DRN baseline in micro-averaged F_1 score.

The results for relation identification are presented in Table 3.8, compared against EEG-RelNet. In general both SACAR-A and SACAR-U are able to produce similar – yet slightly worse – results as EEG-RelNet, with SACAR-V consistently performing worst. It should

Table 3.7. Evaluation Results for Attribute Classification.

Attribute	#	Precision				Recall				F_1			
		DRN	SACAR-V	SACAR-U	SACAR-A	DRN	SACAR-V	SACAR-U	SACAR-A	DRN	SACAR-V	SACAR-U	SACAR-A
Morphology	1438	–	–	–	–	–	–	–	–	0.7716	0.8548	0.8263	**0.8868**
Rhythm	388	0.6087	0.8941	0.8254	0.9549	0.8660	0.9965	0.9820	0.9961	0.7149	0.9425	0.8969	0.9751
V wave	37	0.6977	0.6849	0.5217	0.7143	0.8108	0.8228	0.8000	1.0000	0.7500	0.7475	0.6316	0.8333
Wicket spikes	11	0.5000	0.7000	0.5000	1.0000	0.1818	0.6363	0.1818	1.0000	0.2667	0.6667	0.2677	1.0000
Spike	43	0.8000	0.5319	0.7083	0.7500	0.5581	0.5814	0.3953	0.3023	0.6575	0.5556	0.5075	0.4309
Sharp wave	107	0.9195	0.7091	0.6696	0.6522	0.7577	0.7290	0.7009	0.8333	0.8247	0.7189	0.6849	0.7317
Slow wave	64	0.8929	0.8305	0.8276	0.9048	0.7813	0.7656	0.7500	0.8261	0.8333	0.7967	0.7869	0.8636
K-complex	11	0.8889	0.9000	0.6667	1.0000	0.7273	0.8181	0.5454	1.0000	0.8000	0.8571	0.6000	1.0000
Sleep spindles	28	0.5526	0.7248	0.6875	0.8333	0.7500	0.8081	0.7917	1.0000	0.6364	0.7642	0.7360	0.9091
Spike-and-slow	106	1.0000	1.0000	1.0000	1.0000	0.6132	0.5934	0.4615	0.6250	0.7602	0.7448	0.6316	0.7692
Triphasic wave	16	0.8333	0.7143	0.8000	1.0000	0.6250	0.3125	0.2500	0.8000	0.7143	0.4348	0.3810	0.8889
Polyspike complex	29	0.7500	0.9286	0.8750	0.4000	0.5172	0.4483	0.2414	1.0000	0.6122	0.6047	0.3784	0.5714
Suppression	48	0.5814	1.0000	0.6923	1.0000	0.5208	0.1667	0.3103	0.2857	0.5495	0.2857	0.4286	0.4444
Slowing	174	0.9371	0.7910	0.7453	0.8710	0.7701	0.9138	0.9080	0.9000	0.8454	0.84480	0.8187	0.8852
Breach Rhythm	12	0.8000	1.0000	0.8889	1.0000	0.6667	1.0000	0.6667	1.0000	0.7273	1.0000	0.7619	1.0000
Photic Driving	51	0.8750	1.0000	0.9329	1.0000	0.6863	0.8294	0.7414	1.0000	0.7692	0.9067	0.8262	1.0000
PLEDs	20	0.5625	0.4167	0.2222	0.3333	0.4500	0.2500	0.2000	0.2500	0.5000	0.3125	0.2105	0.2857
Epileptiform discharge	136	0.7934	0.6519	0.6173	0.7000	0.7059	0.7574	0.7353	0.6364	0.7471	0.7007	0.6711	0.6667
Disorganization	98	0.8000	0.7416	0.6574	0.7895	0.6122	0.6735	0.7245	0.6250	0.6936	0.7059	0.6893	0.6977
Unspecified	59	0.3855	0.8158	0.7941	1.0000	0.5424	0.5254	0.4576	0.4286	0.4507	0.6392	0.5806	0.6000
Frequecy Band	1438	–	–	–	–	–	–	–	–	0.8754	0.8911	0.8862	**0.9356**
Alpha	128	0.8632	0.8710	0.8779	0.9286	0.7891	0.8438	0.8984	0.8387	0.8245	0.8571	0.8880	0.8814
Beta	91	0.6881	0.3929	0.3704	0.8750	0.8242	0.2418	0.2198	0.6364	0.7500	0.2993	0.2759	0.7368
Delta	129	0.8416	0.7381	0.7949	0.9524	0.6589	0.4806	0.4806	0.7143	0.7391	0.5822	0.5990	0.8163
Theta	118	0.7755	0.7471	0.7944	0.9000	0.6441	0.5508	0.7203	0.6923	0.7037	0.6341	0.7556	0.7826
N/A	972	0.9067	0.9683	0.9593	0.9744	0.9410	0.9889	0.9797	0.9954	0.9235	0.9785	0.9694	0.9848
Background	1438	0.8904	0.9506	0.9657	**0.9704**	0.8197	0.9110	**0.9506**	0.9428	0.8543	0.9340	**0.9581**	0.9564
Magnitude	1438	–	–	–	–	–	–	–	–	0.8148	0.9175	**0.9461**	0.9273
Low	184	0.7548	0.7669	0.7669	0.7273	0.5939	0.6757	0.6757	0.4444	0.6648	0.7184	0.7184	0.5517
High	175	0.6273	0.5119	0.5119	0.8421	0.6900	0.5890	0.4674	0.4848	0.6571	0.5478	0.7184	0.6154
Normal	1079	0.9210	0.9627	0.9909	0.9636	0.9396	1.0000	0.9992	0.9934	0.9302	0.9810	0.9851	0.9783
Recurrence	1438	–	–	–	–	–	–	–	–	0.7174	0.8746	0.8802	**0.9045**
Continuous	224	0.7244	0.6459	0.6286	0.6750	0.6546	0.7411	0.6185	0.8438	0.6878	0.6902	0.6235	0.7500
Repeated	262	0.7974	0.6809	0.7333	0.7179	0.6461	0.7328	0.6740	0.8740	0.7138	0.7059	0.7024	0.7883
None	952	0.6607	0.9626	0.9677	0.9761	0.8163	0.9563	0.9603	0.9574	0.7303	0.9594	0.9640	0.9667
Dispersal	1438	–	–	–	–	–	–	–	–	0.7283	0.8812	**0.9222**	0.8951
Localized	330	0.5955	0.8144	0.8174	0.7714	0.6424	0.8061	0.9091	0.5094	0.6181	0.8509	0.8608	0.6136
Generalized	246	0.5162	0.8061	0.8132	0.8400	0.5813	0.8618	0.9024	0.4118	0.5468	0.8330	0.8555	0.5526
N/A	862	0.8441	0.8935	0.9713	0.9464	0.7947	0.9153	0.9408	0.9943	0.8186	0.9043	0.9558	0.9698
Hemisphere	1438	–	–	–	–	–	–	–	–	0.8148	0.8878	0.9042	**0.9117**
Right	96	0.7634	0.8346	0.8571	0.8182	0.7396	0.5362	0.5556	0.6429	0.7513	0.6529	0.6742	0.7200
Left	159	0.8257	0.8519	0.8810	0.7647	0.5660	0.7233	0.6981	0.5200	0.6716	0.7823	0.7789	0.6190
Both	246	0.6027	0.9052	0.8832	0.8182	0.5488	0.7764	0.7683	0.4737	0.5745	0.8359	0.8217	0.6000
N/A	937	0.8603	0.9149	0.9248	0.9540	0.9218	0.9755	0.9840	0.9920	0.8900	0.9442	0.9535	0.9727
Modality	1438	–	–	–	–	–	–	–	–	0.9337	0.9613	**0.9690**	0.9550
Factual	1366	0.9696	0.9753	0.9754	0.9682	0.9939	0.9846	0.9883	0.9682	0.9816	0.9800	0.9818	0.9682
Possible	70	0.4308	0.8600	0.8627	0.7824	0.4000	0.6143	0.6286	0.5027	0.4148	0.7167	0.7273	0.6121
Proposed	76	0.2318	0.8776	0.8800	0.7429	0.4605	0.5658	0.5789	0.5098	0.3084	0.6880	0.6984	0.6049
Polarity	1438	0.9100	0.8864	**0.9245**	0.9165	0.7389	0.8121	0.8857	**0.9071**	0.8156	0.8476	0.9047	**0.9118**
Location	692	0.7450	0.6982	0.7285	**0.7506**	0.5958	0.6145	0.5753	**0.6062**	0.6618	0.6537	0.6429	**0.6707**
Frontal	189	0.7302	0.7091	0.7258	0.7522	0.4868	0.5044	0.4990	0.5951	0.5841	0.5895	0.5914	0.6645
Occipital	268	0.7871	0.7088	0.7500	0.7944	0.7313	0.8055	0.7216	0.7401	0.7582	0.7541	0.7355	0.7663
Temporal	128	0.6786	0.7338	0.7348	0.7561	0.5938	0.6418	0.6289	0.6257	0.6333	0.6847	0.6777	0.6847
Central	30	0.7500	0.6724	0.6827	0.6201	0.4109	0.5006	0.4548	0.3195	0.5309	0.5739	0.5459	0.4217
Parietal	10	0.5714	0.2500	0.6000	0.5018	0.4000	0.4000	0.5000	0.3520	0.4706	0.3077	0.5455	0.4138
Frontocentral	50	0.8824	0.8247	0.8422	0.8563	0.6000	0.5818	0.5981	0.6244	0.7143	0.6823	0.6995	0.7222
Frontotemporal	17	0.5618	0.4086	0.5374	0.4218	0.4963	0.3238	0.3877	0.3380	0.5270	0.3613	0.4504	0.3753

Table 3.8. Evaluation Results for Relation Identification.

	EVOKES			EVIDENCES		
Model	Precision	Recall	F_1	Precision	Recall	F_1
EEG-RelNet	**0.8601**	0.8183	0.8387	**0.6754**	**0.6596**	**0.6674**
SACAR-V	0.7853	0.7488	0.7666	0.5658	0.6094	0.5868
SACAR-U	0.8174	0.8684	0.8421	0.6026	0.5871	0.5948
SACAR-A	0.8597	**0.8954**	**0.8767**	0.5937	0.6556	0.6231
	TREATMENT-FOR			CLINICAL-CORRELATION		
Model	Precision	Recall	F_1	Precision	Recall	F_1
EEG-RelNet	**0.6060**	**0.9365**	**0.7358**	0.8422	0.8554	0.8487
SACAR-V	0.5109	0.9164	0.6561	0.8255	0.8573	0.8411
SACAR-U	0.5725	0.9163	0.7047	0.8413	**0.8662**	**0.8536**
SACAR-A	0.6063	0.9245	0.7323	**0.8639**	0.7865	0.8234
	All Relations (Macro Average)					
Model	Precision	Recall	F_1			
EEG-RelNet	**0.7459**	**0.8175**	**0.7727**			
SACAR-V	0.6719	0.7830	0.7127			
SACAR-U	0.7085	0.8095	0.7488			
SACAR-A	0.7309	0.8155	0.7639			

be noted that the SACAR models do not only identify relations, but also recognize concept types and boundaries and classify concept attributes as well.

3.5.3 Discussions

The experimental results indicate that the SACAR identifier is able to jointly extract medical concepts from EEG reports, classify their attributes, and detect relations between them. While SACAR is able to out-perform the sLSTM and DRN for concept type and boundary detection and attribute classification in EEG reports, it does not out-perform EEG-RelNet for relation detection, achieving similar results. In general, both EEG-RelNet and SACAR are able to correctly recognize relations between medical concepts as indicated by the macro-average F_1 scores of 0.7727 and 0.7639, respectively.

In order to better understand the SACAR model, we conducted a brief ad hoc error analysis on a few synthetic excerpts illustrating common errors. Consider the following sentence indicative of typical text found in the description section of an EEG report: "*In addition,*

there is asymmetric anterior predominant small amplitude polyspikes and spike and waves seen generalized that at times have an EMG correlate noted in the arms and in the video on the face." SACAR is unable to correctly identify that two EEG activity anchors are present ("*polyspikes*" and "*spike and wave*"), instead annotating a single erroneous anchor ("*polyspikes and spike and waves*"). This incorrect boundary identification causes a cascade of failures in both attribute classification and relation identification. Since there is only one anchor identified, the anchor is determined to have the Polyspike-complex morphology, leaving the spike-and-slow-wave complex un-annotated. Since the spike-and-slow-wave complex is not identified, the EVOKES relation it has with the EEG Event "*EMG correlate*" is also not identified. Another source of errors for both EEG-RelNet and SACAR stems from the complicated – and sometimes ungrammatical – way in which EEG activities are described. For instance, we ran the models on the example sentence: "*There are rare sharp transients noted in the record but without after going slow waves as would be expected in epileptiform sharp waves.*" This sentence is meant to convey the fact that sharp waves have occurred, but sharp-and-slow-wave-complexes which would indicate epileptiform activity have not occurred. However, three anchors are identified by SACAR, "*sharp transients*", "*slow waves*", and "*epileptiform sharp waves*" with morphologies Sharp-wave, Slow-wave, and Epileptiform-discharge(unspecified), respectively. SACAR is unable to associate the text '*slow waves*" with "*sharp transients*" to identify that the morphology of the "*slow waves*" anchor should be sharp-and-slow-wave-complex. Moreover, the polarity of the epileptiform activity is not correctly identified as being negative.

As can be seen in Table 3.8, both EEG-RelNet and SACAR exhibit moderate performance in identifying TREATMENT-FOR relations due to low precision. A typical erroneous TREATMENT-FOR relation is exemplified in the following synthetic excerpt: "*CLINICAL HISTORY: 50 year old with history of seizures status post code. MEDICATIONS: Dilantin, Trileptal, Keppra, Hydralazine, Ativan.*" In this excerpt, all of the treatments (medications)

listed other than Hydralazine are treatments for the medical problem, "*seizures*". Since they share the same context, SACAR tends to link all of the treatments in TREATMENT-FOR relations with "*seizures*". The performance of the models could potentially be improved in the future by introducing more sophisticated representations of medical problems, treatments, and EEG events using neurological ontologies (Sahoo et al., 2014) or other sources of medical knowledge, like the Unified Medical Language System (UMLS) (Lindberg et al., 1993). For example, when determining if the concept [Lamictal]$_{TREATMENT}$ is a TREATMENT-FOR the concept [seizure]$_{MEDICAL\_PROBLEM}$, it would be beneficial to know that Lamictal is an anticonvulsant – knowledge contained in the UMLS. More sophisticated concept representations could help improve performance for the other relation types as well. The superior performance when detecting EVOKES relations may be explained by (1) the fact that EVOKES relations always involve an EEG activity and (2) the more sophisticated representation of EEG activities compared to the other medical concepts. EEG activities are explicitly characterized by their attributes in EEG-RelNet, and are *implicitly* characterized by them in SACAR due to the token representation shared between the Attribute Classifier and the Relation Detector. Specifically, EEG activities have 18 semantic attributes that capture rich information, but medical problems, treatments, and EEG events only have two attributes: modality and polarity. This suggest that semantic attributes play an important role in detecting relations between medical concepts.

Recall from Section 3.4 that the word embeddings used in SACAR are learned jointly along with the other parameters. The decision to learn word embeddings from scratch was made empirically. While the use of pre-trained word embeddings (Mikolov et al., 2013) has proved to be effective, more recent work (Peters et al., 2018; Devlin et al., 2019) has shown that pre-training entire representation layers that learn to contextualize word embeddings can be more effective. In order to determine if such a pre-training paradigm would improve SACAR, we adopted the BERT (Devlin et al., 2019) pre-training procedure to pre-train the

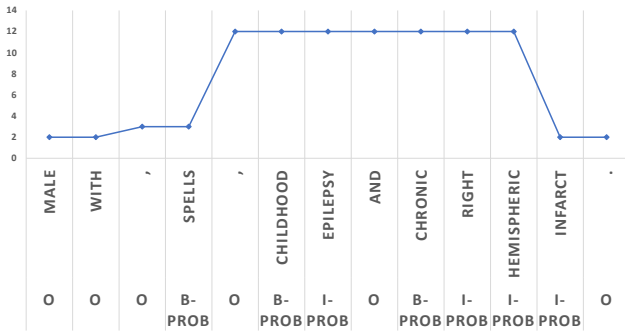

Figure 3.7. Example of the number of blocks used per word in a sentence in the Transformer Narrative Encoder, determined with Adaptive Computation Time.

Transformer Narrative Encoder. We pre-trained the TNE on the text of the entire TUH EEG corpus for 256 epochs before incorporating it into the full SACAR system. Surprisingly, this resulted in slight performance decreases across the board. We believe this is likely due to the comparatively small size of the TUH EEG corpus – just 25,000 reports – allowing the TNE to overfit. How to best pre-train SACAR remains a question for future work with possible areas of investigation including: (a) incorporating large amounts of auxiliary clinical text; (b) the use of word-piece tokenization; and (c) fine tuning a massively pre-trained model on the EEG report domain.

The results show that the SACAR-V model which did not make use of recurrence or Adaptive Computation Time consistently performed worst among the SACAR and baseline models in concept type and boundary detection, as well as relation identification. Interestingly, SACAR-V is the only model evaluated that did not make use of recurrence. However, it should be noted that, while EEG-RelNet applies recurrence sequentially, using the same recurrent cells for each input, the SACAR-U and SACAR-A models apply recurrence in a parallel manner, using separate recurrent blocks for each input.

In order to analyze the properties of Adaptive Computation Time, we graphed the number of Transformer Narrative Encoder blocks used per word in an example sentence along with the concept type and boundary labels of each word in Figure 3.7. The sentence we considered

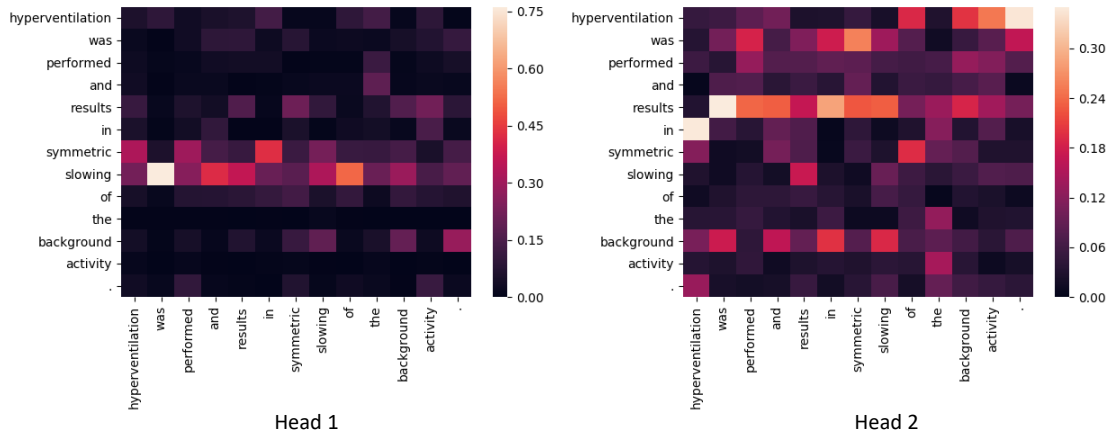

Figure 3.8. Self-Attention weights generated by the Transformer Sentence Encoder for an example sentence.

is: "*Male with, spells, childhood epilepsy and chronic right hemispheric infarct.*" We can see that ACT tends to allocate more computation time to words at the boundaries of concept mentions. ACT allocated the maximum of twelve TNE blocks to each of words 4-10 in the sentence as those words are constantly changing boundary labels, corresponding to two concept mentions, "*childhood epilepsy*" and "*chronic right hemispheric infarct*", separated by a single word, "*and*". This supports the suggestion made in Graves et al. (Graves, 2016) that ACT can learn to '*ponder*' longer on inputs that indicate implicit boundaries or transitions in sequential data.

To better understand the impact of multi-headed self-attention in SACAR, we visualized the attention weights for two attention heads produced by the last block of the Sentence Transformer Encoder for the example sentence: "*Hyperventilation was performed and results in symmetric slowing of the background activity.*" This visualization is presented in Figure 3.8. Each column in the Figure shows the attention distribution for each word in the vertical axis over the words in the horizontal axis. For the word "*slowing*", which corresponds to a mention of an EEG Activity, we notice that the attention head 1 places high attention weights on the words: "*symmetric*" – indicating Attribute 6: Dispersal = Generalized; and "*background*" – indicating Attribute 3: Background = Yes. However, for the same

word "*slowing*" the attention head 2 does not place a high weight on "*symmetric*", attending to the attribute "*background*", but it places high attention weights for the word "*results*", indicating that the two heads tend to focus on different parts of the context.

We believe that the automatic and simultaneous identification of medical concepts and relations in a large collection of EEG reports will enable the generation of EEG-specific knowledge embeddings of high accuracy. High-quality embeddings have been shown recently (Jameel et al., 2017) to be crucial in designing relevance models that rely on deep learning, and thus produce excellent results. We intend in future work to learn knowledge embeddings from the large collection of EEG reports and use them in an enhanced MERCuRY system.

It should be noted that the SACAR model poses two important weakness. First, the final SACAR model itself is comprised of 12,346,589 parameters - which is relatively small for a Transformer-based model. Recent work (Devlin et al., 2019; Radford et al., 2019; Yang et al., 2019) indicates that, given enough data, the performance of Transformer-based models, like SACAR, can be further improved by scaling the size of the model well beyond the 12 million parameters of this work. This necessitates the use of specialized hardware capable of hosting the model in GPU memory requires a large amount of time to train. Second, document-level training results in fewer training examples than sentence- or word-level training from the same amount of data. Each EEG report represents a single training example for SACAR, while the same report would yield more than ten times as many examples for sentence level training, on average. In this way, document-level training necessitates the use of silver-annotated data to adequately train SACAR, requiring pre-trained models capable of producing such silver annotations.

3.6 Summary and Lessons Learned

In this chapter, we described the task of relation extraction on EEG reports, defining four relation types between medical concepts: EVOKES, EVIDENCES, TREATMENT-FOR, and

CLINICAL-CORRELATION. Each of these four relation types can relate medical concepts anywhere in an EEG report, both in the same sentence, or across sentences, encoding clinical knowledge useful for cohort retrieval. This chapter described two deep learning architectures capable of identifying relations that span sentences or even sections in an EEG report. The EEG-RelNet is presented for recognizing long-distance relations which relies on a set of *memory* vectors that are recurrently updated as the model reads through the text of an EEG report. The Self-Attention Concept, Attribute, and Relation (SACAR) identifier is also presented in this chapter. While EEG-RelNet requires medical concepts and their attributes to be identified a priori, SACAR performs the full clinical information extraction pipeline jointly, in an end-to-end fashion.

The results indicate that both EEG-RelNet and SACAR are able to successfully detect both inter- and intra-sentential relations in EEG reports. EEG-RelNet leverages Dynamic Relational Memory and rich attribute representations of medical concepts to narrowly outperform SACAR. However, SACAR is able to jointly perform medical concept detection and attribute classification in addition to relation identification. Specifically, SACAR employs a Transformer Narrative Encoder (TNE) to perform end-to-end multi-task clinical information extraction corresponding to concept detection, attribute classification, and relation extraction with a single neural network. SACAR uses the shared representation of an EEG report produced by the TNE as input to separate prediction modules, out-performing the dedicated systems designed for concept detection and attribute classification introduced in Chapter 2, while performing competitively with EEG-RelNet for relation extraction.

CHAPTER 4

ACTIVE LEARNING OF CLINICAL CONCEPTS AND RELATIONS IN EHRS

The accurate identification of relations in clinical narratives with deep learning methods, like those presented in Chapter 3, requires large amounts of manually annotated data. However, annotating data is a resource-intensive effort requiring domain expertise – especially for complicated annotation schemata, e.g., relation identification from EEG reports. To address this issue, the field of Active Learning (AL) was developed. AL allows a learning model to achieve higher performance with fewer labeled examples by using the learning model itself to select the best unlabeled example for manual labeling (Settles, 2009). In this chapter[1][2], three active learning methodologies are presented. The first framework, **Memory-Augmented Active Deep Learning (MAADL)** is presented for the task of relation identification from EEG reports. Observing that relation identification in EEG reports requires the identification of medical concepts and their attributes in the EEG reports as well, a second active learning methodology is presented, **Multi-Task Active Deep Learning (MTADL)**, for jointly performing concept detection and attribute classification. The third framework, **Improved Multi-task Active Deep Learning with the Active Learning Policy Neural Network (MTADL+)**, is presented which uses a single multi-task learning model to perform concurrently, concept detection, attribute classification, and relation identification.

The purpose of MAADL is to train the EEG-RelNet model (defined in Section 3.3) to accurately extract relations from EEG reports with the minimum number of manually annotated EEG reports necessary. Likewise, the purpose of MTADL is to train the stacked

[1]©2019 Elsevier. Reprinted, with permission, from Ramon Maldonado and Sanda M. Harabagiu, *Active Deep Learning for the Identification of Concepts and Relations in Electroencephalography Reports*. Journal of Biomedical Informatics, Vol. 98 (2019): 103265.

[2]This chapter contains excerpts from Maldonado et al. (2017) and Maldonado et al. (2018).

LSTM and DRLN models (defined in Sections 2.4–2.5) accurately, again with the fewest manually annotated EEG reports. However, using two separate active learning paradigms for relations extraction and concept/attribute detection is inefficient, defeating the purpose of active learning in the first place. Therefore, the purpose of MTADL+ is to train a single joint model to accurately perform concept detection, attribute classification, and relation identification together by selecting the most informative EEG reports for each task for manual annotation.

All three of MAADL, MTADL, and MTADL+ are used to perform an active learning loop whereby an initial set of labeled examples is used to train learning models, and then the learning models are used to select examples for human validation from a pool of unlabeled examples. In the MAADL framework, this selection decision, called an *active learning policy*, is performed by quantifying the uncertainty EEG-RelNet has toward each unlabeled example and selecting the example that it is most uncertain about. Because the MTADL framework has several prediction tasks of interest, two concept detection tasks and 16 attribute classification tasks, its active learning policy identifies the unlabeled example the ensemble of models is most uncertain about. In the MTADL+ framework, the AL policy is *learned* from the initial set of labeled examples by the Active Learning Policy Neural Network (ALPNN). The manually annotated examples selected by the policy learned by ALPNN are used to train the SACAR model (defined in Section 3.4) with performs joint prediction of concepts, attributes and relations.

The remainder of this chapter is structured as follows: A brief background of Active Learning is provided in Section 4.1, the MAADL and MTADL frameworks are presented in Section 4.2, MTADL+ is presented in Section 4.3, and the three frameworks are evaluated in Section 4.4.

4.1 Background

Active Learning is a field of machine learning consisting of methods for training high quality learning models with fewer labeled examples. The most widely used form of active learning is *pool-based* active learning. In pool-based active learning, there is a small labeled training set, a learning model, and a large pool of unlabeled data. The learning model is iteratively improved by (1) training the model on the training set; (2) selecting unlabeled examples from the pool; and (3) manual labeling of the selected examples by a human expert (sometimes referred to as an oracle) and adding the newly labeled examples to the training set. The key hypothesis of active learning is that the internal state of a machine learning model can be used to select unlabeled examples whose labels would be especially beneficial to the learning model (Settles, 2009). These methods are of practical importance because human labeling is a time and resource intensive practice, especially for complex labeling schemata such as those presented in Chapters 2 and 3 which require domain expertise on the part of the oracle. As such, active learning can be employed to minimize the overall labeling cost of an annotation task by ensuring only the most informative unlabeled examples are considered for manual labeling.

The efficacy of an active learning system is determined by its active learning selection policy, i.e., the strategy used to select unlabeled examples for manual labeling. Uncertainty sampling (Lewis and Gale, 1994) is one of the most commonly used selection policies due to its simplicity and effectiveness (Settles, 2009). Uncertainty sampling quantifies the *uncertainty* of a classifier with regards to an unlabeled example by using the classifier's probability distribution over predicted labels. Other families of sampling mechanisms include query-by-committee (Seung et al., 1992), expected model change (Settles et al., 2008), and expected error reduction (Roy and McCallum, 2001). Each of these sampling paradigms is defined with respect to a single prediction task. However, in this work we consider complex information extraction problems involving multiple sub-tasks as in Chapters 2 and 3. To apply active

learning in the multi-task domain, Reichart et al. (2008) introduce the alternating selection and rank combination protocols. These methods extend traditional sampling mechanisms to the multi-task domain by considering the per-task selection criteria for each example and either alternating between them (alternating selection) or combining them (rank combination) to select examples. The rank combination protocol has been shown to be effective for multi-task neural methods for tasks such as semantic role labeling and entity recognition (Ikhwantri et al., 2018). More recently, neural methods have been introduced for *learning* an active learning selection policy from data, as opposed to using a heuristic. Bachman et al. (2017) present a method for a learning how to represent an active learning problem in the form of a data representation, an selection policy, and a method for constructing prediction functions from labeled training sets. Konyushkova et al. (2017) train a regressor to predict the expected error reduction of incorporating any unlabeled example and use that regressor as a selection policy. Liu et al. (2018a) present an imitation learning approach whereby the selection policy is a neural network trained to imitate an optimal selector on a series of simulated active learning problems.

Past work has shown that AL methods can be used to enhance supervised NLP methods operating on health narratives. For example, AL was used to annotate pathological phenomena in Medline abstracts in the PathoJen system (Hahn et al., 310) by relying on a set of classifiers and computing the disagreement between them to inform the selection of the annotation that needs to be validated or edited. AL has also been used for high-throughput phenotyping algorithms. By integrating AL with SVM-based classifiers, the research published by Chen et al. (Chen et al., 2013) showed that AL can reduce the number of sampled annotations required for achieving an area under the curve (AUC) of 0.95. Dligach et al. (2013) used AL with Naive Bayes classification for extracting phenotypes and observed that AL generated a significant reduction in annotation efforts: only one third of annotations were required. More recently, a new study targeting a cost-sensitive AL for clinical phenotyping

was published by Ji et al. (Ji et al., 838), using the identification of breast cancer patients as a use case. The cost model was generated based on linear regression using some heuristic features, and used to maximize the informativeness/cost ratio when selecting samples for validation/editing. However, Ji et al. (838) rely on a simple logistic regression model as their learner. In this chapter, we show the benefit of combining state-of-the-art deep learning models with active learning for information extraction in clinical narratives.

4.2 Active Deep Learning of Concepts, Attributes, and Relations in EEG Reports

This section presents two active deep learning systems for information extraction from EEG reports: (1) the Memory-Augmented Active Deep Learning (MAADL) system for relation identification in EEG reports and (2) the Multi-task Active Deep Learning (MTADL) system aiming to perform concurrently multiple annotation tasks corresponding to the identification of medical concepts and their attributes. In order to perform relation identification, MAADL relies on pre-identified concepts and attributes. Therefore, MAADL relies on the MTADL system for concept detection and attribute classification. Both MAADL and MTADL make use of a 5-step process designed to be applied in general to any information extraction tasks on a large dataset of clinical notes, however in this section we describe the application of this process to the TUH EEG corpus described in Chapters 2 and 3. While MAADL uses standard uncertainty sampling, the selection policy of MTADL is designed to be robust to multiple concurrent annotation tasks, which is necessary for operating on the EEG reports of the TUH EEG corpus. This section begins with the description of the 5-step process for active deep learning shared by MAADL and MTADL. Because the relation identification of MAADL requires pre-identified concepts and attributes, MTADL is specified first, followed by MAADL.

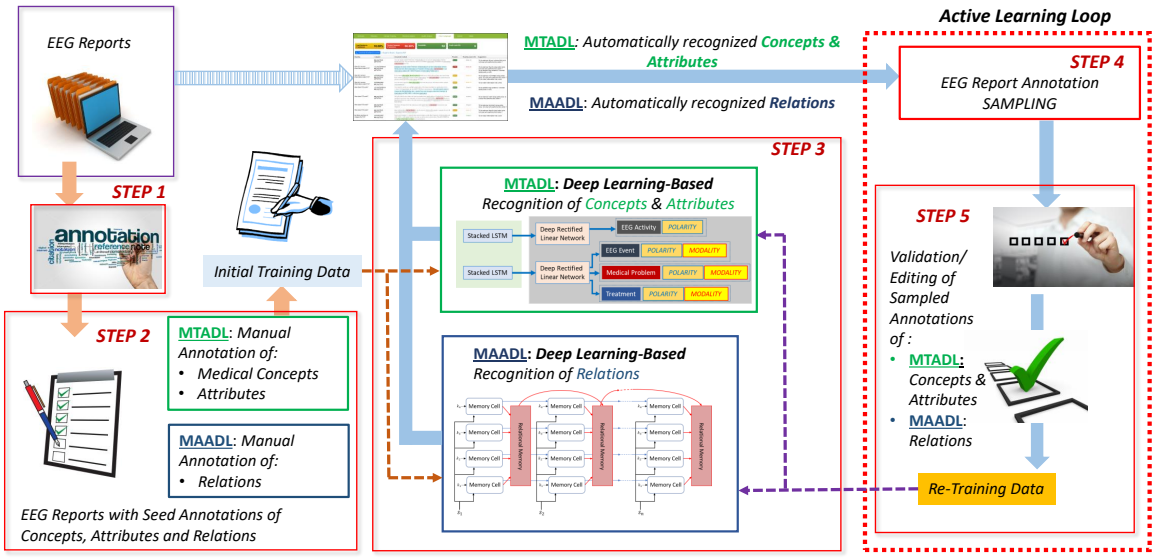

Figure 4.1. The Multi-task (MTADL) and Memory-Augmented (MAADL) Active Deep Learning Systems for Annotating EEG Reports. The modules specific to MTADL are highlighted in green, while the modules specific to MAADL are highlighted in dark blue.

4.2.1 Five-Step Active Deep Learning Architecture for Automatically Annotating EEG Reports

The MTADL and MAADL paradigms, depicted in Figure 4.1, consists of the following five steps:

Step 1: The development of an annotation schema

Step 2: Annotation of initial training data

Step 3: Design of deep learning methods that are capable to be trained on the data

Step 4: Development of the sampling mechanism

Step 5: Usage of the active learning system which involves:

Step 5.a: Accepting/Editing annotation of sampled examples

Step 5.b: Re-training the deep learning models

It should be noted that while both MAADL and MTADL employ the same abstract 5-step process, each system requires a specific implementation. These steps are described in detail in the remainder of this section, first for MTADL, then for MAADL.

4.2.2 Multi-task Active Deep Learning for Concept Detection and Attribute Classification in EEG Reports

To identify medical concepts and attributes in EEG reports, we apply the five-step active deep learning methodology, depicted in Figure 4.1, via the following implementation:

STEP 1: Annotation Schema: The first step is the development of an annotation schema. The annotation schemata used by MTADL for concept detection and attribute classification are defined in Section 2.2. Recall from Chapter 2 that the EEG corpus contains mentions of medical concepts including EEG activities, EEG events, medical problems, and treatments. Recall further that each of these medical concepts is characterized by a set of annotations including modality and polarity and a set of 16 EEG-activity specific attributes. Because of the complex surface forms in which concept attributes are expressed, EEG activities are identified by their *anchor* which is defined by the text corresponding to an activity's MORPHOLOGY attribute. MTADL will train its classifiers (defined in Step 3) to extract information from the free text of the EEG reports according to this schema.

STEP 2: Annotation of Initial Training Data: Initially, a sub-set of 40 EEG reports was manually annotated. The annotations were created by first running the medical concept recognition system reported in (Roberts and Harabagiu, 2011) to detect medical problems, tests, and treatments and their polarity and modality. The annotations that were obtained were manually inspected and edited, while manual annotations for EEG Activities and EEG events, as well as their attributes, are also generated in the sub-set of 40 EEG reports. The initial annotations represented the initial set of training data for two deep learning architectures, as illustrated in Figure 4.1.

STEP 3: Design of Deep Learning Architectures: The first architecture aims to identify (1) the anchors of all EEG activities mentioned in an EEG report; as well as (2) the boundaries of all mentions of EEG events, medical problems, medical treatments and medical tests. To this end we employ two stacked LSTM models described in Section 2.4, one for detecting EEG activity anchors and one for detecting the boundaries of all other medical concepts. We employ a second deep learning architecture, the Deep ReLU Network (DRN), which is designed to recognize (i) the sixteen attributes that we have considered for each EEG activity, as well as (ii) the type of the EEG-specific medical concepts, discriminated as either an EEG event, a medical problem, a medical test or a medical treatment. In addition, the DRN identifies the modality and the polarity of these concepts. The DRN is described in Section 2.5. As with the boundary detection models, we employ two DRNs, one for EEG activity attributes and one for the attributes and types of all other medical concepts.

After training the stacked LSTMs and DRNs on the initial training data obtained with manual annotations, we were able to automatically annotate concepts and attributes in the entire corpus of EEG reports. Because these automatically created annotations are not always correct, we developed an active learning framework to validate and edit these annotations, and provide new training data for the deep learning architectures, which are iteratively refined as active learning progresses.

STEP 4: Development of the Sampling Mechanism The choice of sampling mechanism is crucial for validation as it determines what makes one annotation a better candidate for validation over another. MTADL is an active learning paradigm for multiple annotation tasks where new EEG reports are selected to be as informative as possible for a set of annotation tasks instead of a single annotation task. The sampling mechanism that we designed used uncertainty sampling (Lewis and Gale, 1994) with the rank combination protocol (Reichart et al., 2008), which combines several single-task active learning selection decisions into one. The usefulness score, $s_{X_j}(a)$ of each un-validated annotation a from an EEG Report is

calculated with respect to each annotation task X_j and then translated into a rank $r_{X_j}(a)$ where higher usefulness means lower rank (examples with identical scores are assigned the same rank). Then, for each EEG Report, we sum the ranks of each annotation task to get the overall rank $\sum_{j=1} r_{x_j}(a)$. All examples are sorted by this combined rank and annotations with lowest ranks are selected for validation. For each annotation task, we score an EEG Report $d : s_{x_j}(d) = \frac{1}{|d|} \sum_{a \in d} H(a)$ where a is an annotation from d and $|d|$ is the number of annotations in document d, and $H(a) = \sum_c q_c^a \log q_c^a$ is the Shannon Entropy of a. Shannon Entropy (Shannon, 1948) is an information theoretic method for quantifying the uncertainty expressed in a probability distribution. By considering the rank of each task for an example as opposed to the raw Shannon Entropy scores, the rank combination protocol ensures that no single task is more important than the others while performing sampling. As such, this protocol favors selecting documents containing annotations the model is uncertain about from all annotation tasks.

STEP 5: Usage of the Multi-Task Active Deep Learning System We performed several active learning sessions with our deep learning architectures. At each iteration, the deep learners are trained to predict annotations using the new validations in addition to the previous training data. This process is repeated until (a) the error rate is acceptable; and (b) the number of validated examples is acceptable.

4.2.3 Memory-Augmented Active Deep Learning for Identifying Relations Between Medical Concepts in EEG Reports

While MTADL uses the rank combination protocol to select unlabeled examples which are informative for concept detection and attribute classification tasks, incorporating relations into this selection policy is not trivial. There is only a single relation identification task, while there are two concept detection tasks and eighteen attribute classification tasks, so it is likely that the rank combination protocol will simply ignore relation identification when

selecting examples. Therefore, we present the Memory-Augmented Active Deep Learning (MAADL) system for relation identification in EEG reports. The five steps of MAADL, depicted in Figure 4.1, are defined as:

STEP 1: Annotation Schema: For relation identification, we relied on the schema defined in Section 3.2, focusing on three types of relations between medical concepts found in EEG reports: EVOKES, EVIDENCES, and TREATMENT-FOR.

STEP 2: Annotation of Initial Training Data: In order to perform relation identification, medical concepts and their attributes must be identified first. Therefore, we relied on the same sub-set of 40 manually annotated EEG reports from Step 2 of MTADL, annotating relations on the same set. 198 EVIDENCES, 146 EVOKES, and 72 TREATMENT-FOR relations were identified in the seed set of 40 EEG reports.

STEP 3: Design of Deep Learning Architectures: For relation identification we employ EEG-RelNet, which detects relations between medical concepts in each EEG report by using a neural network augmented with two types of memories: (i) a memory for each medical concept; and (ii) a memory for each relation between each pair of medical concepts. EEG-RelNet is fully specified in Section 3.3.

STEP 4: Development of the Sampling Mechanism: Since MAADL is focused on relation detection between pairs of medical concepts, we chose a selection policy that only prioritizes relation detection performance, ignoring the quality of medical concepts and their attributes. Therefore, we do not use the rank combination protocol reported of MTADL, opting for standard uncertainty sampling (Settles, 2009) whereby EEG reports containing relations for which the model is most uncertain are selected for manual validation. The uncertainty of a report is measured at the report level by averaging the uncertainty of each relation classification decision in the report. The uncertainty of a relation classification decision is calculated using Shannon Entropy, $H(R) = -\sum_t R_t \log R_t$, where R is a vector representing the probability distribution over possible relation types. These probability

distributions are derived by EEG-RelNet from the learned *dynamic relation memory*, as defined in Equation 3.8 in Section 3.3.

STEP 5: Usage of the Memory Augmented Active Deep Learning System: As in MTADL, the active learning loop is iterated until (a) the error rate is acceptable; and (b) the number of validated examples is acceptable.

The MTADL and MAADL frameworks allowed us to rely on the active learning loop to enhance the quality of the concepts, attributes and relations we have discovered automatically in the EEG reports. However, while active learning with the rank combination protocol is useful for the task of training separate classifiers on the same training examples, it does have drawbacks. For example, it is likely that if one classifier is uncertain about an example, other classifiers are also likely to be uncertain because the tasks are heavily correlated, causing the rank combination protocol to inflate the importance of such examples. In general, selecting the most informative unlabeled examples is a complex and difficult task, so heuristics like uncertainty sampling and rank combination will only go so far. Moreover, have two separate frameworks for active learning from the same corpus is inefficient, providing the possibility of manually annotating the same EEG report twice. In the next section, a method for *learning* a sampling mechanism from data to make these complex selection decisions over all three annotation problems is presented.

4.3 Improved Multi-task Active Deep Learning with the Active Learning Policy Neural Network

In this section we present a new paradigm for Multi-task Active Deep Learning of Concepts, Attributes, and Relations using the Active Learning Policy Neural Network. The Active Learning Policy Neural Network (ALPNN) is a meta-learning network that *learns* an active

learning selection policy from data by training on a series of simulated active learning problems. The improved paradigm for Multi-task Active Deep Learning of concepts, attributes, and relations in EEG reports (MTADL+) boasts several improvements over MTADL presented in Section 4.2 including (1) the joint prediction of boundary detection, attribute classification, and relation identification; (2) the learning of a sampling mechanism by ALPNN; and (3) the usage of a single end-to-end neural model for identifying concepts, attributes, and relations, the SACAR model. The SACAR model is defined in Section 3.4. MTADL+ improves active learning over MTADL by eschewing the rank combination protocol for the learned selection policy of ALPNN. Combining the uncertainty of the medical concepts and their attributes in an EEG report with the uncertainty of the relations between medical concepts in the same EEG report is not a trivial task. Therefore we chose to *learn* the EEG report selection strategy best suited for the identification of concepts, attributes and relations in our corpus of EEG reports with ALPNN. As in Liu et al. (Liu et al., 2018b) the problem of learning the example selection policy is cast as a imitation learning problem. The usage of ALPNN is enabled by the multi-task SACAR model since ALPNN is able to represent an unlabeled example by its internal representation from SACAR, instead of as a series of probability distributions over predictions (as in the rank combination sampling mechanism of MTADL). As shown in the Figure 4.2, MTADL+ uses the following six steps:

Step 1: The *initial manual annotation* of medical concepts, attributes and relations between medical concepts in EEG reports. In addition to the expert annotations used before in Section 4.2, we also made use of *silver annotations*, produced on the entire corpus of EEG reports, using the previous methods for concept and attribute recognition as well as relation identification

Step 2: *Learn to recognize concepts, their attributes and relations between concepts* by training SACAR on the current manually annotated training data along with the silver annotations

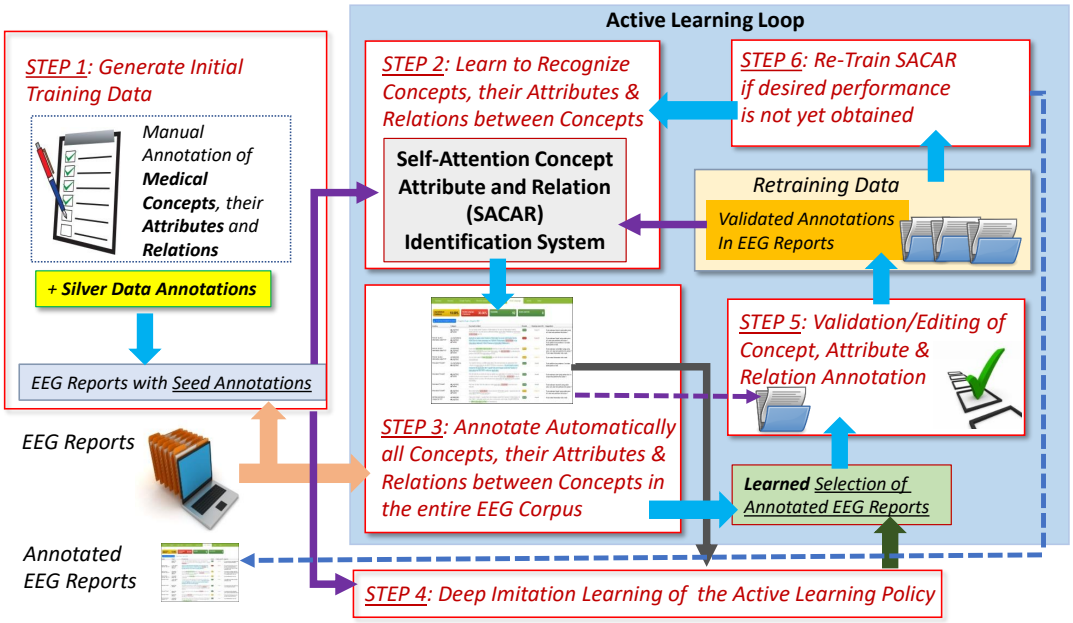

Figure 4.2. Multi-Task Active Deep Learning of Concepts, Attributes and Relations from EEG Reports.

Step 3: *Automatically annotate all concepts, their attributes and relations between concepts* given the current SACAR model

Step 4: *Deep imitation learning of the active learning policy* is performed if a selection policy is not yet learned. The learned policy is then applied on the entire set of un-annotated EEG reports, given the current state of the SACAR model. This step is performed only once, while the resulting learned selection policy of the annotated EEG reports will be used repeatedly, inside the AL loop, as illustrated in Figure 4.2

Step 5: *Accept/Edit annotations* of concepts, attributes and relations in sampled EEG reports, made available by the learned selection of annotated EEG reports

Step 6: *Re-training* SACAR with the new training data, containing the validated annotations. Go to Step 3 until the desired performance is obtained or the time for active learning is exhausted.

As shown in Figure 4.2, the active learning loop is comprised of Steps 2, 3, 5 and 6. Central to MTADL+ is the active learning policy which, once learned in Step 4, it is applied in the AL loop in a static fashion.

4.3.1 Deep Imitation Learning of the Active Learning Policy

To select unlabeled examples for manual annotation in the active learning loop, we used a neural network, namely the *the Active Learning Policy Neural Network (ALPNN)*, illustrated in Figure 4.3, which we trained with the deep imitation learning algorithm developed by Liu et al. (2018b). The ALPNN represents an active learning problem in terms of (1) a model (in this case the SACAR model); (2) a set of labeled data for training, D_T; (3) a set of labeled data for evaluation, D_E; and (4) a set of unlabeled data D_U. Given an AL problem, the Policy Network determines a score for each unlabeled data example and returns the example with the highest score for manual annotation.

In order to train the ALPNN, we use the initial manually annotated dataset, D, to develop a series of *simulated active learning* problems. Each simulated active learning problem consists of (1) three random partitions of D to form (a) the training data, D_T, (b) the evaluation data D_E, and (c) the unlabeled data, D_U; along with (2) the SACAR model. The ALPNN is trained to select the optimal unlabeled example for each simulated problem. To determine the optimal selection, K examples $X_1, X_2, \ldots X_K$ are randomly selected from D_U and K different SACAR models are trained using D_T augmented with one of the examples, e.g., SACAR$_i$ is trained using D_T augmented with X_i, for each $i \in \{1 \ldots K\}$. When evaluating each of the K SACAR models using the evaluation data D_E, we were able to determine the SACAR model with the best performance, e.g., SACAR$_j$ and thus conclude that example X_j is the the optimal selection. In this way, the ALPNN is trained to *imitate* an expert selection policy that selects the unlabeled example that will most improve the performance of SACAR.

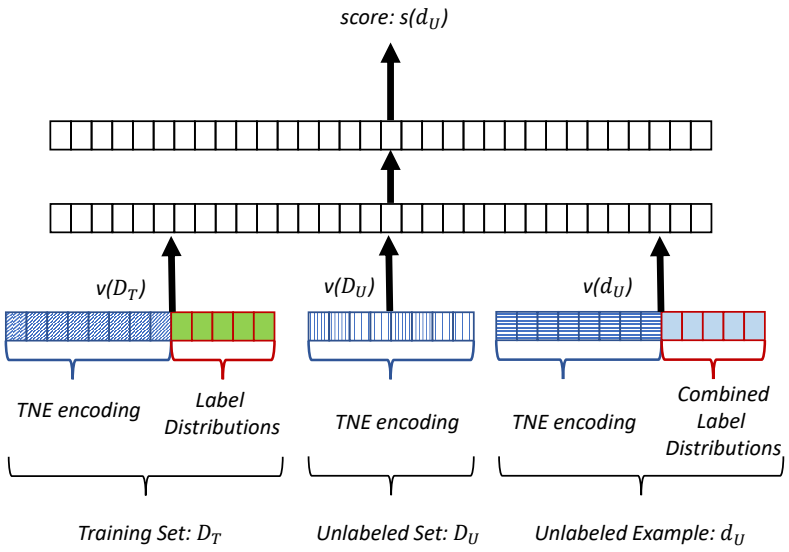

Figure 4.3. The Active Learning Policy Neural Network (ALPNN).

The ALPNN is a two-layer feed-forward neural network that calculates a score, $s(d_U)$, for an unlabeled example, d_U, given an active learning problem. The inputs to the ALPNN, as shown in Figure 4.3 consist of 3 fixed-size vector representations: (1) $v(D_U)$ representing the entire unlabeled data set, D_U; (2) $v(D_T)$ representing the entire labeled data set, D_T; and (3) $v(d_U)$ representing the unlabeled example, d_U, along with the predicted labels for it, $\hat{y}_U$, generated by SACAR. Specifically, $v(D_U)$, the vector representation of the unlabeled data set D_U, is produced by aggregating the encodings for each example, $d \in D_U$:

$$v(D_U) = \sum_{d \in D_U} \frac{1}{|d|} \sum_{w_i \in d} b_i \tag{4.1}$$

where b_i is the encoding produced by SACAR's TNE for word w_i in the EEG report d; and $|d|$ is the number of words in d. Similarly, the vector representation, $v(D_T)$, for the labeled data set consists of an aggregation of (i) the encodings for each example $d \in D_T$ concatenated with vectors for the empirical distributions of class labels for (ii) the concept type & boundary detection task; (iii) the 18 attribute classification tasks, and (iv) the relation prediction task

in the labeled data set:

$$x = \sum_{d \in D_T} \frac{1}{|d|} \sum_{w_i \in d} b_i \tag{4.2}$$

$$y_l^t = P(t = l | D_T) \tag{4.3}$$

$$v(D_T) = [x, y^1, \ldots, y^{|t|}] \tag{4.4}$$

where x is the sum of the average TNE encoding for each word in each example $d \in D_T$; y_t^a is the probability of task t having class l under the empirical distribution of D_T; and v_T is the concatenation of x and all of the distributions, y^t for each task t. The vector representation $v(d_U)$ of the unlabeled example, d_U, along with its predicted labels, $\hat{y}_U$, is derived as follows:

$$v_x = \frac{1}{|d_U|} \sum_{w_i \in d} b_i \tag{4.5}$$

$$v_y^t = \sum_i p_i^t \tag{4.6}$$

$$v(d_U) = [v_x, v_y^1, \ldots, v_y^{|t|}] \tag{4.7}$$

where v_x is the average TNE encoding for each word in the example d_U, v_y^t is the sum of the predicted distributions p_i^t (from Equations 3.26, 3.28, and 3.34 in Section 3.4) for each instance of task t in d_U, and $v(d_U)$ is the concatenation of v_x and all of the combined predicted distributions v_y^t for each task t.

The score $s(d_U)$, illustrated in Figure 4.3, is then calculated as:

$$s(d_U) = W_P^1(W_P^0[v(D_U), v(D_T), v(d_U)] + b_P^0) + b_P^1 \tag{4.8}$$

where $W_P^0 \in \mathbb{R}^{d_p \times d_0}$ and $W_P^1 \in \mathbb{R}^{1 \times d_p}$ are weight matrices, d_p is the dimension of the hidden state, d_0 is the dimension of the three input vectors concatenated together, and $b_P^0 \in \mathbb{R}^{d_p}, b_P^1 \in \mathbb{R}$ are bias vectors.

The ALPNN is trained using the imitation learning algorithm described in Liu et al. (2018b) using the following loss function which maximizes the probability of selecting the

optimal unlabeled example, $\hat{d_U}$, from D_U for each simulated problem:

$$\mathcal{L}_P = - \sum_{\substack{(\hat{d_u}, D_U, D_T, D_E) \\ \in SIM}} \frac{\exp s(\hat{d_U})}{\sum_{d_U \in r(D_U)} \exp s(d_U)} \tag{4.9}$$

where the tuple $(\hat{d_u}, D_U, D_T, D_E)$ represents a simulated active learning problem, and SIM is the set of simulated examples. The set SIM is dynamically generated during training of ALPNN using the algorithm described in Liu et al. (2018b), which uses the Dataset Aggregation method (Ross et al., 2011) which is meant to increase the generalization of the Policy Network by exposing it to problems similar to those it is likely to encounter during the active learning loop.

4.4 Experimental Results and Discussions

The impact of the active learning systems presented in this chapter is evaluated by measuring the change in performance after each additional round of active learning. The active learning systems are evaluated for 10 rounds where 10 unlabeled EEG reports are sampled from the entire unlabeled pool for manual annotation, starting with a seed set of 40 labeled documents. Results are presented in terms of F_1 score using 7-fold cross validation. We compare the MTADL and MAADL systems (Section 4.2) against MTADL+ (Section 4.3) and against a random sampling baseline (RAND) using the SACAR model as its learner. Specifically, the active learning policy of the random baseline is simply to select an unlabeled example completely at random.

Figure 4.4 presents the learning curves for all tasks, macro-averaged. Since MAADL performs only relation identification and MTADL performs only concept detection and attribute classification, their performances are averaged to compare against MTADL+ and the random sampling baseline. Figure 4.4 clearly shows that both MTADL+ and MTADL outperform random sampling, with MTADL+ achieving the best performance. In fact, MTADL+

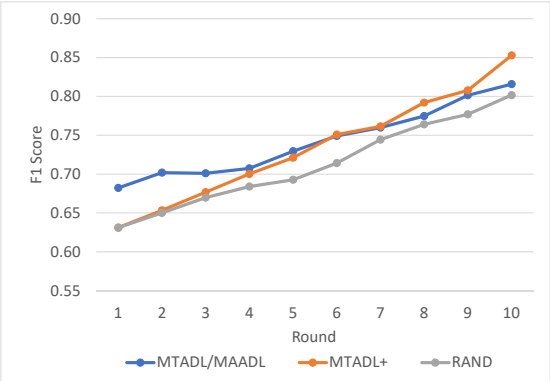

Figure 4.4. Aggregate learning curves for all tasks (macro-averaged) for 10 rounds of active learning measured by F_1 Score.

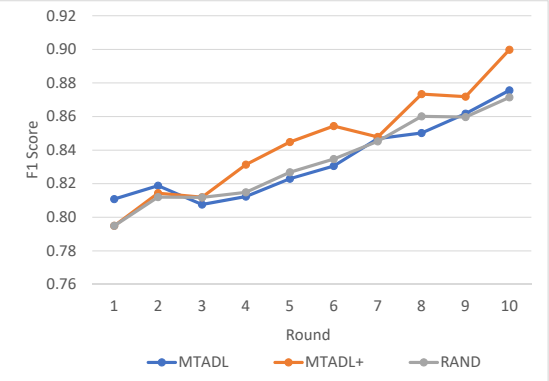

Figure 4.5. Learning curves for concept detection for 10 rounds of active learning measured by F_1 Score.

achieves a 35% relative increase over the 10 rounds of active learning, while MTADL achieves 20% and random sampling using SACAR achieves 27%. The improved relative increases of ALPNN for each task compared to random sampling illustrates the effectiveness of the approach. Moreover, we found that MTADL+ selects documents with slightly fewer concepts and a similar number of relations compared to random sampling. Specifically, the average document sampled by MTADL+ contained 24.54 concepts and 7.57 relations while the average randomly sampled document contained 27.20 concepts with 7.89 relations. This indicates that the policy learned by ALPNN doesn't simply bias towards longer documents with more annotations to trivially outperform random sampling.

Figure 4.5 presents the learning curves for concept detection, while Figure 4.6 presents the learning curves for attribute classification, and Figure 4.7 presents the learning curves for relation identification. For concept detection, we see a slight dip in performance as active learning begins for MTADL, while it recovers to achieve an overall increase of 8% over the 10 rounds. The performance of MTADL+ begins lower than that of MTADL, but quickly outpaces it, achieving a relative increase of 13%. Similarly for attribute classification, MTADL+ quickly surpasses the performance of MTADL, achieving a relative increase in F_1 score of 47% vs. 33%. For relation identification, we compare the joint learning of MTADL+

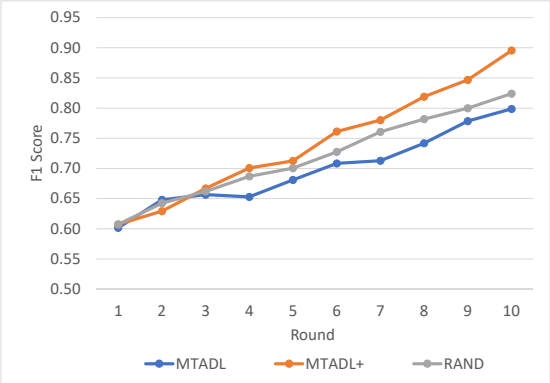

Figure 4.6. Learning curves for attribute classification for 10 rounds of active learning measured by F_1 Score.

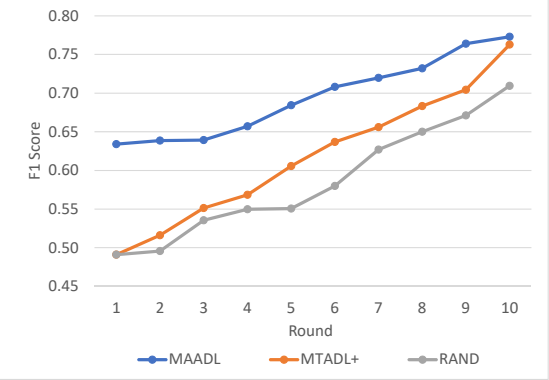

Figure 4.7. Learning curves for relation identification for 10 rounds of active learning measured by F_1 Score.

against the more focused, single-task MAADL system. Compared to MAADL, MTADL+ starts with much lower initial performance for relation identification. This could be due to the fact that the underlying model of ALPNN, SACAR, is performing concept boundary detection and attribute classification in addition to relation identification while MAADL's model is able to focus on relation identification alone. While the SACAR model used by ALPNN is just starting active learning, it focuses on concept boundaries and attributes as there are more concepts and attributes than relations to be identified, dominating the loss function. However, as the performance for concepts and attributes increases, so to does the performance for relation identification - to the point where the performance is comparable to that of MAADL. In this way, ALPNN is able to match the performance of the dedicated relation identification system while also out-performing another dedicated system for concept boundary and attribute identification at the same time.

4.5 Lesson Learned

In this chapter three active deep learning systems are presented for training deep learners to extract information from clinical text. The MAADL system improves relation identification

quality while the MTADL system improves concept detection and attribute classification. Both of these systems select unlabeled examples for manual annotation by quantifying the uncertainty that the underlying deep learners have about the unlabeled examples. However, when a single multi-task neural network which performs concept detection, attribute classification, and relation identification is provided, a new active learning paradigm is enabled, MTADL+. MTADL+ is able to learn an active learning selection policy from data and the resulting policy is shown to outperform both uncertainty sampling and random sampling in experiments.

CHAPTER 5

DEEP LEARNING OF BIOMEDICAL KNOWLEDGE EMBEDDINGS

Biomedical ontologies encoding biomedical knowledge have been a major focus of the biomedical research community over the past two decades, justified by the steady increase in biological and biomedical research and the growth of data that is being collected in all areas of biology and medicine. As the number of ontologies increases and their size grows, their relevance in biomedical research also rises as they contribute to the interpretation of the biomedical data and enable complex inference over their encoded knowledge. The BioPortal[1] of the National Center for Biomedical Ontology (NCBO) is the most comprehensive repository of biomedical ontologies in the world (as of this writing it includes 817 ontologies, with over 10 million classes and almost 40 million indexed records). Many of the ontologies available from the BioPortal have become widely used resources, e.g., the Gene Ontology (Ashburner et al., 2000) (GO), one of the most important resources available in genomics research. For instance, survey published in Huang et al. (2009) discusses 68 bioinformatics enrichment tools informed by GO, that have played a very important and successful role contributing to the gene functional analysis of large gene lists for various high-throughput biological studies, evidenced by thousands of publications citing these tools. Moreover, LePendu et al. (2011) showed that it is possible to create reference annotation sets for enrichment analysis [2] using other ontologies than GO, still available from BioPortal, e.g., the Human Disease Ontology (DO). The ontologies in the BioPortal define graph-theoretic structures, with concepts connected by edges representing relations such as 'Is-A' or 'Part-Of' or others from the OBO Relation Ontology (RO), generating well-principled ontologies for many biomedical domains.

[1] http://bioportal.bioontology.org

[2] Gene set enrichment (also functional enrichment analysis) is a method to identify classes of genes or proteins that are over-represented in a large set of genes or proteins, and may have an association with disease phenotypes.

The knowledge encoded in the ontologies available through the BioPortal of the National Center for Biomedical Ontology (NCBO), although informative and originating from biomedical expertise, cannot be used easily, especially when considering techniques informed by deep learning similar to those presented in Miotto et al. (2016) and Rajkomar et al. (2018). The main difficulty arises from the fact that deep learning techniques operate on special representations, called embeddings, while ontologies encode knowledge differently, describing concepts and relations between them. Therefore, in order to take advantage of the vast knowledge encoded in biomedical ontologies, *knowledge embeddings* have been considered, which can be seamlessly integrated into deep learning systems. Knowledge embeddings are continuous vector representations of concepts and relations representing a knowledge graph that encode the inherent structure of the graph. This chapter[3] presents three different frameworks of generating biomedical knowledge embeddings:

Framework 1: in which knowledge embeddings are used to represent instances of concepts encoded in a specific ontology;

Framework 2: in which knowledge embeddings represent concepts as well as relations from a very large ontology widely used in medical informatics; and

Framework 3: in which knowledge embeddings provide an alignment between separate ontologies, thus enabling the incorporation of knowledge that overlaps in separate ontologies.

In Framework 1, biomedical knowledge discovered from a large corpus of EEG reports is used to augment an existing ontology developed for encoding knowledge relevant to epilepsy. Framework 2 presents novel knowledge graph embedding methods operating on a very large biomedical ontology, namely the Unified Medical Language System (UMLS) (Lindberg et al.,

[3]This chapter contains excerpts from Maldonado et al. (2017), Maldonado et al. (2019), and Maldonado and Harabagiu (2019).

1993). In Framework 3, knowledge embedding methods are extended to jointly model disparate ontologies, and enable their alignment using a shared embedding space.

The purpose of Framework 1 is to (a) provide new ontologies with instances of relations involving concepts encoded in the ontology and (b) identify new data-driven knowledge that should be included in the ontology. The purpose of Framework 2 is to expose the expert-curated knowledge found in the UMLS to deep learning systems. As such, Framework 2 informs a predictive model using deep learning based on a hierarchical attention mechanism using the knowledge contained in the UMLS embeddings. The purpose of Framework 3 is to perform *ontology alignment*. Ontology alignment is the task of finding correspondences between concepts in disparate ontologies. Many biomedical ontologies are developed in isolation and therefore exhibit subsets of overlapping information, imposing the task of ontology alignment.

In Framework 1, techniques from Chapters 2 and 3 are used to construct a data-driven knowledge graph by identifying instances of concepts and relations between concepts throughout the corpus. The concepts in this knowledge graph are linked against an existing ontology, and embedded to allow for probabilistic inference. Framework 2 is comprised of a novel embedding method based on Generative Adversarial Networks (GAN) (Goodfellow et al., 2014) operating on distinct graphs within the UMLS. Framework 3 extends the embedding methods of Framework 2 to jointly perform alignment and embedding.

The knowledge embeddings produced by Framework 1 can be used to perform inference, identifying the most likely relations expressed in the data that should be added to the ontology. Framework 2 produces embeddings for each concept in the UMLS – these concept embeddings can be used to characterize medical concepts in deep learning systems. As an example, a clinical prediction model is augmented with the UMLS concept embeddings, improving results. Framework 3 is applied to the task of aligning three specific biomedical ontologies, showing promising results.

This chapter is organized as follows: Section 5.1 provides background on knowledge graph embedding methods, Section 5.2 presents Framework 1 for learning knowledge embeddings, while Section 5.3 presents Framework 2, Section 5.4 presents an application of Framework 2, and finally Section 5.5 presents Framework 3 for learning knowledge embeddings for ontology alignment.

5.1 Background

Knowledge graph embedding is the task of encoding the relational structure of a multi-relational knowledge graph into a set of real-valued vectors (Wang et al., 2014). In a multi-relational knowledge graph $G = (V, E)$, vertices, V, represent concepts and edges, E, represent relations between concepts in the form of relation triples, $\langle c_1, r, c_2 \rangle$ where $c_1, c_2 \in V$ are concepts and r is a relation from a fixed set R. For example, the edge $\langle \text{cancer}, \text{IS\_A}, \text{disease} \rangle$ represents a hierarchical "IS_A" relation between the biomedical concepts "cancer" and "disease". Given a knowledge graph, vectors $c \in V$ and $r \in R$ are learned for each concept and relation, respectively. These vectors are referred to as *knowledge embeddings* since they encode the relational structure of the knowledge graph and thus the knowledge contained therein. Using the relational information encoded in knowledge graphs directly is known to be difficult and inefficient, especially as the size of the graph grows (Wang et al., 2014; Nickel et al., 2011). Knowledge embeddings facilitate the usage of this relational knowledge since it allows for subsets of concepts and relations to be manipulated in isolation, while still maintaining information about their context within the graph as a whole (Wang et al., 2014). To this end Nickel et al. (2011) introduced RESCAL, a tensor factorization method that derives a factorization of the multi-relational adjacency matrix such that each concept and relation is a separate learned factor. Bordes et al. (2013) reformulate the problem by projecting concepts onto a shared embeddings space and casting relations as *translations* between concepts on that embedding space, introducing TransE. Socher et al. (2013) introduce a

more expressive model, the Neural Tensor Network, that learns a series of weight matrices for each relation type along with concept embeddings. Building on the success of TransE, various other methods were more recently introduced that project concept embeddings onto more structured embedding spaces with various geometric properties (Wang et al., 2014; Lin et al., 2015; Ji et al., 2015a). Noting that concepts found in knowledge graphs are often characterized by information that is not contained within the graphical structure itself, Guo et al. (2015) use concept type information to impose smoothing constraints on knowledge embeddings and Trivedi et al. (2018) incorporate attribute information into their embedding scheme. As we will see later in the chapter, these methods are particularly well-suited to biomedical ontologies, which contain rich attribute information. Knowledge embeddings learned by these techniques have wide-ranging uses including discovering new relations in a knowledge graph (Bordes et al., 2013), question answering (Bordes et al., 2014), and relation extraction (Weston et al., 2013). In this chapter knowledge embeddings are applied to probabilistic question answering in Section 5.2, predictive modeling in Section 5.4, and ontology alignment in Section 5.5.

5.2 Knowledge Embeddings Meets Biomedical Ontologies

This section describes the *medical knowledge embeddings* (MKE) automatically learned from a large corpus of EHRs, specifically the Temple University Hospital (TUH) EEG Corpus described in Chapter 2. EEG reports contain a wealth of epilepsy-related knowledge, derived from clinical practice, expressed through narratives describing medical concepts and implicit relations between these concepts. The methodology described in this section links medical concepts throughout the corpus against an existing biomedical ontology from BioPortal, the Epilepsy Syndrome and Seizure Ontology[4] (ESSO). ESSO encodes 2,705 classes

[4] `http://bioportal.bioontology.org/ontologies/ESSO`

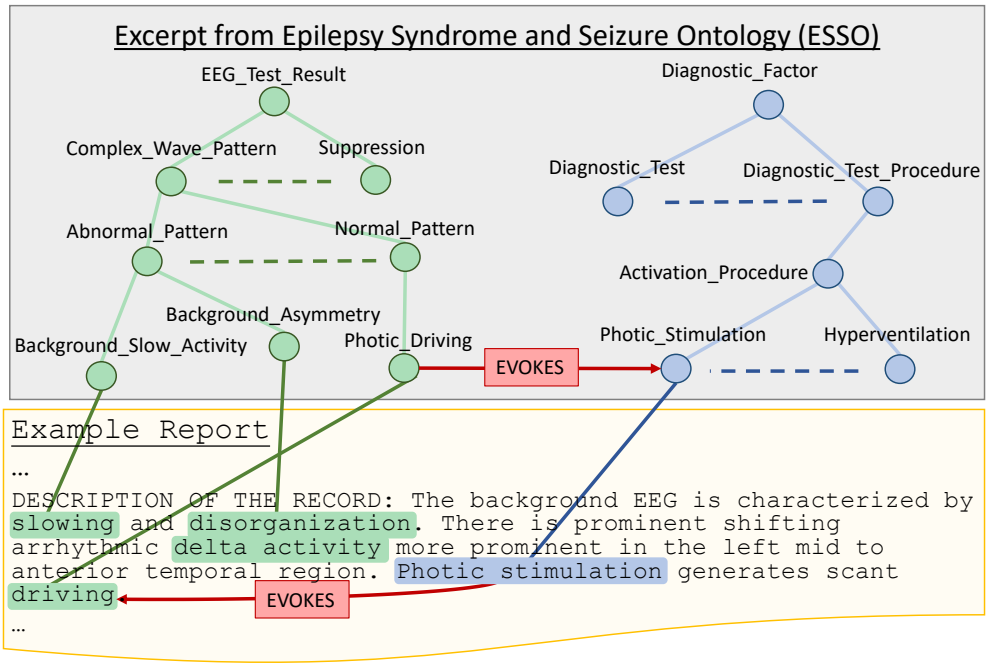

Figure 5.1. Concept mentions from clinical text linked to the ESSO ontology. A new relation (EVOKES) is inferred from the text.

with an upper ontology targeting epilepsy, a disease which affects an estimated 2.2 million people in the United States. However, ESSO is not a complete ontology, and its continuous creation could benefit from data-driven knowledge suggestion, especially when the data originates in clinical documents produced in the practice of neurology in patients exhibiting epilepsy symptoms. For this purpose, a large set of EEG reports can inform the generation of knowledge embeddings specific for epilepsy. Moreover, knowledge embeddings are used to infer new relations evidenced by the clinical narratives for possible inclusion in the ontology.

Figure 5.1 illustrates the application of the framework presented in this section. The depicted example report contains several mentions of medical concepts including EEG activities (highlighted in green) and EEG events (highlighted in blue), defined in Chapter 2. Each of these EEG activities and events is linked to a specific concept in ESSO, providing instances of the concepts represented in the ontology. While EEG activities and EEG events are linked to ESSO, medical problems and treatments extracted from the corpus are linked

to the UMLS (Lindberg et al., 1993) since ESSO does not contain a comprehensive set of medical problems and treatments. Figure 5.1 also contains an example of an EVOKES relation (defined in Chapter 3) that is inferred from the text and suggested for addition to the ontology. The relation triple ⟨*photic stimulation*, EVOKES, *photic driving*⟩ is identified in the example report illustrated in Figure 5.1.

The set of relations between medical concepts extracted from the TUH EEG corpus constitutes a data-driven knowledge graph, where several relations are considered. Using knowledge graph embedding methods, the most likely relations expressed in the data can be identified for inclusion in the ontology using probabilistic inference. Unlike concepts and relations encoded in the BioPortal ontologies, the knowledge graph embeddings associate relations between medical concepts with a probability or likelihood, enabling a probabilistic representation of biomedical knowledge. For example in Figure 5.1, the relation triple identified in the example report is deemed to be plausible and is therefore a candidate for inclusion as a new relation to the ontology.

To the best of our knowledge, this is the first report of the development of an embedded medical knowledge graph using free text clinical records. We learned knowledge graph embeddings representing 1,195,927 instances of binary relations between epilepsy-related concepts. These relations involved 2,442 instances of medical concepts.

5.2.1 Extraction of the Medical Knowledge Graph and Generation of Knowledge Embeddings

Extraction of the medical knowledge graph relies on the automatic identification of concepts and relations from data to enable (i) the population of the knowledge representation and (ii) linking the acquired knowledge to existing ontologies. In learning medical knowledge embeddings (MKE) from EEG reports we do not only perform bottom-up acquisition of medical knowledge from EEG reports, but we also represent the knowledge probabilistically

in a multi-dimensional space allowing inference to be performed on it. To do so, we followed a methodology which involves the following three steps:

STEP 1: Decide which medical concepts and which relations between them are expressed in the EEG reports;

STEP 2: Automatically generate the Knowledge Graph by extracting medical concepts and relations from the EEG reports;

STEP 3: Learn Medical Knowledge Embeddings (MKE) from the associated Knowledge Graph;

It is to be noted that the the MKE represent only knowledge available from the EEG reports, which do not discuss the taxonomic organization of medical concepts or their partonymy relations. These forms of relations are encoded in medical ontologies, thus the MKE provide complementary knowledge to medical ontologies. However, many of the concepts represented in the MKE are also encoded in existing medical ontologies, providing a simple mechanism of linking the MKE to various ontologies available in BioPortal. For example, the clinical history and the medication list of EEG reports mention multiple medical concepts already encoded in the Unified Medical Language System (UMLS) (Lindberg et al., 1993) ontology:

> Example 1: *CLINICAL HISTORY: This is a 20-year-old female with history of seizures described as generalized tonic-clonic with loss of consciousness for a few minutes. Last seizures occurred 2 years ago.*
>
> *MEDICATIONS: Keppra and Lamictal.*

Medical problems such as "*loss of consciousness*", and treatments such as *"Keppra"*, *"Lamictal"* are encoded in UMLS while concepts such as *seizures* will be linked both to UMLS and the ESSO ontology. However, these ontologies do not capture relations between such concepts that are implied in the EEG reports, e.g., which brain activities evidence some epilepsy-specific medical problems. Our three-step methodology aims to capture and represent such relationships such that their probabilistic likelihood can be expressed via inference.

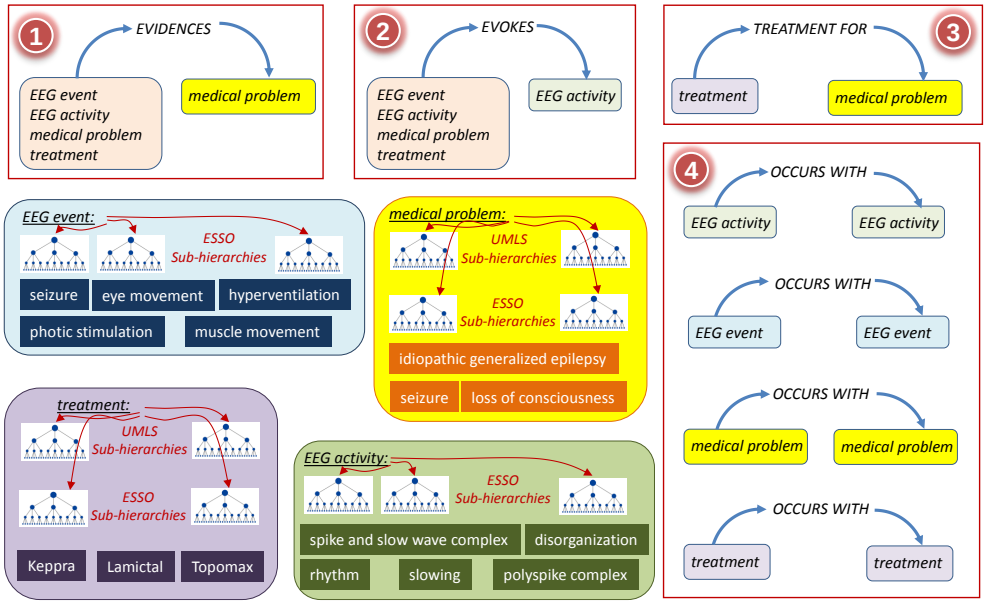

Figure 5.2. Medical concepts and relations considered for Medical Knowledge Embeddings (MKE), and their linkage to biomedical ontologies.

Knowledge graph embedding enables such probabilistic inference by providing a mechanism for calculating the likelihood of any possible relation between concepts expressed in the data. Therefore, in addition to the three-step methodology for generation and embedding of a knowledge graph for epilepsy, we present the application of the knowledge embeddings to probabilistic inference.

STEP 1: Decide which medical concepts and relations between them are expressed in EEG reports

In addition to medical problems and treatments that describe the clinical picture and therapy of a patient, EEG reports mention EEG events and EEG activities (defined in Chapter 2). Thus, we decided to encode in the MKE four types of medical concepts: (1) EEG events; (2) EEG activities; (3) medical problems and (4) treatments. Whenever these concepts are also encoded in other ontologies, we linked to them. EEG events and EEG activities are linked

against ESSO using a manual mapping between the event and activity schemata introduced in Chapter 2 and ESSO. Medical problems and treatments are linked against the UMLS using MetaMap Lite (Demner-Fushman et al., 2017). Medical problems and treatments that are contained in ESSO are also linked against ESSO using a manual mapping between the UMLS and ESSO. For example, medical problems such as *idiopathic generalized epilepsy*, when identified in an EEG report with methods developed in the STEP 2 of our methodology, shall be linked to UMLS through its concept unique identifier (CUI) discovered by MetaMap.

In addition to these four types of concepts, we decided to discern four types of binary relations that are implicit in the EEG reports. Each of these relations operates between a *source argument* and a *destination argument*. The relations along with examples of the four types of medical concepts are illustrated in Figure 5.2. The four binary relation types that we considered were motivated by discussions with several practicing neurologists and surgeons, corresponding to the implicit knowledge they discern from EEG reports. As shown in Figure 5.2, the EVIDENCES binary relation always has a medical problem as its destination concept, which is always mentioned in the clinical correlation section of the EEG report. The following example shows how the medical problem *idiopathic generalized epilepsy*, is evidenced by findings such as *polyspike discharges*, which is a mention of an EEG activity, in the impression section:

> Example 3: *IMPRESSION: This is an abnormal EEG recording capturing wakefulness through stage II sleep due to generalized spike and wave and polyspike discharges seen during wakefulness.*
>
> *CLINICAL CORRELATION: The above findings are consistent with idiopathic generalized epilepsy.*

As shown in Figure 5.2, the EVIDENCES relation considers EEG events, EEG activities, treatments, and medical problems as providing *evidence* for the medical problem from the clinical correlation section of the EEG report. The EVOKES binary relation always has an

Table 5.1. Examples of the Relations and Concepts expressed in EEG reports.

Evidences	**Evokes**
⟨*seizures*, EVIDENCES, *idiopathic generalized epilepsy*⟩	⟨*photic stimulation*, EVOKES, *photic driving response*⟩
⟨*polyspike discharges*, EVIDENCES, *idiopathic generalized epilepsy*⟩	⟨*hyperventilation*, EVOKES, *slowing*⟩
⟨*facial grimacing*, EVIDENCES, *psychogenic seizure*⟩	⟨*seizures*, EVOKES, *periodic lateralized epileptiform discharge*⟩
⟨*toxoplasmosis*, EVIDENCES, *degenerative brain disorder*⟩	⟨*shaking*, EVOKES, *rhythm*⟩

Treatment For	**Occurs With**
⟨*lamictal*, TREATMENT-FOR, *idiopathic generalized epilepsy*⟩	⟨*keppra*, OCCURS-WITH, *lamictal*⟩
⟨*depakote*, TREATMENT-FOR, *generalized anxiety disorder*⟩	⟨*encephalopathies*, OCCURS-WITH, *occipital lobe epilepsy*⟩
⟨*dilantin*, TREATMENT-FOR, *hematoma, subdural, chronic*⟩	⟨*cerebral dysgenesis*, OCCURS-WITH, *recurrent convulsions*⟩
⟨*ampicillin*, TREATMENT-FOR, *infection of foot*⟩	⟨*spike and slow wave complex*, OCCURS-WITH, *polyspike complex*⟩

EEG activity as a destination concept, as it attempts to capture the medical concepts that *evoke* the respective EEG activity. Those medical concepts can be either EEG events, or other EEG activities, medical problems or treatments followed by the patient. The third relation, namely OCCURS-WITH constrains both its arguments to be of the same type, e.g., either EEG activities, medical problems or treatments. The TREATMENT-FOR relation captures the treatments prescribed for certain medical problems. Table 1 illustrates examples of each of the four relations we considered, involving medical concepts illustrated in Figure 5.2, which lists all the EEG events and EEG activities that we decided to encode in the MKE, while providing several examples of medical problems and treatments, along with their UMLS CUIs. We used the vocabularies of EEG Activities and EEG Events from Chapter 2 based on the International Federation of Clinical Neurophysiology's glossary of terms (Noachtar et al., 1999).

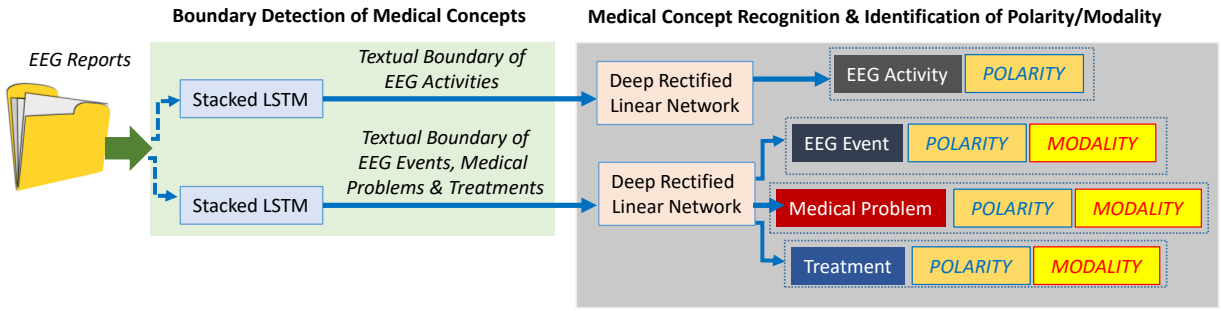

Figure 5.3. Deep Learning Architectures used for Recognizing Qualified Medical Concepts from EEG Reports.

STEP 2: Automatically generate the Knowledge Graph by extracting medical concepts and relations from the EEG reports

The extraction of medical knowledge from EEG reports consists of (1) automatic identification of medical concepts and (2) binary relation detection. Medical concept identification aims to recognize all the four types of concepts mentioned in EEG reports, along with their inferred *polarity* and *modality*, as described in Chapter 2. Through the identification of modality and polarity of the clinical concepts, we aimed to capture the neurologist's beliefs about the clinical concepts mentioned in the EEG report. Thus our medical concept identification method needed also to qualify the concepts by their polarity and modality.

Medical Concept Identification was performed by taking advantage of our existing active deep learning methodology described in Sections 2.4–2.5, which is illustrated in Figure 5.3. As described in Sections 2.4–2.5, the methodology first uses two stacked Long Short-Term Memory (LSTM) networks for detecting medical concepts in the text of the EEG report then uses two Deep Rectified Linear Networks (DRLNs) to (a) identify the medical concepts and (b) discern their polarity and modality. EEG activities are identified only with their polarity, as their mentions are always assumed to be factual, as illustrated in Figure 5.3. Moreover, medical problems and treatments are both normalized into UMLS concepts using MetaMap Lite (Demner-Fushman et al., 2017).

Detecting Relations between Medical Concepts was possible when pairs of medical concepts identified in the same EEG report were considered. Specifically, we established the four types of relations illustrated in Figure 5.2 by considering: (1) a potential EVIDENCES relation between any medical concepts from an EEG report and a medical problem identified in its clinical correlation section; (2) a potential EVOKES relation between any medical concept and an EEG activity, provided that the treatments were not identified in the clinical correlation section, as they may indicate possible or recommended treatments; (3) a potential TREATMENT-FOR relation between any treatment and a medical problem identified in the history section of the EEG report; and (4) a potential OCCURS-WITH relation between pairs of EEG activities, medical problems and treatments that are identified in the same section of the EEG report. We discard potential relations involving medical concepts with "negative" polarities and "possible" modalities since these medical concepts, while mentioned, were not actually observed. All these potential relations are indicative of implied relations, that are not always directly stated in the text of the EEG report.

Taken together, the set of medical concepts extracted from the entire TUH EEG corpus along with the collection potential relations between them constitute a *Knowledge Graph*, $G = \{V, E\}$ where V is the set of graph vertices and E is the set of graph edges. In our knowledge graph, V is the set of medical concepts and E is the set of relations between them. The medical knowledge embeddings learned in the following step can be used to provide the likelihood of any possible example of one of these relations.

STEP 3: Learning medical knowledge embeddings (MKE) from the knowledge graph

Learning MKE is made possible by relying on the TransE (Bordes et al., 2013) method, widely used (Guo et al., 2015; Lin et al., 2015; Wang et al., 2014) for representing multi-relational data corresponding to concepts and relations by modeling concepts as points in a

continuous vector space, $\mathbb{R}^d$, called the *embedding space*, where d is a parameter indicating the dimensionality of the embedding space. In our use of the TransE framework, relations between medical concepts are represented as *translation vectors*, also in $\mathbb{R}^d$, that connect the two points representing the two medical concepts in the embedding space. TransE learns an embedding, $\vec{\boldsymbol{c_i}}$, for each concept c_i and an embedding, $\vec{\boldsymbol{r}}$, for each relation type r such that the relation embedding is a *translation vector* between the two concept embeddings representing its arguments. This means that for any medical concept c_i, the concept most likely to be related to c_i by the relation r should be the medical concept whose embedding is closest to $(\vec{\boldsymbol{c_i}} + \vec{\boldsymbol{r}})$ in the embedding space. By modeling the medical concepts as points in the embedded space and the relations between them as translation vectors, we can measure the *plausibility* of any potential relation between any pair of concepts using the geometric structure of the embedding space. The plausibility of a relation between a source medical concept and a destination medical concept, represented as a triple, $\langle c_s, r, c_d \rangle$, is inversely proportional to the *distance* in the embedding space between the point predicted by our model $(\vec{\boldsymbol{c_s}} + \vec{\boldsymbol{r}})$ and the point in the embedding space representing the destination argument of the relation, i.e., $(\vec{\boldsymbol{c_d}})$. In this work, we use Manhattan Distance as our distance function:

$$f(c_s, r, c_d) = ||\vec{\boldsymbol{c_s}} + \vec{\boldsymbol{r}} - \vec{\boldsymbol{c_d}}||_{L1} \tag{5.1}$$

where $|| \cdot ||_{L1}$ is the $L1$ norm. Using this distance function, *plausible* triples have low value of f (since $\vec{\boldsymbol{c_s}} + \vec{\boldsymbol{r}} \approx \vec{\boldsymbol{c_d}}$ for plausible triples) and *implausible* triples have a high value of f.

Neural Network Architecture for learning MKE. To learn the concept embeddings and translation vectors, we use a neural network that will in fact produce the MKE. Formally, let C be the set of medical concepts found in the EEG reports and L be the set of relation types. Let $X = \{x^1 = \langle c_s^1, r^1, c_d^1 \rangle, \ \ldots, \ x^m = \langle c_s^m, r^m, c_d^m \rangle\}$ be the set of m relation triples extracted from the corpus of EEG reports at Step 2; where each $c_s^i, c_d^i \in C$ is a medical concept and each $r^i \in L$ is a relation type. The embedding, $\vec{\boldsymbol{c_j}}^{\,\boldsymbol{i}}$, for a concept c_j^i is calculated by first

generating a *one-hot* vector representation of c_j^i given by $v(c_j^i)$ which is a $|C|$-dimensional vector of zeros with a one in the dimension corresponding to the index of the concept c_j^i in the set of concepts C. The embedding $\vec{\boldsymbol{c_j}}^{\boldsymbol{i}} = v(c_j^i)\mathbf{E}$ is derived by multiplying the one-hot vector $v(c_j^i)$ with the *embedding matrix* $\mathbf{E} \in \mathbb{R}^{|C| \times N}$. Each row of $\mathbf{E}$ corresponds to a medical concept embedding and the operation $v(c_j^i)\mathbf{E}$ corresponds to selecting the k^{th} row of $\mathbf{E}$ if $v(c_j^i)_k = 1$. Likewise, the embedding for a relation type r^i is given by $\vec{\boldsymbol{r}}^{\boldsymbol{i}} = w(r^i)\mathbf{R}$ where $w(r^i)$ maps r^i to a one-hot vector of size $|L|$ and $\mathbf{R}$ is the relation embedding matrix. Consequently, Equation 5.1 can be computed using:

$$f(c_s^i, r^i, c_d^i) = ||v(c_s^i)\mathbf{E} + w(r^i)\mathbf{R} - v(c_d^i)\mathbf{E}||_{L1} \tag{5.2}$$

To learn *useful* embeddings we must also define a training objective that encodes *useful* relationships. Inspired by the work of Bordes et al. (2011), we use the following training objective: if either the source argument or destination argument from a training triple is removed, the model should be able to predict the correct medical concept. For example, the model should ensure that value of f(*keppra*,TREATMENT-FOR, *idiopathic generalized epilepsy*) is less than the value of f(*morphine*,TREATMENT-FOR, *idiopathic generalized epilepsy*) since keppra is a treatment for idiopathic generalized epilepsy, but morphine is not. Formally, we wish to learn the values of $\mathbf{E}$ and $\mathbf{R}$ such that for any training triple $x_i = \langle c_s^i, r^i, c_d^i \rangle$, the following two constraints are met:

$$f(c_s^i, r^i, c_d^i) < f(c_s^j, r^i, c_d^i), \forall j : \langle c_s^j, r^i, c_d^i \rangle \notin X \tag{5.3}$$

$$f(c_s^i, r^i, c_d^i) < f(c_s^i, r^i, c_d^j), \forall j : \langle c_s^i, r^i, c_d^j \rangle \notin X \tag{5.4}$$

To learn the optimal embedding matrices $\mathbf{E}$ and $\mathbf{R}$, we optimize the objective defined by the constraints outlined in Equations 5.3-5.4 by iterating the following process:

1. Randomly select a training triple $x^i = \langle c_s^i, r^i, c_d^i \rangle$ from X.

2. Create a *corrupted* version of the triple x_i^{neg} by selecting a medical concept c^{neg} at random from the set of medical concepts C and randomly replacing either c_s^i or c_d^i in x_i such that $x_i^{neg} \notin X$
3. Update $\mathbf{E}$ and $\mathbf{R}$ by backpropagating the ranking margin loss (Guo et al., 2015), $\max(0, \gamma + f(x_i) - f(x_i^{neg}))$, where γ is the *margin* parameter that determines how much of a margin should exist between triples in the training set and triples not in the training set.
4. Normalize each row e of E (i.e., $e := \frac{e}{||e||}$)

This process is repeated for each triple in X a fixed number of iterations (200,000 in this work). Our collection of 1,195,927 relation triples extracted from the TUH EEG corpus consisted of $|X| = 138,369$ unique relation triples. It is important to note that, as reported in (Bordes et al., 2013), the normalization in the fourth step prevents the model from trivially minimizing the loss by artificially increasing entity embedding norms.

Application of the MKE: Inference as Probabilistic Question Answering

The knowledge embeddings learned from the data-driven knowledge graph extracted from the TUH EEG corpus can be used to perform inference. Inference from a knowledge base can be viewed as answering questions using its encoded knowledge. Answering questions like (Q1) "what is the most likely treatment for idiopathic generalized epilepsy?", (Q2) "what EEG activity is most likely to occur with polyspike discharges?", and (Q3) "what is the likelihood that a patient with background slowing is diagnosed with cerebral dysfunction?" requires the ability to perform probabilistic inference. The MKE can be used to perform probabilistic inference by (1) representing the question as a relation triple q and (2) measuring the *plausibility* of q using equation 5.1 with the embedding matrices E and R automatically learned from the TUH EEG corpus. We estimated the probability of $q = \langle c_s^q, r^q, c_d^q \rangle$ in terms

of the geometric structure of the embedding space. Formally:

$$P(c_s^q, r^q, c_d^q) = 1 - \frac{f(c_s^q, r^q, c_d^q)}{\sum_{\langle c_s^i, r^i, c_d^i \rangle \in X} f(c_s^i, r^i, c_d^i)} \tag{5.5}$$

For example, answering (Q1) is the result of $\hat{c_s} = \arg\max_{c_s \in C} P(c_s,$ TREATMENT-FOR, *idiopathic generalized epilepsy*); answering (Q2) is the result of $\hat{c_d} = \arg\max_{c_d \in C} P($*polyspike discharges*, OCCURS-WITH, $c_d)$; and answering (Q3) is the result of $P($*background slowing*, EVOKES, *cerebral dysfunction*). The augmented relations from inference are detailed in the next subsection.

5.2.2 Experimental Results and Discussions

The medical knowledge embeddings for epilepsy were evaluated in terms of (a) their plausibility; and (b) their completeness. The plausibility of a knowledge graph embedding model measures how that model's plausibility function can be used to score valid relation triples above invalid relation triples. The completeness of a knowledge graph embedding model measures how well that model can be used to infer new information. The plausibility of relations encoded in MKE was assessed in three ways, measuring how well MKE *rank* triples from a test set T, of 1,000 relation triples held out from the data used to train the MKE. For each triple τ in the test set, we randomly remove either the source or destination argument and produce a set of *candidate* triples by replacing the removed argument with every medical concept $c \in C$. We rank the candidate triples in ascending order according to the distance function f. This allows us to calculate the following metrics using the rankings produced from every triple in the test set:

- **Mean Reciprocal Rank** (MRR) is a standard ranking evaluation that measures how high the first correct triple is ranked according to the model. $MRR = \frac{1}{|T|} \sum_{i=1}^{|T|} \frac{1}{rank_i}$ where $rank_i$ refers to the rank of the first correct triple in the ranking, where a *correct* triple is defined as any triple from any of the training, validation, or tests sets.

- **Precision at 10** (P@10) is another standard ranking evaluation that measures the percentage of the top K ranked triples are correct. As with MRR, correct triples are defined as any triple from any of the training, validation, or tests sets. The Precision at 10 evaluation shows how well the MKE ranks the triples about which the model is most confident.
- **Hits at K** (H@10, H@100) is a standard evaluation used for knowledge graph embeddings (Bordes et al., 2011, 2013) for evaluating link prediction. Hits at K measures how often the *specific* test triple t occurs in the K highest ranked triples, as opposed to precision which measures how often *any* correct triple occurs in the k highest ranked triples. We report both Hits at 10 and Hits at 100 to illustrate how often t is ranked among the most plausible triples, and how often t is ranked in the top 5% of triples.

The evaluation of completeness of the relations from the MKE also used the test set, T. We evaluated how well the MKE can *infer* new knowledge in the form of new relations from the test set. To measure how well the MKE can model relations of the held out triples from the test set, we consider each test triple, $\tau \in T$, and a *corrupted* version of the test triple, z, created by randomly replacing either the source argument or destination argument with a random medical concept and compute the *Pairwise Plausibility Accuracy* (PPA). The PPA measures the percentage of test triples for which the *plausibility*, $P(c_s^\tau, r^\tau, c_d^\tau)$, of the test triple τ is higher than plausibility, $P(c_s^z, r^z, c_d^z)$, of the corrupted triple. PPA demonstrates how well the MKE can differentiate between a correct triple, τ, and an incorrect triple, z, even if the model had never encountered τ during training. For these evaluations, the MKE were learned from 137,369 training triples automatically extracted from the TUH EEG corpus as described in the Methods section. We selected the dimension of the embedding space $d = 50$ from $[25, 50, 100, 200]$ and the margin parameter $\gamma = 1.0$ from $[0.1, 1.0, 5.0, 10.0]$ using grid search on a validation set of 500 relation triples.

Table 5.2 presents these results. The results for the Pairwise Plausibility Accuracy show that the MKE can correctly distinguish between relations that occur in the data (but that

Table 5.2. Quality of relations encoded in the MKE, measured using Pairwise Plausibility Accuracy (PPA), Mean Reciprocal Rank (MRR), Precision at 10 (P@10), Hits at 10 (H@10) and Hits at 100 (H@100).

Relation Type	**PPA**	**MRR**	**P@10**	**H@10**	**H@100**
EVIDENCES	86.04 %	96.44 %	77.22 %	63.37 %	84.43 %
EVOKES	94.22 %	96.10 %	84.62 %	84.91 %	97.17 %
OCCURS-WITH	90.00 %	62.30 %	45.58 %	27.77 %	68.36 %
TREATMENT-FOR	82.89 %	83.78 %	72.28 %	45.18 %	80.70 %
MICRO-AVERAGED	88.95 %	83.33 %	66.73 %	47.35 %	81.30 %

the model has not seen during training) and corrupted relations 88.95% of the time. The micro-averaged Mean Reciprocal Rank of 83.33% indicates that for the majority of triples in the test set, the top ranked candidate triple is correct. While the MRR of the OCCURS-WITH relation is the lowest (62.3%), it should be noted that, on average, there is at least one correct candidate triple ranked in the top two. The Precision at 10 metrics show that 66.73% of the top 10 ranked triples were correct, in general. It is interesting to note that the results for the Hits at 10 metric have the most variability between relation types. For the OCCURS-WITH relation, test triple, t, only occurs within the top 10 ranked triples 27.77% of the time. In contrast, for the EVOKES relation, t occurs within the top 10 ranked triples 84.91% of the time. In general, the Hits at 100 results show that the MKE correctly ranks t in the top 5% of candidate triples 81.3% of the time.

To analyze the correctness of medical knowledge distilled from EEG reports in the MKE, we manually inspected the 30 most plausible triples for each relation type. Specifically, for each triple, we determined whether that triple is consistent with established medical knowledge. In general, we found the EVOKES relation type to have the highest percentage of correct triples, highlighting the ability of the MKE to capture neurological experience from EEG reports. By contrast, the MKE successfully identified a number of unexpected OCCURS-WITH relations, including ⟨*hypothyroidism*, OCCURS-WITH, *turner syndrome*⟩, and ⟨*infantile spasms*, OCCURS-WITH, *MELAS Syndrome*⟩. Whereas the coincidence of hypothyroidism and Turner Syndrome is fairly well known, the relationship between infantile spasms

and MELAS syndrome is relatively obscure. Infantile Spasms, also known as West syndrome, is an exceedingly rare condition, with an estimated incidence in the United States of about 0.25-0.4 per 1000 live births (Paciorkowski et al., 2011). The MELAS syndrome is an even rarer inherited disorder of mitochondrial function which may be responsible for 8% of cases of infantile spasms (Sadleir et al., 2004). That the MKE recognized the connection between these two very rare conditions is quite interesting, and suggests that knowledge graph embedding holds promise for the elucidation of unusual concepts and relations from EEG reports in particular, and perhaps in medical reports more generally.

Owing to the data-driven nature of our technique, we generated some incorrect triples, as might be expected when using noisy free text data. For example, we observed two common types of errors when evaluating the EVIDENCES relation: (1) relation *inversion*, inverting the source and destination arguments of the relation; and (2) relation *confusion*, confusing one relation type with another. Consider the following example of a triple exhibiting relation inversion: (E1) ⟨*liver cirrhosis*, EVIDENCES, *encephalopathies*⟩. As defined in Figure 5.2, the source argument of the EVIDENCES relation is a medical concept suggesting or supporting the diagnosis listed in the destination argument. By contrast, it could be argued that, for triple (E1), the destination argument *encephalopathies* more commonly evidences the source argument *liver cirrhosis*. We believe these types of error could be addressed by incorporating semantic attributes (e.g., temporal information) to contextualize or constrain the arguments allowed for each relation type. Relation confusion is exemplified by the triple (E2) ⟨*rifaximin*, EVIDENCES, *brain diseases, metabolic*⟩. The source argument *rifaximin* is an antibiotic used in the management of the encephalopathy (i.e., the destination argument *brain diseases, metabolic*) related to severe liver failure. Thus, whereas there is a biologically plausible explanation for (E2), the EVIDENCES relation clearly does not accurately describe the relation; instead, the relation OCCURS-WITH may be preferred. This type of error could be mitigated in future work by introducing constraints into the knowledge embedding framework, as reported in Guo et al. (2015). Finally, there were rare cases in which the MKE assigned a high

plausibility to triples in which the source argument contradicts the destination argument, i.e., ⟨*insulin*, TREATMENT-FOR, *Diabetes Mellitus, Non-Insulin-Dependent*⟩. We believe that these types of error may be resolved by incorporating knowledge from existing ontologies to enforce consistency.

5.2.3 Lessons Learned

In this section, we presented the medical knowledge embeddings (MKE) automatically learned from clinical text in EEG reports. Unlike traditional ontologies which encode curated knowledge, the MKE infers probabilistic knowledge by extracting a large number of potential relation triples from clinical narratives. By applying knowledge graph embedding techniques, we were able to discover data-driven knowledge which can potentially be linked to ontologies from the BioPortal. Experimental results demonstrate the promise of this approach and highlight the potential of the MKE for bridging the knowledge gaps of existing neurological ontologies. The MKE presented in this section showcase the way in which deep learning techniques applied to large collections of medical records can supply medical knowledge derived from clinical practice to complement the knowledge already encoded in existing biomedical ontologies. By also considering the plausibility of medical knowledge, the MKE also enable probabilistic reasoning as a form of medical question answering.

5.3 Knowledge Embeddings for the Unified Medical Language System

In this section, we present a framework for extracting knowledge embeddings from the Unified Medical Language System (UMLS) (Lindberg et al., 1993). The UMLS integrates knowledge from nearly 200 source vocabularies into a single ontology consisting of over 3 million concepts linked together by over 3,000 relation types. The medical knowledge contained in the UMLS represents biomedical expertise that is relevant to myriad deep learning applications from predictive modeling to cohort retrieval.

In order to learn representations of the structured knowledge encoded in the UMLS that can be used in deep learning models, we present in this section a novel Generative Adversarial Network (GAN) (Goodfellow et al., 2014) method that leverages the unique properties of the UMLS. Both the learned UMLS knowledge embeddings and the knowledge embedding learning methodology are publicly available[2].

5.3.1 Knowledge Graphs in the Unified Medical Language System

The UMLS is comprised of two separate knowledge graphs: the Metathesaurus and the Semantic Network. The Metathesaurus is a traditional biomedical knowledge graph with relations between medical concepts, while the Semantic Network is comprised of relations between groups of medical concepts called *semantic types*. Each concept encoded in the UMLS Metathesaurus is linked to the corresponding concept names in various source vocabularies (e.g., ICD-10) and can be connected to other concepts via various relations, e.g., *"Is-A"*, *"Is-Part"* or *"Is caused by"*. Each concept from the Metathesaurus is assigned one or more *semantic types* (or categories), which are linked with one another through *semantic relationships*. The UMLS Semantic Network is a catalog of these semantic types (e.g., "anatomical structure" or "biological function") and semantic relationships between them (e.g., "spatially related to" or "functionally related to"). While there are over 3 million concepts in the UMLS Metathesaurus, there are only 180 semantic types and 49 semantic relationships in the UMLS Semantic Network. The UMLS Metathesaurus graph along with the UMLS Semantic Network graph encode a wealth of medical knowledge, capturing ontological and biomedical expertise which could also be used by deep learning methods, in addition to the concept embeddings derived from the EHRs. To enable the usage of the knowledge encoded in UMLS in deep learning methods, we need to learn *knowledge embeddings* which represent (1) the UMLS concepts; (2) the relations between UMLS concepts;

[2]https://github.com/r-mal/umls-embeddings

(3) the nodes of the UMLS Semantic Network, representing the semantic types assigned to concepts; and (4) the semantic relations shared between the nodes of the UMLS Semantic Network (i.e., the semantic types).

5.3.2 Adversarial Learning of Knowledge Embeddings for the UMLS

The structure of the UMLS knowledge encoding poses a challenge to the applicability of existing knowledge graph embedding models, which assume a single knowledge graph. The UMLS encodes two different and jointly connected graphs, namely (a) the UMLS Metathesaurus; and (b) the UMLS Semantic Network. In this section we present a neural model capable of learning UMLS knowledge embeddings representing concepts, relations between them, semantic types and semantic relations.

In Section 5.2, we relied on the TransE (Bordes et al., 2013) method for knowledge graph embedding which represents medical concepts and relations between them as real-valued vectors $\vec{c}, \vec{r} \in \mathbb{R}^d$. By modeling the concepts as points in the embedding space and the relations between them as translation vectors, it is possible to measure the *plausibility* of any potential relation between any pair of medical concepts using the geometric structure of the embedding space: $f(c_1, r, c_2) = ||\vec{c_1} + \vec{r} - \vec{c_2}||_{L1}$ where $||\cdot||_{L1}$ is the $L1$ norm. This scoring function can be used for (a) assigning a plausibility score to each triple $\tau = \langle c_1, r, c_2 \rangle$, encoding a relation between a pair of concepts from the UMLS Metatheraurus; as well as (b) assigning another plausibility score to each triple $\lambda = \langle t_1, sr, t_2 \rangle$ encoding semantic relationships (sr) between semantic types (t_1, t_2) encoded in the UMLS Semantic Network.

In addition to TransE, several other knowledge graph embedding models, which represent concepts and relations as vectors or matrices in an embedding space, have shown promise in recent years. We list in Table 5.3 two additional models that we have used when learning UMLS embeddings. TransD (Ji et al., 2015b) learns two embedding vectors for each concept in a knowledge graph: $[\vec{c}, \vec{c_p}]$ as well as two embeddings for each relation in the graph: $[\vec{r}, \vec{r_p}]$,

Table 5.3. Scoring functions used in models that learn knowledge embeddings. $\boldsymbol{I}$ is the identity matrix.

Model	**Scoring Function**
TransE	$\|\|\vec{c_1} + \vec{r} - \vec{c_d}\|\|_{L1}$
TransD	$\|\|(\boldsymbol{I} + \vec{r_p} \times \vec{c_{sp}}^{\top}) \times \vec{c_1} + \vec{r}$ $-(\boldsymbol{I} + \vec{r_p} \times \vec{c_{dp}}^{\top}) \times \vec{c_d}\|\|_{L1}$
DistMult	$\sum_i \vec{c_1}^i \cdot \vec{r}^i \cdot \vec{c_d}^i$

where the first vector represents the "knowledge meaning" of the concept or relation while the second vector is a *projection* vector (with subscript p), used to construct a dynamic mapping matrix for each concept/relation pair. If the knowledge meaning of a concept or relation refers to the reason why the concept or relation was encoded in the knowledge graph, the projection of concepts in the space of the relations is used to capture the interaction between concepts participating in relations and relations holding various concepts as arguments. Essentially, TransD constructs a dynamic mapping matrix for each entity-relation pair by considering the diversity of entities and relations simultaneously. As each source concept c_1 is translated into a pair $[\vec{c_1}, \vec{c_{sp}}]$ and each destination concept is translated into a pair $[\vec{c_d}, \vec{c_{dp}}]$, while the relation between them is translated into $[\vec{r}, \vec{r_p}]$, the plausibility of the relation is measured by the scoring function listed in Table 5.3. DistMult (Yang et al., 2015), another knowledge embedding model, is a simplification of the traditional bilinear form of matrix decomposition using only a diagonal matrix that has been shown to excel for probabilistic models. Its scoring function, listed in Table 5.3, is equivalent to the dot product between the vector representations of the source concept, the relation and the destination concept.

Training any of these embeddings models requires both positive examples encoded in the knowledge graph (in our case UMLS Metathesaurus and Semantic Network), and negative examples representing relations that do not occur in the knowledge graph. In Section 5.2, negative examples are obtained by removing either the correct source or destination concept and replacing it with a concept randomly sampled from a uniform distribution (as in Equations 5.3-5.4). As noted in Cai and Wang (2018), this approach of generating negative

examples is not ideal, because the sampled concept (or semantic type) may be completely unrelated to the original UMLS concept (or source UMLS semantic type), resulting in a learning framework using too many obviously false examples. To address this challenge, we have extended the KBGAN (Cai and Wang, 2018) adversarial learning framework, which is currently one of the state-of-the-art learning methods for knowledge embeddings.

Generative Adversarial Networks (GANs) are at the core of our framework for learning knowledge embeddings for the UMLS. GANs typically use a *generator* and a *discriminator*, as introduced by Goodfellow et al. (2014). Metaphorically, the generator can be thought of as acting like a team of counterfeiters, trying to produce fake currency and use it without detection. The discriminator can be thought of as acting like the police, trying to detect the counterfeit currency. Competition in this game enabled by the GAN drives both teams to improve their methods until the counterfeiters are indistinguishable from the genuine articles. In the KBGAN (Cai and Wang, 2018) framework, the discriminator learns to score the plausibility of a given relation triple and the generator tries to fool the discriminator by generating plausible, yet incorrect, triples. In order to accomplish this goal, the generator calculates a probability distribution over a set of negative examples of relation triples and then samples one triple from the distribution as its output. However, a single generator is not sufficient for creating UMLS embeddings, because the UMLS graph contains two types of relations, namely (1) relations between UMLS concepts and (2) semantic relations between UMLS semantic types. Therefore, we extended the KBGAN by using two different generators: an UMLS Metathesaurus generator G_1 and an UMLS Semantic Network generator G_2, as illustrated in Figure 5.4. Given any relation between two concepts encoded in the UMLS Metathesaurus, G_1 calculates the probability distribution over a set of candidate negative examples of the relation, samples it and produces a negative example. Given the ground truth relation triple $R_1 = \textbf{IsA}$*(Opioid Abuse, Drug Abuse)* from the Metathesaurus G_1 will produce the negative example $R_1^N = \textbf{IsA}$*(Opioid Abuse, UMLS concept$_i$)*, as illustrated in

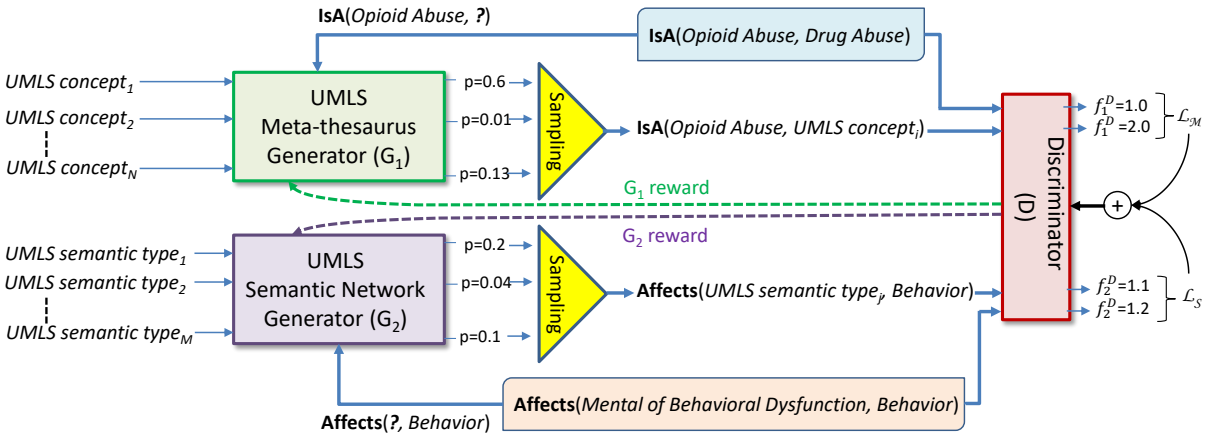

Figure 5.4. Adversarial Learning Framework for Producing Knowledge Embeddings for UMLS.

Figure 5.4, where *UMLS concept*$_i$ is some UMLS concept which is not in an **IsA** relation with *Opioid Abuse* in the UMLS Metathesaurus. Similarly, given a ground truth semantic relation triple $R_2 =$ **Affects***(Mental or Behavioral Dysfunction, Behavior)*, G_2 generates the negative example, $R_2^N =$ **Affects***(UMLS Semantic Type$_j$, Behavior)* where *UMLS Semantic Type$_j$* is not in an **Affects** relation with UMLS semantic type *Behavior*. Both negative examples generated by G_1 and G_2 are sent to the Discriminator D along with the two ground truth relation triples: R_1 and R_2, respectively. D uses the function f_1^D to compute the scores for R_1 and R_1^N while it uses the function f_2^D to compute the scores for R_2 and R_2^N. For both of f_1^D and f_2^D we experimented with two alternatives: (1) the scoring function of TransE, and (2) the scoring function of TransD, listed in Table 5.3.

Intuitively, the discriminator D should assign low scores produced by the functions f_1^D and f_2^D to high-quality negative samples generated by G_1, and G_2 respectively. Moreover, the discriminator D should assign *even lower* f_1^D and f_2^D scores to the ground truth triples than to the high-quality negative samples. Suppose that G_1 produces a distribution of negative triples $p_{G_1}(\tau'|\tau)$ for a positive example $\tau = \langle c_1, r, c_2 \rangle$ encoded in the UMLS Metathesaurus and generates $\tau' = \langle c_1', r, c_2' \rangle$ by sampling from this distribution. Similarly, suppose that

G_2 produces a distribution of negative triples $p_{G_2}(\lambda'|\lambda)$ for a positive example $\lambda = \langle t_1, sr, t_2 \rangle$ encoded in the UMLS Semantic Network. Let f_1^D and f_2^D be the two scoring functions of D. Then the objective of the discriminator is to minimize *the marginal loss* between the ground truth (or positive) triples and the negative example triples generated by G_1 and G_2. To jointly minimize the marginal loss of D, we extend the marginal loss function of KBGAN (Cai and Wang, 2018) to have two terms: (i) the Metathesaurus loss function, $\mathcal{L}_M$ and (ii) the Semantic Network loss function $\mathcal{L}_S$. We defined $\mathcal{L}_M$ as:

$$\mathcal{L}_M = \sum_{\tau \in \mathcal{M}} ||f_1^D(\tau) - f_1^D(\tau') + \gamma_1|| \tag{5.6}$$

where $\mathcal{M}$ represents all valid triples from the Metathesaurus, while $f_1^D(\tau)$ measures the plausibility of the triple $\tau = \langle c_1, r, c_2 \rangle$ and γ_1 is a margin hyper-parameter. We defined $\mathcal{L}_S$ as:

$$\mathcal{L}_S = \sum_{\lambda \in \mathcal{S}} ||f_2^D(\lambda) - f_2^G(\lambda') + \gamma_2|| + \sum_i \left[\vec{t_i} - \frac{1}{|\delta(t_i)|} \times \sum_{c \in \delta(t_i)} c \right] \tag{5.7}$$

where $\mathcal{S}$ represents all valid triples from the UMLS Semantic Network, while $f_2^D(\lambda)$ measures the plausibility of the triple $\lambda = \langle t_1, sr, t_2 \rangle$ from the UMLS Semantic Network expressing the semantic relation sr between the semantic types t_1 and t_2 in the UMLS Semantic Network and γ_2 is a margin hyper-parameter. The embedding of the UMLS semantic type t_i is denoted by $\vec{t_i}$ and $\delta(t_i)$ represents the set of UMLS concepts having the semantic type t_i. In this way, the centroid of the embeddings of UMLS concepts having the UMLS semantic type t_i is represented as $1/\delta(t_i) \times \sum_{c \in \delta(t_i)} c$. This allows us to measure in $\mathcal{L}_S$ not only the margin between the semantic relation produced by G_2 to the ground truth semantic relation encoded in the UMLS Semantic Network, but also the cumulative distance between the embeddings of each semantic type t_i and the centroid of the embeddings corresponding to the UMLS concepts sharing the semantic type t_i. Hence, $\mathcal{L}_S$ measures the loss of (a) not correctly recognizing a plausible semantic relation from the UMLS Semantic Network, but also (b) the loss of not recognizing plausible semantic types in the UMLS Semantic Network, given as

reference all the UMLS semantic concepts that share same semantic type. This ensures that we learn embeddings of semantic relations from UMLS by taking into account the semantic types and the concepts that are encoded in UMLS.

In the adversarial framework presented in Figure 5.4, the objective of generator G_1 is to maximize the following expectation:

$$\mathcal{R}_{G_1} = \sum_{\tau \in \mathcal{M}} \mathbb{E}[-f_1^D(\tau')] \quad \tau' = \langle {c_1}', r, {c_2}' \rangle \sim p_{G_1}(\tau'|\tau) \tag{5.8}$$

Similarly, the objective of generator G_2 is to maximize the following expectation:

$$\mathcal{R}_{G_2} = \sum_{\lambda \in \mathcal{S}} \mathbb{E}[-f_2^D(\lambda')] \quad \lambda' = \langle {t_1}', sr, {t_2}' \rangle \sim p_{G_2}(\lambda'|\lambda) \tag{5.9}$$

Both G_1 and G_2 involve a sampling step. To find the gradient of R_{G_1} and R_{G_2} we used a special case of the Policy Gradient Theorem (Sutton et al., 1999), which arises from reinforcement learning (RL). To optimize both R_{G_1} and R_{G_2}, we maximized the *reward* returned by the discriminator to each generator in response to selecting negative examples for the relations encoded in UMLS, providing an excellent framework for learning the UMLS embeddings that benefits from good negative examples in addition to the abundance of positive examples. Finally, both generators G_1 and G_2 need to have scoring functions, defined as $f_{G_1}(\tau)$ and $f_{G_2}(\lambda)$. Several scoring function can be used, selecting from those that have been implemented in several knowledge graph embeddings, listed in Table 5.3. In our implementation, we have used for both generators the same scoring function as the one used in DistMult. Then given a set of candidate negative examples for the UMLS Metathesaurus: $Neg^{\mathcal{M}}(\tau) = \{\langle {c_1}', r, c_2 \rangle | {c_1}' \in \mathcal{C}\} \cup \{\langle c_1, r, {c_2}' \rangle | {c_2}' \in \mathcal{C}\}$ (where $\mathcal{C}$ represents all the concepts encoded in the UMLS Metathesaurus), the probability distribution p_{G_1} is:

$$p_{G_1}(\tau'|\tau) = \frac{exp(f_{G_1}(\tau'))}{\sum_{\tau^* \in Neg^{\mathcal{M}}} exp(f_{G_1}(\tau^*))} \tag{5.10}$$

Similarly, given a set of candidate negative examples for the UMLS Semantic Network $Neg^{\mathcal{S}}(\lambda) = \{\langle {t_1}', sr, t_2 \rangle |\ {t_1}' \in \mathcal{T}\ \} \cup \{\ \langle t_1, sr, {t_2}' \rangle | {t_2}' \in \mathcal{T}\}$ (where $\mathcal{T}$ represents all the

semantic types encoded in the UMLS Semantic Network), then the probability distribution p_{G_2} is modeled as:

$$p_{G_2}(\lambda'|\lambda) = \frac{exp(f_{G_2}(\lambda'))}{\sum_{\lambda^*\rangle \in Neg^S} exp(f_{G_2}(\lambda^*))} \tag{5.11}$$

In this adversarial training setting, the generators G_1 and G_2 and the discriminator D are alternatively trained towards their respective objectives, informing the two forms of embeddings for the UMLS: knowledge embeddings for the UMLS Metathesaurus and knowledge embeddings for the UMLS Semantic Network.

5.3.3 Experimental Results and Discussions

The quality of the UMLS knowledge embeddings was evaluated in terms of plausibility and completeness, as in Section 5.2.2 using PPA, Hits@10, and MRR. We also introduce a new evaluation in this section: Relation Triple Classification (RTC) – an evaluation that uses the model to classify candidate triples as valid or invalid.

To perform RTC, we defined two *plausibility functions* informed by the scoring functions used in the Discriminator: the first operating on the UMLS Metathesaurus, defined as $\rho_1 = -f_1^D$ and the second operating on the UMLS Semantic Network, defined as $\rho_2 = -f_2^D$. Therefore, we measured how well ρ_1 can be used to predict a correct relation $\tau = \langle c_1, r, c_2 \rangle$ encoded in the UMLS Metathesaurus, e.g., answering the questions "Is OPIOID ABUSE **a kind of** DRUG ABUSE?", or how well ρ_2 can be used to predict a semantic relation $\lambda = \langle t_1, sr, t_2 \rangle$ encoded in the UMLS Semantic Network, e.g., answering the question "Can MENTAL OR BEHAVIORAL DYSFUNCTION **affect** BEHAVIOR?". Given a relation triple τ, we use $\rho_1(\tau)$ to classify τ as either a valid or invalid relation triple depending on if $\rho_1(\tau) > \omega_1$ for some threshold ω_1. Likewise, we discover a threshold value ω_2 for the Semantic Network using the scoring function ρ_2. To perform RTC, we split the relation triples from each knowledge graph into train/validation/test splits. For each triple in each validation and test set, we create an invalid triple by replacing either the source or destination concept

and add the invalid triple to the respective set. We then use the trained model to score each triple from the validation set and select the optimal values of ω_1 and ω_2 for each model. We evaluate RTC on the test sets using Precision (RTC-P) and Recall (RTC-R). RTC-P can be thought of as measuring plausibility while RTC-R measures completeness.

We experimented with several methods for learning knowledge embeddings, and list their results in Table 5.4. It can be noted that the plausibility and completeness obtained by the GAN-based model, presented in this paper, consistently obtained the best results. We evaluated the performance of six knowledge embedding models by varying (a) the scoring functions (TransE and TransD), (b) information from the UMLS Semantic Network, and (c) the generative (GAN) adversarial learning framework. The TransE and TransD models were trained using only the Metathesaurus loss function, $\mathcal{L}_M$ shown in equation 5.6. The TransE_SN and TransD_SN models incorporated the Semantic Network by using both $\mathcal{L}_M$ and the Semantic Network loss function, $\mathcal{L}_S$ shown in equation 5.7. The GAN (TransE_SN+DistMult) and GAN (TransD_SN+DistMult) are trained with the full adversarial framework described in the Methods section, using the TransE and TransD scoring functions in their discriminators, respectively. Both GAN models use the DistMult scoring function for both their Metathesaurus (G_1) and Semantic Network (G_2) generators.

Each model was trained for 13 epochs using 9,169,311 Metathesaurus triples between 1,726,364 concepts spanning 388 relation types. The results for the Metathesaurus evaluations were obtained using the entire Semantic Network (6217 triples between 180 semantic types spanning 49 relation types). The dimension of the embedding space $d = 50$ was selected from $[25, 50, 100, 200]$ and the margin parameters $\gamma_1, \gamma_2 = 0.1$ from $[0.1, 1.0, 5.0]$ using grid search on the validation set. All models are optimized using Adam (Kingma and Ba, 2015) with default parameters. TransE and TransD models are learned with the usual constraint that the L_2-norm of each embedding is ≤ 1 and the DistMult models use L_2 regularization. Table 5.4 shows that the GAN-based models outperform the non-adversarially

Table 5.4. Plausibility and completeness of the UMLS knowledge embeddings. _SN indicates the incorporation of the Semantic Network in the embeddings, which otherwise were learned only from the Metathesaurus.

	UMLS Metathesaurus				
Model	RTC-P	RTC-R	PPA	H@10	MRR
TransE	0.7712	0.6479	0.9340	0.2161	0.1400
TransD	0.9080	0.8895	0.9734	0.2780	0.1674
TransE_SN	0.8649	0.8019	0.9746	0.2240	0.1425
TransD_SN	0.9188	0.8915	0.9729	0.2775	0.1670
GAN (TransE_SN+DistMult)	0.8959	0.8424	**0.9833**	0.2727	0.1650
GAN (TransD_SN+DistMult)	**0.9311**	**0.9130**	0.9803	**0.3164**	**0.1886**
	UMLS Semantic Network				
TransE_SN	0.5105	0.7790	0.9150	0.7125	0.4882
TransD_SN	0.6840	0.8771	0.9017	0.7300	0.4680
GAN (TransE_SN+DistMult)	0.6109	0.7898	**0.9367**	**0.7883**	**0.5373**
GAN (TransD_SN+DistMult)	**0.8419**	**0.8546**	0.9200	0.7867	0.5236

learned models in each evaluation for the Metathesaurus and Semantic Network, demonstrating their effectiveness.

The results listed in Table 5.4 indicate that the TransD models outperform the TransE models on the Metathesaurus evaluations (by 16% on H@10 and 14.3% on MRR), however TransE outperforms TransD on the Semantic Network evaluations, albeit by a lesser margin (1.8% on H@10 and 2.5% on MRR). Clearly, the full GAN model using TransD outperforms the other model configurations, attaining the top performance in RTC-P, RTC-R, H@10 and MRR for the Metathesaurus. This work demonstrates that adversarial learning of UMLS knowledge embeddings is an effective strategy for learning embeddings representing medical concepts, relations between them, semantic types and semantic relations.

The learned knowledge embeddings exhibit interesting properties. For example, the 5 nearest neighbors of the UMLS concept '*Malignant neoplasm of the lung*' (C0242379) are all different kinds of malignant neoplasm, including neoplasms of the skin (C0007114), brain (C0006118), pancreas (C0017689), bone (C0279530), and trachea (C0153489), each having the semantic type *Neoplastic Process* (T191). Likewise, the 10 nearest neighbors of the

medical concept '*Heroin Dependence*' are all *Mental or Behavioral Dysfunction*s (T048) indicating different drug abuse/dependency problems and the 10 nearest neighbors of '*Quantitative Morphine Measurement*' (C0202428) are all *Laboratory Procedure*s (T059) testing for some kind of opioid. The model does, however, struggle with concepts that have low connectivity in the knowledge graph (e.g., the 10 nearest neighbors of the concept '*Stearic monoethanolamide*', which only appeared in 3 relation triples, are largely unrelated due to the low degree of that concept in the Metathesaurus graph).

5.3.4 Lessons Learned

In order to produce UMLS embeddings, a novel generative adversarial network (GAN) is presented which took into account the two knowledge graphs of the UMLS: the Metathesaurus and the Semantic Network. This necessitated the use of two generators in the GAN framework, one for each of the knowledge graphs from the UMLS, and a single discriminator able to encode knowledge from both graphs jointly. The experimental results suggest that the proposed method improves knowledge representation quality indicating that the multi-generator framework is able to leverage knowledge from each graph to improve the representation of the other. The knowledge embeddings, which have been made publicly available, can be used in a multitude of deep learning methods to benefit from the knowledge encoded in UMLS.

5.4 Application of the UMLS Knowledge Embeddings in a Clinical Prediction Model

To showcase the impact of using UMLS knowledge embeddings, we have considered the task of building a deep learning model for discovering (1) the incidence of opioid use disorders (OUD) after onset of opioid therapy and (2) chronic opioid therapy (COT) achievement and persistence. OUD is defined as a problematic pattern of opioid use that causes significant

impairments or distress. COT is defined as 45+ days supply of opioid analgesics in a calendar quarter (3 months) for at least one quarter within the 7-year time range. As such COT *achievement* occurs when the conditions for COT are first noted while COT *persistence* is observed when a patient continues to be prescribed a 45+ days supply of opioid analgesics in consecutive quarters. OUD is part of the current opioid use epidemic, which is among the most pressing public health issues in the United States as opioid related poisonings and deaths have increased at alarming rates since 2014. Long-term opioid therapy poses a much higher risk of OUD and other adverse outcomes. In 2014, US retail pharmacies dispensed 245 million prescriptions for opioid pain relievers. Of these prescriptions 65% were for short term therapy ($<$ 3 weeks). However, 3-4% of the adult population (9.6-11.5 million patients) were prescribed longer term ($>$ 90 days) opioid therapy.

In this section, we present a predictive model for discovering the incidence of OUD after onset of opioid therapy and COT achievement and persistence uses a deep learning architecture based on hierarchical attention. Superior prediction are obtained when the model is informed by the UMLS knowledge embeddings generated with the methodology presented in Section 5.3.

5.4.1 Hierarchical Attention Networks with UMLS Embeddings

A deep learning method using a hierarchical attention mechanism was used to predict (1) the incidence of Opioid Use Disorders (OUD) after onset of opioid therapy and (2) Chronic Opioid Therapy (COT) achievement and persistence, The hierarchical attention network, illustrated in Figure 5.5, relies on embeddings representing ICD-10 codes, medications ordered and laboratory results. To produce these embeddings, we considered that if a patient had records spanning N quarters, each having at most M different diagnostic codes assigned during the quarter, we could denote each diagnostic code as d_{it}, to represent the t-th ICD-10 code in the i-th quarter, with $i \in [1, N]$. To encode each diagnostic code d_{it} as a low-dimensional vector c_{it} (also called ICD-10 code embedding) we compute: $c_{it} = \mathbb{Q} \times d_{it}$, with

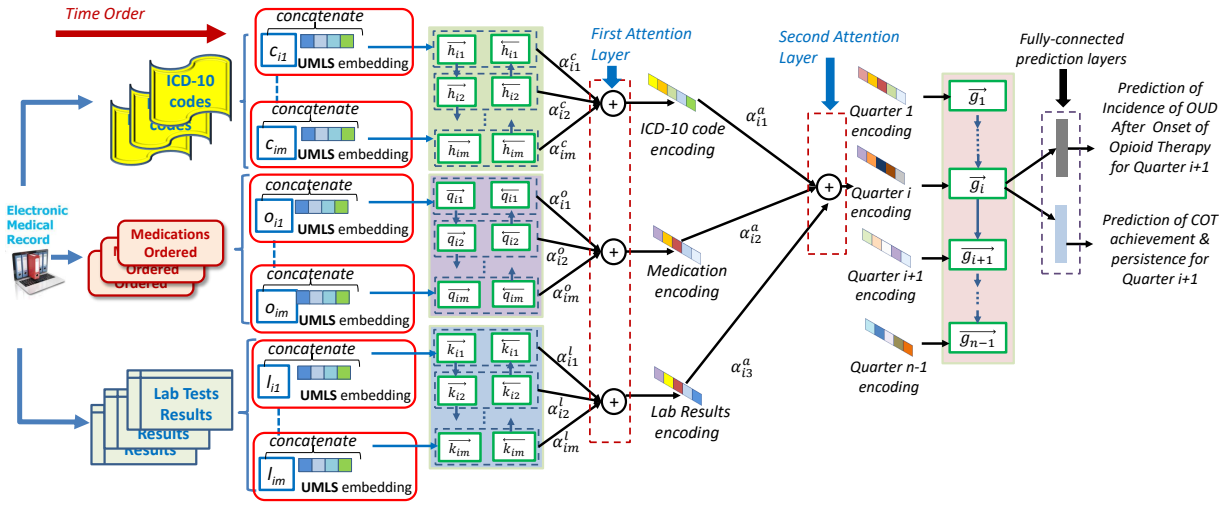

Figure 5.5. Architecture of a Hierarchical Attention-Based Prediction Model incorporating the UMLS embeddings.

$t \in [1, M]$, in which $\mathbb{Q}$ represents the embedding matrix obtained from WORD2VEC(Mikolov et al., 2013). As shown in Figure 5.5, each ICD-10 code embedding was concatenated with an UMLS knowledge embedding, such that the knowledge from clinical practice (ICD-10 code WORD2VEC embeddings) can be combined with complementary knowledge, available from the UMLS ontology. The concatenation of UMLS knowledge embeddings was informed by the mapping of the ICD-10 vocabulary into UMLS concepts provided by the UMLS. We were then able to encode this combined knowledge representation pertaining to diagnoses and their ICD-10 codes as well as UMLS concepts representing them using a Recurrent Neural Network (RNN), implemented with bi-directional gated-recurrent units (GRUs). More specifically, for each concatenated embedding cc_{it} , we computed two vectors: (1) $\overrightarrow{h_{it}} = \overrightarrow{GRU}(cc_{it})$) for $t \in [1, M]$; and (2) $\overleftarrow{h_{it}} = \overleftarrow{GRU}(cc_{it})$, for $t \in [M, 1]$; generating the encoding $x_{it} = [\overrightarrow{h_{it}}, \overleftarrow{h_{it}}]$. Similarly, we computed the encodings of medications ordered o_{it} and of laboratory results l_{it} using the same type of RNNs as we did for ICD-10 codes. These encodings were also concatenated with corresponding UMLS embeddings. The medications and laboratory results are mapped to UMLS concepts using MetaMap Lite (Demner-Fushman

et al., 2017). For the concatenated encodings of medications ordered o_{it} with UMLS embeddings, denoted oo_{it}, embeddings for medications were produced using again bi-directional GRUs as: $y_{it} = [\overrightarrow{q_{it}}, \overleftarrow{q_{it}}]$, where $\overrightarrow{q_{it}} = \overrightarrow{GRU}(oo_{it})$ and $\overleftarrow{h_{it}} = \overleftarrow{GRU}(oo_{it})$. When concatenating the encodings of laboratory results l_{it} with UMLS embeddings, denoted ll_{it}, the bi-directional GRUs generated embeddings for laboratory results: $z_{it} = [\overrightarrow{k_{it}}, \overleftarrow{k_{it}}]$, where $\overrightarrow{k_{it}} = \overrightarrow{GRU}(ll_{it})$ and $\overleftarrow{k_{it}} = \overleftarrow{GRU}(ll_{it})$. In addition, since not all ICD-10 codes, medications or laboratory results contribute equally to the clinical picture of the patient, we introduce an attention mechanism, that enables the predictive model to pay more *attention* to the more informative ICD-10 codes, medications and laboratory test results. Attention mechanisms are a new trend in deep learning, loosely based on visual attention mechanisms in humans, that have been successfully used in caption generation (Xu et al., 2015) and medical predictions (Choi et al., 2016; Sha and Wang, 2017). In our predictive model, illustrated in Figure 5.5, we used a form of *hierarchical attention mechanism*, inspired by the work of Yang et al. (2016). The first layer of attention learned how each of the combinations of ICD-10 codes and corresponding UMLS embeddings, ordered medications and corresponding UMLS embeddings and laboratory test results and corresponding UMLS embeddings contribute to the predictions and how to pay more attention to the more impactful ones. In the case of ICD-10 code embedding, attention is learned through the following equations: $u_{it} = tanh(W_c \times x_{it} + b_c)$; $\alpha^c_{it} = \exp(u^\intercal_{it} \times cc_{it}) / \sum_t \exp(u^\intercal_{it} \times cc_{it})$; $ICD^{encod}_{10} = \sum_t \alpha^c_{it} \times x_{it}$. As illustrated in Figure 5.5, similar attention mechanisms are implemented in the first attention layer for the medication and for the laboratory results encodings, with the attention parameters α^o_{it} and α^l_{it} respectively.

A second layer of attention was also implemented, since we wanted the prediction model to also learn which form of clinical information combined with UMLS knowledge was the most impactful in deciding for the following quarter the COT achievement/persistence and the OUD incidence. Therefore, we learned an encoding for each quarter from the clinical

picture and therapy of the patients available from ICD-10 codes, medications ordered and laboratory results of all hospital visits in a given quarter. The attention mechanism of the second layer uses parameters: α^a_{i1} (for ICD-10 codes encodings), α^a_{i2} (for medications ordered encodings) and α^a_{i3} (for laboratory result encodings). The results of the second layer of the hierarchical attention mechanism feed into a GRU which feeds into a fully prediction layer, as illustrated in Figure 5.5, allowing one binary classifier to decide whether COT will be achieved or persist in the next quarter, whereas a second binary classifier decides whether the incidence of OUDs will be observed.

5.4.2 Data Used by the Prediction Model

In this section, a large clinical dataset from the University of Washington Medical Center and Harborview Medical center is used for predicting the incidence of OUD after onset of opioid therapy and COT achievement and persistence. Adult patients (age $\geq$ 18 years), were considered eligible in this study if they were prescribed with COT for chronic non-cancer pain between 2011-2017 (7 years). A cohort of 6355 patients receiving COT with a total of 23,945 COT quarters (avg:3.77, min:1, max:27) was created using the described inclusion/exclusion criteria. There were 3446 patients (54%) with 1 COT quarter, 1856 patients (29%) with 2-5 COT quarters, 420 patients (6.6%) with 6-10 COT quarters, and 680 patients (10.7%) with >10 COT quarters. A *longitudinal dataset* was created for the selected patients spanning 10 years (2008-2017) to capture a wider range of background clinical data. The dataset contained 1,089,600 outpatient, 20,449 inpatient, and 25,232 emergency department visits. Each visit is characterized by the ICD-10 codes corresponding to diagnoses, the medication ordered, and the laboratory results, which were mapped into UMLS. The retrospective review of the described de-identified longitudinal dataset has been approved by University of Washington Institutional Review Board as well as The University of Texas at Dallas Institutional Review Board.

Table 5.5. The impact of UMLS knowledge embeddings on the prediction of incidence of Opioid Use Disorder (OUD) and the achievement/persistence of Chronic Opioid Therapy (COT). HAN^{UMLS} incorporates UMLS knowledge embeddings whereas HAN does not.

	OUD		**COT**	
	HAN^{UMLS}	HAN	HAN^{UMLS}	HAN
F_1 Score	**0.843**	0.774	**0.778**	0.776
Sensitivity	**0.749**	0.629	**0.850**	0.773
Specificity	0.963	**0.981**	0.717	**0.792**
DOR	**77.86**	72.99	**14.30**	12.99
AUROC	**0.856**	0.784	**0.783**	0.778

5.4.3 Experimental Results and Discussions

The impact of the UMLS knowledge embeddings on clinical prediction of the incidence of Opioid Use Disorders (OUD) after onset of opioid therapy and Chronic Opioid Therapy (COT) achievement and persistance is presented in the section. Two models were trained: HAN^{UMLS}, configured as described in the Methods section, and HAN, a baseline model that does not make use of the UMLS knowledge embeddings. HAN^{UMLS} uses the UMLS embeddings while HAN skips the concatenation step (described in Section 5.4.1), using only the embeddings learned using word2vec. Both models are evaluated using F_1 score, sensitivity, specificity, Diagnostic Odds Ratio (DOR) and Area Under the Receiver Operating Characteristic curve (AUROC). The results, presented in Table 5.5, show that incorporating ontological knowledge in the form of UMLS knowledge embeddings improved performance when predicting Opioid Use Disorder while maintaining performance on Chronic Opioid Therapy achievement/persistence without major changes to the model.

The results also show that the UMLS knowledge embeddings improve the prediction of incidence of Opioid Use Disorder after onset of opioid therapy and Chronic Opioid Therapy achievement and persistence, out-of-the-box, by simply concatenating the UMLS knowledge embeddings with the traditional, WORD2VEC-style embeddings typically used in deep learning systems. We analyzed the attention weights assigned to each medical concept and to each class of concepts (i.e., ICD-10 codes, medications, and lab results) in the test set for both the

HAN^{UMLS} and HAN models to determine the impact of the knowledge embeddings on the model. Interestingly, including the UMLS knowledge embeddings in the HAN^{UMLS} model caused the model to pay more attention to diagnoses (attention weight of 0.617 vs 0.4942) and less attention to medications (0.2706 vs 0.4762) on average indicating that the inclusion of the UMLS knowledge embeddings made the diagnoses more informative for prediction than the medications. Moreover, the diagnosis with the highest average attention weight in the HAN^{UMLS} model is '*Chronic pain syndrome*' (C1298685) with an average attention weight of 0.5750 while in the HAN model, the diagnosis with the highest weight of 0.8092 was '*Predominant Disturbance of Emotions*'.

5.4.4 Lessons Learned

Through the use of the UMLS embeddings in a neural model for predicting the incidence of Opioid Use Disorder and Chronic Opioid Treatment achievement and persistence, we have found that these knowledge embeddings improve prediction. We also learned that the UMLS embeddings can be seamlessly incorporated into existing neural architectures operating on concept embeddings by simply concatenating the UMLS embeddings with the existing embedding for each concept. The same model is evaluated with and without the UMLS knowledge embeddings, showing that the embeddings improve results.

5.5 Knowledge Embeddings for Ontology Alignment

In this section, we present the *Knowledge-graph Alignment and Embedding Generative Adversarial Network* (KAEGAN) which learns (a) to represent the relational knowledge from two distinct biomedical ontologies in the form of knowledge embeddings and (b) to use them for ontology alignment, by also relying on their ontology semantics. Learning how to automatically align biomedical ontologies has been a long-standing goal, given the ever-growing

content of such ontologies and the many applications that rely on them. Because the knowledge graphs underlying biomedical ontologies enable neural learning techniques to acquire knowledge embeddings as representations of these ontologies, neural learning can also consider ontology alignments. In order to facilitate alignment, KAEGAN embeds the knowledge graphs associated with two distinct biomedical ontologies into the same semantic embedding space. KAEGAN uses a novel adversarial learning framework to learn alignment-oriented knowledge embeddings. More specifically, KAEGAN uses multiple generators to model interactions inside and across two knowledge graphs in order to embed them into the same semantic space, ensuring that aligned concepts will have similar embeddings. The ontology alignment is bootstrapped by iteratively predicting new alignments informed by the most similar alignment-oriented knowledge embeddings. We show that by jointly learning knowledge graph alignments and knowledge embeddings for any pair of biomedical ontologies, we improve the results of learning either knowledge embeddings or knowledge alignments in isolation. Therefore, the main novelty of KAEGAN arises from its adversarial framework which enables joint learning of knowledge alignments while learning how to discriminate valid relations in each knowledge graph, both with competitive results.

5.5.1 Joint Learning of Knowledge Graph Alignment and Embedding

To learn knowledge embeddings informing ontology alignments we have designed the *Knowledge-graph Alignment and Embedding Generative Adversarial Network* (KAEGAN), which learns (1) how to embed two distinct biomedical ontologies into the same semantic space such that pairs of concepts from different ontologies can be aligned based on their similarity, while also (2) encoding the semantics of both ontologies in the same, shared embedding space. In KAEGAN, knowledge alignments are learned using the structure of the ontologies, i.e., relying on (a) the relations spanning concepts and (b) the attributes of concepts. KAEGAN also leverages a Generative Adversarial (Goodfellow et al., 2014; Cai and Wang, 2018) framework

to learn high-quality alignment-oriented knowledge embeddings, which in turn are used to iteratively bootstrap the learning of ontology alignments. To describe the functionality of KAEGAN, we first present the way in which alignments between ontologies are learned and then we detail the way in which KAEGAN learns alignment-oriented embeddings.

Formally, let C_X and C_Y denote the set of concepts encoded in ontologies X and Y, respectively. An alignment $A = \{(c_x, c_y) \in C_X \times C_Y | c_x \equiv c_y\}$ is a set of pairs of concepts from X and Y that represent equivalent concepts across the ontologies. For example, the concept "*Arterial structure*" from SNOMED CT represents the same concept as "*Artery*" in the NCI Thesaurus (concept C0003842 in the UMLS), so there would be a pair (Arterial_structure, Artery) in A. It should be noted that $c_1 \equiv c_2$ cannot hold if c_1 and c_2 are from the same ontology, so a concept c_x from X can be aligned with at most one concept c_y from Y, and vice-versa. We consider a subset of A, A' as training data. This allows us to model alignment as a classification problem where the probability, $q(c_y|c_x)$, of a concept c_y being aligned with a concept c_x is a function of the similarity of their *concept embeddings*:

$$q(c_y|c_x) = \text{softmax}\left(\text{sim}(c_x, c_y)\right) = \frac{e^{\text{sim}(c_x,c_y)}}{\sum_{j \in Y} e^{\text{sim}(c_x,c_j)}} \tag{5.12}$$

$$\text{sim}(c_x, c_y) = \frac{\vec{v}(c_x) \cdot \vec{v}(c_y)}{||\vec{v}(c_x)||_2 ||\vec{v}(c_y)||_2} \tag{5.13}$$

where $\vec{v}(c_x)$ represents the concept embedding for c_x, whereas $\text{sim}(\cdot, \cdot)$ is computed by the cosine similarity.

Inspired by Trivedi et al. (2018), we adopt a *contextualized* concept embedding method that is particularly well suited to knowledge graph alignment. LinkNBed (Trivedi et al., 2018) contextualizes concept embeddings using (1) concept attributes and (2) the *neighborhood* of nearby concepts in the knowledge graph. In a biomedical ontology, concept attributes encode auxiliary information via attribute triples of the form $\langle c, t, v \rangle$ where c represents a concept, t represents an attribute type, and v represents the attribute value, usually a string. For example in the National Cancer Institute Thesaurus (Sioutos et al., 2007), the

concept c="*Eicosapentaenoic Acid*" has the attribute t="*definition*" value v="*A class of polyunsaturated fatty acids with 20 carbons and 5 double bonds*". To further contextualize each concept embedding, nearby concept embeddings are aggregated using random walks in the knowledge graph. Formally, the embedding $\vec{v}(c)$ for the concept c in KAEGAN is calculated as:

$$\vec{v}(c) = \sigma\left(\vec{v_0}(c) + W_n N_c(c) + W_a A_c(c)\right) \tag{5.14}$$

where σ is the sigmoid function, $\vec{v_0}(c) \in \mathbb{R}^d$ is an initial d-dimension embedding of c; $N_c(c)$ is the aggregate neighborhood context embedding of c; and $A_c(c)$ is the aggregate attribute embedding of c, while $W_n, W_c \in \mathbb{R}^{d \times d}$ are weight matrices. More specifically:

- The aggregate neighborhood context embedding $N_c(c)$ is computed by averaging the initial embedding vectors $\vec{v_0}(c_i)$ for each concept c_i in the *neighborhood* of c. The neighborhood of a concept c is approximated as the set of concepts other than c encountered on k random walks of length l executed when starting at c.

- The aggregate attribute embedding $A_c(c)$ is computed by max-pooling over the attribute embeddings of each attribute of c.

An attribute embedding a is calculated by passing an attribute type embedding a_t and an attribute value embedding a_v through a fully connected sigmoid layer: $a = \sigma(W_t a_t + W_v a_v)$, where $W_t, W_v \in \mathbb{R}^{d \times d}$ are weight matrices. Attribute type embeddings are learned from scratch for each attribute and attribute value embeddings are learned using PARAGRAPH2VEC (Le and Mikolov, 2014) as in Trivedi et al. (2018).

Using the concept embeddings to calculate $q(c_y|c_x)\ \forall (c_y, c_x) \in C_Y \times C_X$, we can measure the quality of a predicted alignment according to q using cross entropy:

$$-\sum_{x \in X} \sum_{y \in Y} \mathbf{1}_{[c_x \equiv c_y]} \log\, q(c_y|c_x) \tag{5.15}$$

However, Equation 5.15 will only measure how well the model captures similarity between concepts aligned in the training data, which represents a small subset of $C_X \times C_Y$. Inspired

by Sun et al. (2018), we extended Equation 5.15 to incorporate uncertainty for unlabeled alignments, using the function $\phi(c_x, c_y)$ in place of the indicator function of Equation 5.15:

$$\phi(c_x, c_y) = \begin{cases} \mathbf{1}_{[c_x \equiv c_y]} & \text{if } c_x \text{ is aligned in the training data} \\ \frac{1}{N_{unl}} & \text{if } c_x \text{ is unlabeled} \end{cases} \tag{5.16}$$

where N_{unl} is the number of currently unaligned concepts from Y. $\frac{1}{N_{unl}}$ represents a uniform distribution over the possible alignment candidates for c_x and serves to bias the system against erroneous alignments. Using ϕ in the cross-entropy calculation of Equation 5.15 in place of the indicator function, we obtain the Alignment Classification loss, $\mathcal{L}_C$:

$$\mathcal{L}_C = -\sum_{x \in X} \sum_{y \in Y} \phi(c_x, c_y) \log\, q(c_y | c_x) \tag{5.17}$$

By minimizing $\mathcal{L}_C$, KAEGAN learns the probability alignment function, q, which enables the maximum likelihood alignment between X and Y to be found. As in Sun et al. (2018), we produce an alignment between X and Y given an alignment probability function, q, by solving the max-weighted matching problem on the bipartite graph whose nodes are concepts from X and Y with edges $\langle c_x, c_y \rangle$ denoting alignment weighted by $q(c_y|c_x)$. We only consider alignment between each concept pair $\langle c_x, c_y \rangle$ for which $q(c_x, c_y)$ is above a certain threshold. It should be noted that this represents a max-weighted, one-to-one matching between subsets of X and Y with maximum total likelihood according to q.

Moreover, Sun et al. (2018) have shown that such alignments can be iteratively refined through bootstrapping by adding newly predicted alignments to the training data at each iteration and altering the predictions if a more likely alignment emerges. Specifically, at iteration i, given a training alignment A'_i (where $A'_0 = A'$), KAEGAN learns *alignment-oriented knowledge embeddings* for each concept in $C_X \cup C_Y$ using the adversarial method described below, after which the alignments are learned and new concept matches are added to A'_{i+1}. If a more likely matching emerges for a particular concept according to q, the less

likely matching is simply replaced. KAEGAN terminates when no new concept matchings are added to the alignment.

In order to learn alignment-oriented knowledge embeddings KAEGAN uses a Generative Adversarial Network (Goodfellow et al., 2014) (GAN) composed of a Discriminator and four Generators illustrated in Figure 5.6. In this work, we extend our GAN framework for learning knowledge graph embedding (Section 5.3) to learn alignment-oriented knowledge embeddings. In KAEGAN, the Discriminator learns embeddings that are used to discriminate between valid and invalid relation triples from both biomedical ontologies while also ensuring that any aligned concepts from the pair of ontologies have similar embeddings, given a probability of alignment between the concepts, defined as q. The relation triple $\tau = \langle c_1, r, c_2 \rangle$ is considered to be *valid* if the concept c_1 is related to the concept c_2 by the relation r in either of the knowledge graphs X or Y and is said to be *invalid* otherwise. As in Section 5.3, KAEGAN's Generators learn to produce more plausible – yet still invalid – relation triples in an attempt to fool the Discriminator. In order to facilitate alignment, the Discriminator is trained to evaluate the plausibility of both intra-graph relation triples produced by Generators G_X and G_Y and inter-graph relation triples produced by Generators G_{XY} and G_{YX}.

Consider two relation triples $\tau_x = \langle c_1^x, r^x, c_2^x \rangle$ and $\tau_y = \langle c_1^y, r^y, c_2^y \rangle$ from the knowledge graphs represented by ontologies X and Y, respectively shown in Figure 5.6. Attempting to fool the Discriminator, Generator G_X uses τ_x to generate an invalid triple $\tau_x' = \langle c_1^{x\prime}, r^{x\prime}, c_2^{x\prime} \rangle$ by swapping out either c_1^x or c_2^x with another concept from X such that τ_x', while invalid, is more plausible than a randomly sampled triple (e.g., $\tau_x = \langle$Artery, IS_A, Blood_Vessel$\rangle$ and $\tau_x' = \langle$Artery, IS_A, Blood_capillary$\rangle$). Generator G_Y does the same for triple τ_y, generating the invalid triple τ_y' using Y. The Discriminator's job is then to determine which of τ_x and τ_x' and which of τ_y and τ_y' is more plausible. Generators G_{XY} and G_{YX} operate similarly, however they generate invalid triples by sampling concepts across the knowledge graphs corresponding

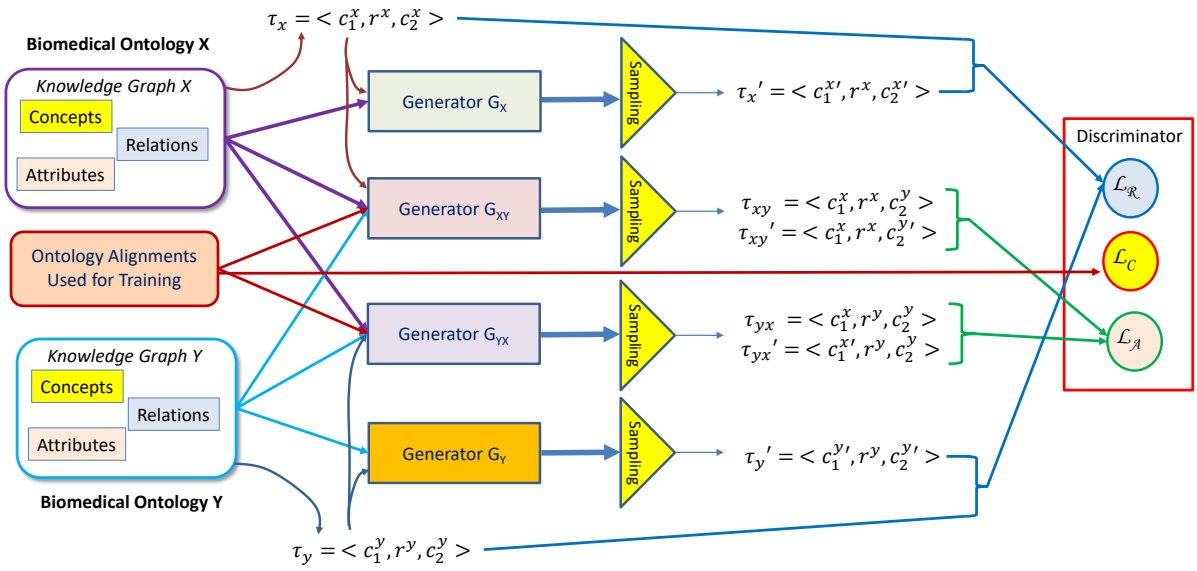

Figure 5.6. Adversarial Learning of Knowledge Graph Embeddings for Biomedical Ontology Alignment.

to the pair of biomedical ontologies. Specifically, G_{XY} generates the invalid triple τ'_{xy} by first relying on the valid triple τ_{xy} and replacing either c_1^x or c_2^x with its aligned concept from Y, c_1^y or c_2^y, based on an alignment A', available from the training data used for learning ontology alignments. Then G_{XY} samples a concept from Y to replace c_1^y or c_2^y to produce τ'_{xy}. For example, consider the triple $\tau_x = \langle$Artery, IS_A, Blood_Vessel$\rangle$ from the Foundational Model of Anatomy Ontology (FMA) (Rosse and Mejino Jr, 2003). Blood_Vessel could be replaced with the concept Human_Blood_Vessel from National Cancer Institute (NCI) Thesaurus (Golbeck et al., 2003) to create $\tau_{xy} = \langle$Artery, IS_A, Human_Blood_Vessel$\rangle$ and the generator might sample the semantically related concept Angiogram to produce $\tau'_{xy} = \langle$Artery, IS_A, Angiogram$\rangle$. G_{YX} operates in the same way to create τ_{yx} and τ'_{yx} from τ_y. In this case, the Discriminator needs to determine which of τ_{xy} and τ'_{xy} and which of τ_{yx} and τ'_{yx} is more plausible, using a loss function $\mathcal{L}_A$. The Discriminator also minimizes $\mathcal{L}_C$ from Equation 5.17 to jointly learn the similarity function q, yielding the final objective

function for the Discriminator:

$$\mathcal{L}_D = \mathcal{L}_R + \mathcal{L}_A + \mu\mathcal{L}_C \tag{5.18}$$

where μ is a hyper-parameter controlling the importance of the Alignment Classification loss. Next we shall describe how $\mathcal{L}_R$ and $\mathcal{L}_A$ are defined and evaluated.

In order to discriminate valid relation triples from invalid relation triples, the Discriminator is trained to measure the *plausibility* of a triple $\tau = \langle c_1, r, c_2 \rangle$ where $c_1, c_2 \in C_X \cup C_Y$ are concepts and r is a relation from either X or Y. As in Sections 5.2 and 5.3, each concept c and relation r is associated with learned embeddings $\vec{v}(c), \vec{v}(r) \in \mathbb{R}^d$, where d is the dimensionality of the embedding space. The concept embeddings $\vec{v}(c_1), \vec{v}(c_2)$ are computed using Equation 5.14 and the relation embedding $\vec{v}(r)$ is calculated via simple embedding lookup. Plausibility is measured using a scoring function: $f_D(\tau) = ||\vec{v}(c_1) + \vec{v}(r) - \vec{v}(c_2)||_{L2}$.

In order to learn the plausibility of the relations encoded in an ontology, KAEGAN minimizes the following marginal loss function:

$$\mathcal{L}_R = \sum_{\tau \in X \cup Y} \max\left(0, f_D(\tau) - \gamma_1\right) + \max\left(0, \gamma_2 - f_D(\tau')\right) \tag{5.19}$$

where γ_1, γ_2 are margin hyperparameters and $f_D(\tau')$ is the score of an invalid triple generated by either G_X or G_Y using τ. While the triple τ represents semantic knowledge that we wish to encode in our knowledge embeddings, τ' represents erroneous knowledge that KAEGAN should learn is implausible. The two margin parameters $\gamma_1, \gamma_2 > 0$ where $\gamma_2 > \gamma_1$ enforce the property that plausible triples have lower score than negative triples since $f(c_1', r, c_2') - f(c_1, r, c_2) \geq \gamma_2 - \gamma_1 > 0$. By modeling the margin with two parameters and setting γ_1 to a small positive value, we also enforce the property that plausible triples have low absolute scores, which has shown to be effective for alignment-oriented embedding (Sun et al., 2018; Zhou et al., 2017).

The corrupted triple τ' is sampled by either Generator G_X or Generator G_Y depending on whether the original triple τ was encoded in ontology X or in ontology Y. Given a

triple $\tau_x \in X$, Generator G_X generates the corrupted triple τ'_x by producing a probability distribution $P_X(\tau'_x|\tau_x)$ and sampling from that distribution. The distribution P_X is calculated using Generator G_X's own scoring function f_{G_X}:

$$P_X(\tau'_x|\tau_x) = \frac{e^{f_{G_X}(\tau')}}{\sum_{\tau^* \in Neg^X} e^{f_{G_X}(\tau^*)}} \tag{5.20}$$

where Neg^X is a set of invalid relation triples derived from $\tau_x = \langle c_1^x, r^x, c_2^x \rangle$ with either c_1^x or c_2^x replaced with another concept from ontology X such that the resulting relation triple does not occur in the knowledge graph X. KAEGAN uses the DistMult (Yang et al., 2015) scoring function, defined as $f_{G_X} = (\vec{v}(c_1) \odot \vec{v}(r)) \cdot \vec{v}(c_2)$, as in the Generators described in Section 5.3.2. Intuitively, P_X is used to sample the most plausible, yet still incorrect triples in an attempt to fool the Discriminator. Generator G_Y samples from the distribution P_Y calculated using the scoring function f_{G_Y} defined similarly.

For alignment-oriented embeddings, the scoring functions f_D, (of the Discriminator) and f_G (of the generators) should have the *replacement property* that if $c_1^x \in X$ has an aligned concept $c_1^y \in Y$, $f(c_1^x, r^x, c_2^x) \approx f(c_1^y, r^x, c_2^x)$ when f is either f_D or f_G. That is to say, if a concept from a plausible triple of ontology X is replaced with an aligned concept from ontology Y, the new triple should remain plausible under both scoring functions. The same property should hold if c_2^x is replaced with its aligned concept or if a concept from a triple from ontology Y is replaced with an aligned concept from ontology X. The replacement property of f_D and f_G ensures that relation semantics are preserved across knowledge graphs. In order to capture this property, we minimized the cross-graph marginal loss, defined as:

$$\begin{aligned} \mathcal{L}_A = &\sum_{\tau_x \in X^A} \max\left(0, f_D(\tau_{xy}) - \gamma_1\right) + \max\left(0, \gamma_2 - f_D(\tau'_{xy})\right) \\ &+ \sum_{\tau_y \in Y^A} \max\left(0, f_D(\tau_{yx}) - \gamma_1\right) + \max\left(0, \gamma_2 - f_D(\tau'_{yx})\right) \end{aligned} \tag{5.21}$$

where X^A, Y^A represent the subsets of triples in the knowledge graphs of the ontologies X and Y that involve concepts aligned across ontologies; τ_{xy} represents a triple $\tau_x = \langle c_1^x, r^x, c_2^x \rangle \in X$

with either c_1^x or c_2^x replaced with its aligned concept from Y, and τ'_{xy} represents a triple $\langle c_1^x, r^x, c_2^x \rangle \in X$ with either c_1^x or c_2^x replaced with an incorrect concept from Y that does not represent a true alignment, sampled by G_{XY}. τ_{yx} and τ'_{yx} are defined similarly.

5.5.2 Experimental Results and Discussions

The alignment-oriented knowledge embeddings learned by KAEGAN were evaluated in terms of their ability to (1) produce an alignment between two biomedical knowledge graphs, and (2) model the semantics of the two encoded knowledge graphs. For the evaluations, we used three ontologies available from BioPortal: SNOMED Clinical Terms (Donnelly, 2006), the National Cancer Institute (NCI) Thesaurus (Golbeck et al., 2003), and the Foundational Model of Anatomy (FMA) (Rosse and Mejino Jr, 2003). SNOMED CT, the largest of the three ontologies, encodes medical terms used in clinical documentation and reporting, consisting of 349,548 medical concepts as of January 31, 2019. The NCI Thesaurus is an ontology providing a reference terminology for 66,724 cancer and cancer-related medical concepts. The Foundational Model of Anatomy (FMA) is an ontology with the stated goal of representing the phenotypic structure of the human body, encoding 78,989 concepts. Each of these ontologies is distinct and was designed independently. Consequently, a subset of the concepts in each ontology also have representations in the other two ontologies. As each of these ontologies has been integrated into the Unified Medical Language System (UMLS) (Lindberg et al., 1993), we used UMLS as a reference alignment for evaluating the quality of the learned knowledge alignments. We were inspired by the Ontology Alignment Evaluation Initiative's (OAEI) Large BioMed Track (Achichi et al., 2017)[5], which in fact uses these three ontologies and the reference alignment provided by the UMLS to evaluate ontology alignment systems in a yearly competition. The OAEI Large BioMed track is an evaluation of ontology alignment systems aimed at aligning large biomedical ontologies. It

[5]http://www.cs.ox.ac.uk/isg/projects/SEALS/oaei/

Table 5.6. Alignment data distribution.

Alignment	**Training**	**Validation**	**Testing**
SNOMED-NCI	3,442	1,721	12,047
SNOMED-FMA	1,205	602	4,219
NCI-FMA	537	269	1,880

provides six sub-tasks aligning aligning subsets of the ontologies described above. However, the OAEI Large BioMed track is an evaluation for unsupervised systems and does not concern itself with distilling ontological knowledge into embeddings. As such, we have adapted the alignments provided by the OAEI to the supervised knowledge graph alignment problem. SNOMED CT, the NCI Thesaurus and FMA proved to be ideal ontologies for our work since they not only have a widely accepted alignment evaluation, but each ontology contains rich relational knowledge that can be used to inform the learning of knowledge embeddings, enabling KAEGAN to jointly learn knowledge graph alignment and embedding.

Ontology Alignment Evaluation

For the alignment task, we randomly sampled 20% of the aligned concepts of each pair of ontologies to create our initial training Alignment, A'_0, reserving 10% for validation, leaving the remaining 70% as test data, summarized in Table 5.6. For both tasks, we evaluate KAEGAN against the following alternate configurations of KAEGAN to explore the efficacy of the methods used therein:

1. **KA-Classifier** is a neural classifier that learns a similarity function $q(\cdot)$ which minimizes the objective in Equation 5.17, but does not use the relational information of the two knowledge graphs being aligned. KA-Classifier is meant to evaluate the effect of relational information on embedding-based alignment. KA-Classifier does not use either loss from Equations 5.19 or 5.21 and therefore can not be trained in an adversarial fashion.

2. **KAEGAN-No_Class** is a configuration of the KAEGAN system that does not use the classification loss, $\mathcal{L}_C$. KAEGAN-No_Class learns to embed and align concepts using *only* the relational information from the two knowledge graphs as represented by $\mathcal{L}_R$ and $\mathcal{L}_A$. This configuration serves to evaluate the effect of $\mathcal{L}_C$ on both alignment and embedding.

3. **KA-Discriminator** is a model comprised only of the Discriminator of KAEGAN, meant to evaluate the effect of the GAN training regime. KA-Discriminator is trained using standard negative sampling instead of the Generators.

4. **KAEGAN-No_Context** is a configuration of KAEGAN that uses simple embedding lookup for $\vec{v}(\cdot)$ in Equation 5.14, eschewing context information from attributes and concept neighborhoods.

5. **KAEGAN-No_Attr** is a configuration of KAEGAN that does not use attribute information, leaving $W_a A_c(c)$ out of Equation 5.14.

6. **KAEGAN-No_Neigh** is a configuration of KAEGAN that does not use neighborhood information, leaving $W_n N_c(c)$ out of Equation 5.14.

Following the Ontology Alignment Evaluation Initiative (Achichi et al., 2017), ontology alignment is evaluated using Precision (P), Recall (R), and F_1 as well as Hits@10 and Mean Reciprocal Rank as in Sun et al. (2018). Hits@10 reports the percentage of concepts c_x for which the true aligned concept $\hat{c_y}$ is one of the top-10 most likely matches according to $q(c_y|c_x)$. Mean reciprocal rank reports the average reciprocal rank of the correct aligned concept among all possible concepts, ranked by q.

The ontology alignment results are presented in Table 5.7. We compare the different configurations of KAEGAN to the top performing system from the OAEI Large BioMed track, AgreementMakerLight (AML) (Faria et al., 2013), a hand-engineered expert system which

Table 5.7. Evaluation of biomedical ontology alignment using alignment-oriented knowledge embeddings.

	SNOMED CT - NCI					SNOMED CT - FMA				
Model	P	R	F_1	H@10	MRR	P	R	F_1	H@10	MRR
KA-Classifier	0.841	0.573	0.682	0.851	0.773	0.820	0.606	0.697	0.843	0.740
KAEGAN-No_Class	0.665	0.592	0.626	0.700	0.587	0.676	0.512	0.583	0.660	0.577
KA-Discriminator	0.837	0.591	0.693	0.867	0.763	0.832	0.624	0.713	0.850	0.738
KAEGAN-No_Context	0.551	0.392	0.458	0.585	0.497	0.521	0.432	0.473	0.588	0.485
KAEGAN-No_Attr	0.557	0.389	0.458	0.590	0.504	0.523	0.427	0.470	0.580	0.494
KAEGAN-No_Neigh	0.840	0.588	0.692	0.866	0.757	0.824	0.616	0.705	0.842	0.743
KAEGAN	**0.857**	**0.613**	**0.715**	**0.890**	**0.784**	**0.847**	**0.631**	**0.723**	**0.867**	**0.766**
AML (Faria et al., 2013)	0.904	0.668	0.768	–	–	0.882	0.687	0.772	–	–
	NCI - FMA					**Macro-Average**				
Model	P	R	F_1	H@10	MRR	P	R	F_1	H@10	MRR
KA-Classifier	0.817	0.656	0.727	0.824	0.769	0.826	0.612	0.702	0.839	0.761
KAEGAN-No_Class	0.675	0.558	0.611	0.691	0.616	0.672	0.554	0.607	0.684	0.593
KA-Discriminator	0.827	0.657	0.732	0.840	0.769	0.832	0.624	0.713	0.852	0.757
KAEGAN-No_Context	0.537	0.476	0.503	0.541	0.519	0.536	0.433	0.478	0.571	0.500
KAEGAN-No_Attr	0.534	0.468	0.499	0.544	0.519	0.538	0.428	0.476	0.572	0.505
KAEGAN-No_Neigh	0.818	0.667	0.734	0.832	0.773	0.827	0.624	0.710	0.847	0.758
KAEGAN	**0.842**	**0.682**	**0.753**	**0.849**	**0.792**	**0.849**	**0.642**	**0.730**	**0.869**	**0.781**
AML (Faria et al., 2013)	0.838	0.872	0.855	–	–	0.875	0.742	0.798	–	–

leverages background information found outside the ontologies being aligned. While not at the level of AML, clearly, the full KAEGAN model out-performs the alternate configurations for each matching. Interestingly, the worst performing configurations were the KAEGAN-No_Context and KAEGAN-No_Attr models, illustrating the importance of attribute information for ontology alignment. Moreover, the strong performance of the KA-Classifier model and the relatively poor performance of the KAEGAN-No_Class model demonstrate the efficacy of modeling alignment matching explicitly through the alignment classification loss, $\mathcal{L}_C$.

In general, the results show that KAEGAN is able to embed distinct knowledge graphs into the same semantic space such that the learned embeddings can be used for both ontology alignment and representation learning. The ontology alignment experiments presented in Table 5.7 elucidate several interesting phenomena including the importance of attributes in ontology alignment. Since attributes are useful in qualifying the concepts - making them distinct from the other concepts in the ontology - it is not unexpected that ignoring them results

in the large performance drops seen by KAEGAN-No_Context and KAEGAN-No_Attr. The name and description attributes are particularly helpful, since many concepts have similar names or descriptions across ontologies. However, the performance of KAEGAN-No_C demonstrates the efficacy of leveraging the structured knowledge in the form of relation triples for alignment. One possible reason for this is that concepts with differing names and descriptions from different ontologies may share similar relations with more well-defined concepts, causing the model to discover their alignment. Another interesting trend we can glean from the alignment experiments is the small improvement of Hits@10 over Precision, a fundamentally more difficult evaluation. Precision is akin to 'Hits@1', but when we relax the evaluation to the top 10 we only see a 2% increase on average. This finding indicates that when the model finds an alignment it is generally correct, but when the model is not able to determine alignment correctly, the probability it learns for the correct alignment is especially low.

While the alignment results show promise, they do not surpass the current state-of-the-art on the OAEI Large BioMed track, represented by AgreementMakerLight (AML) (Faria et al., 2013). AML is hand-engineered for aligning large biomedical ontologies, leveraging outside ontologies (Mungall et al., 2012; Lipscomb, 2000; Kibbe et al., 2014) as intermediaries to facilitate matching. We believe KAEGAN can be improved in the same way – by jointly modeling the alignment more than two ontologies. Likewise, noting that AML makes use of string matching, we believe KAEGAN can be improved by incorporating character-level information about a concept's name and other attributes.

Knowledge Embedding Evaluation

To determine the ability of the learned knowledge embeddings to model the semantics of the embedded knowledge graphs, we evaluate the embeddings using RTC-P, RTC-R, PPA, H@10, and MRR (described in Sections 5.2.2 and 5.3.3) using the plausibility function $\rho = -f_D$.

Table 5.8. Biomedical knowledge embeddings evaluated by their ability to model knowledge graph plausibility and completeness.

	SNOMED CT					NCI Thesaurus				
Model	RTC-P	RTC-R	PPA	H@10	MRR	RTC-P	RTC-R	PPA	H@10	MRR
KAEGAN-No_Class	0.873	0.860	0.971	0.417	0.304	0.756	0.737	0.943	0.707	0.485
KA-Discriminator	0.883	0.866	0.958	0.410	0.301	0.726	0.719	0.914	0.687	0.461
KAEGAN-No_Context	0.850	0.837	0.938	0.386	0.274	0.720	0.721	0.905	0.677	0.450
KAEGAN-No_Attr	0.850	0.847	0.938	0.395	0.281	0.725	0.721	0.925	0.684	0.450
KAEGAN-No_Neigh	0.890	0.869	0.967	0.431	0.304	0.763	0.756	0.938	0.719	0.475
KAEGAN	**0.897**	**0.871**	**0.974**	**0.432**	**0.316**	**0.764**	**0.757**	**0.962**	**0.723**	**0.486**
	FMA					**Macro-Average**				
Model	RTC-P	RTC-R	PPA	H@10	MRR	RTC-P	RTC-R	PPA	H@10	MRR
KAEGAN-No_Class	0.775	0.767	0.963	0.526	0.391	0.801	0.788	0.959	0.550	0.393
KA-Discriminator	0.770	0.765	0.951	0.511	0.396	0.793	0.783	0.941	0.536	0.386
KAEGAN-No_Context	0.744	0.722	0.920	0.354	0.341	0.771	0.760	0.921	0.472	0.355
KAEGAN-No_Attr	0.748	0.733	0.922	0.364	0.352	0.774	0.767	0.928	0.481	0.361
KAEGAN-No_Neigh	0.781	**0.771**	0.959	0.502	0.403	0.811	0.799	0.955	0.551	0.394
KAEGAN	**0.784**	0.771	**0.963**	**0.548**	**0.406**	**0.815**	**0.800**	**0.966**	**0.568**	**0.403**

To perform the evaluations, we split the relation triples from each knowledge graph into 85%/5%/10% train/validation/test splits.

The results presented in Table 5.8 indicate that KAEGAN is able to model the semantics of distinct knowledge graphs by embedding them in the same semantic space facilitated by modeling their alignment. The KA-Classifier model does not perform knowledge graph embedding, so it is not evaluated along with the other configurations. Likewise, we do not evaluate KAEGAN against general-purpose, non-alignment-oriented embedding models (e.g., Trans-X, SePLi(Wu et al., 2015), or TATEC(García-Durán et al., 2015)) in order to focus on the impact of modeling alignment on knowledge graph embedding. Although it should be noted that the general KAEGAN learning framework can accommodate any general purpose embedding paradigm, such as the ones listed above, by replacing Equation 5.14. The improvement of KAEGAN over KAEGAN-No_Class shows that using alignment classification improves relational learning. Likewise, the improvement of KAEGAN over KAEGAN-Discriminator demonstrates the efficacy of the GAN learning framework. It should be noted that, while the removal of concept context in the form of attributes and neighborhoods has a

marked effect on performance, the decrease is minimal compared to its effect the alignment task.

The results presented in Table 5.8 indicate that biomedical knowledge embeddings learned from distinct graphs are improved by jointly modeling the alignment between the graphs. We believe the primary reason for this is that modeling both graphs in the same semantic space allows each graph can inform the other - filling possible knowledge gaps and providing additional context for both overlapping concepts and the rest of the graphs. Guo et al. (2015) have shown that *semantically smooth knowledge embeddings* can be achieved by imposing constraints on the learned embeddings. The alignment classification loss $\mathcal{L}_C$ plays a similar role in KAEGAN constraining aligned embeddings to be similar to one another, imposing its own type of smoothing which is shown to improve results.

5.5.3 Lessons Learned

The *Knowledge-graph Alignment and Embedding Generative Adversarial Network* (KAEGAN) was able learn how to jointly embed distinct biomedical ontologies into the same semantic embedding space such that the resulting embeddings can be used for knowledge graph alignment. KAEGAN leverages relational knowledge encoded in the knowledge graph of an ontology as well as attributes of the medical concepts to learn alignment-oriented embeddings. In addition to ontology alignment, the learned embeddings can be used to model the semantics of the encoded knowledge graphs. Results indicate that jointly learning to align and embed the knowledge graphs improves upon learning the alignment and the embedding separately. Moreover, the results suggest that the multi-generator GAN framework modeling both intra- and inter-graph relations improves representation quality for both graphs, leading to a better learned alignment.

CHAPTER 6

THE IMPACT OF KNOWLEDGE EMBEDDINGS ON RELATION EXTRACTION IN CLINICAL NARRATIVES

In this chapter, we investigate the use of knowledge graph embeddings for extracting relations between medical concepts in discharge summaries. Discharge summaries use many medical concepts to convey important information, as discussed in Chapter 2. The relations between medical concepts expressed in text capture additional important clinical information, as it was shown to tbe the case in EEG reports discussed in Chapter 3. Relation extraction from discharge summaries is a well-studied topic in clinical informatics, with the 2010 i2b2/VA challenge (Uzuner et al., 2011) proving to be a useful benchmark for (a) relation extraction as well as (b) concept detection and (c) assertion classification. The top-performing submission for the relation extraction task of the 2010 i2b2/VA challenge used an SVM-based method operating on hand-engineered features, including dependency parse patterns and biomedical ontologies (Rink et al., 2011). Later, specialized end-to-end neural methods with no feature extraction were designed for the relation extraction task, but these methods were not able to achieve the performance of the feature-based method (Luo et al., 2017; Li et al., 2019), probably due to the comparatively small size of the dataset. However, the breakthrough of pre-trained neural language models was able to surpass the state-of-the-art set by Rink et al. (2011) using the Bidirectional Encoder Representations from Transformers (BERT) model (Devlin et al., 2019).

The pre-training of the BERT model offers the advantage that it eliminates the need for large training corpora because it learns most of the weights of the network on massive language modeling data. The pre-trained BERT model is fine-tuned for an end-task via transfer learning. In order to adapt BERT to medical relation extraction, it was first pre-trained on general-domain text, then pre-trained further on medical text, and finally fine-tuned to relation extraction as described in Section 2.6 (Peng et al., 2019). The resulting model, called

BlueBERT, set a new state-of-the-art for relation extraction in discharge summaries as well as several other tasks (Peng et al., 2019). The BlueBERT network models binary relation extraction at the sentence-level by (a) replacing two concept mentions in the same sentence with special tokens denoting their types; and (b) performing relation extraction as sentence classification. For example, the sentence "*An echocardiogram revealed a pericardial effusion.*" is changed into "*An* @TEST$ *revealed a* @PROB$.". The test "*echocardiogram*" is replaced by the special token @TEST$ denoting a test, and the medical problem "*pericardial effusion*" is replaced by @PROB$ denoting a problem. By using the special tokens, the transformed sentence can be seen as a masked token sequence. This masked sequence is fed into BERT to produce a relation type prediction. This allows the BlueBERT network to focus on learning textual cues indicating relations by ignoring the textual expression of the concepts that constitute teh arguments of each relation.

However, by completely ignoring the concepts being related, BlueBERT is ignoring important information that could lead to better relation prediction performance. In this chapter, we present the Knowledge-Informed BERT (KIBERT) model for relation extraction that incorporates concept information into the BlueBERT model to improve performance. Two forms of concept information are incorporated by KIBERT: (1) lexical information and (2) background knowledge. Lexical information is incorporated by using the full input sentence, without BlueBERT's type-token masking, while still allowing the model to focus on contextual cues indicating relations via a novel attention-masking strategy. Background knowledge is incorporated in the form of *knowledge embeddings.* In Chapter 5, knowledge embeddings are presented for distilling knowledge from biomedical ontologies into real-valued embeddings that can be incorporated into deep learning architectures. Particularly, in Section 5.3 knowledge embeddings from the Unified Medical Language System (UMLS) (Lindberg et al., 1993) were presented representing medical concepts, groups of medical concepts called *semantic types*, and relations between them. Such UMLS relations capture relevant information for

the task of relation extraction from clinical text. For instance, the *may treat* relation defined in the UMLS (e.g., ⟨*alosetron, may treat, abdominal pain*⟩) could help determine if a treatment is a recognized treatment for a medical problem mentioned in the same sentence. However, associating the mention of a medical concept with its ontological representation in a structured ontology, known as *entity disambiguation* is known to be a difficult task (Mohan and Li, 2019). In the 2010 i2b2/VA Challenge Dataset, medical concept mentions are not annotated with UMLS codes, however each concept is annotated with one of three types: (1) medical problem; (2) test; or (3) treatment. In KIBERT, each of these concept types is associated with a *type embedding* derived from the UMLS Semantic Type embeddings learned in Section 5.3. When classifying the relation between two medical concepts, KIBERT uses the type embedding of each concept to inform the classification decision. In this chapter, we show that such type embeddings learned via knowledge graph embedding improve relation extraction performance.

The remainder of this chapter is formatted as follows: Section 6.1 provides details about the task and the dataset, Section 6.2 presents the KIBERT model for relation extraction from discharge summaries, Section 6.3 presents experimental results, and Section 6.4 concludes the chapter.

6.1 Data

The 2010 i2b2/VA Shared Task on Challenges for Clinical Data (Uzuner et al., 2011) provides a dataset of 871 discharge summaries annotated with medical concepts and assertions, as described in Section 2.3, as well as relations between annotated concepts. The original dataset of 871 discharge summaries was made available only to teams that participated in the Shared Task. However, a smaller dataset of 426 discharge summaries was provided publicly to foster research in clinical Information Extraction (IE). In order to compare the

Table 6.1. Relation Annotation Definitions and Examples from the 2010 i2b2/VA Challenge Dataset. Definitions and examples are taken from the Task Annotation Guidelines (Uzuner et al., 2011). Subscripts P, TR, and TE indicate medical problems, treatments, and tests, respectively.

Relation	**Definition**	**Example**
TrIP	Treatment improves medical problem	[Hypertension]$_{\text{P}}$ *was controlled by* [hydrochlorothiazide]$_{\text{TR}}$
TrWP	Treatment worsens medical problem	*culture taken from the lumbar drain showed* [Staphylococcus aureus]$_{\text{P}}$ *resistant to* [Nafcillin]$_{\text{TR}}$
TrCP	Treatment causes medical problem	[Bactrim]$_{\text{TR}}$ *could be a cause of* [these abnormalities]$_{\text{P}}$
TrAP	Treatment is administered for medical problem	[Dexamphetamine]$_{\text{TR}}$ *2.5 mg. p.o. q. A.M. for* [depression]$_{\text{P}}$
TrNAP	Treatment is not administered because of medical problem	[Relafen]$_{\text{TR}}$ *which is contra-indicated because of* [ulcers]$_{\text{P}}$
TeRP	Test reveals medical problem	[an echocardiogram]$_{\text{TE}}$ *revealed* [a pericardial effusion]$_{\text{P}}$
TeCP	Test conducted to investigate medical problem	*suggest* [echocardiogram]$_{\text{TE}}$ *to check for* [vegetation]$_{\text{P}}$
PIP	Medical problem indicates medical problem	[Azotemia]$_{\text{P}}$ *presumed secondary to* [sepsis]$_{\text{P}}$

methods presented in this chapter with state-of-the-art systems, we consider the smaller dataset of 426 discharge summaries.

The 2010 i2b2/VA challenge dataset contains annotations of eight types of relations between medical problems, tests, and treatments annotated in the discharge summaries. These relations are defined in Table 6.1 along with a prototypical example of each relation type provided by the task annotation guidelines[1]. There are five relation types between treatments and medical problems (TrIP, TrWP, TrCP, TrAP, and TrNAP), two relation types between tests and medical problems (TeRP and TeCP) and a single relation type between medical problems (PIP). Each of the examples listed in Table 6.1 illustrate clinical language typical of how relations are expressed in discharge summaries.

[1]https://www.i2b2.org/NLP/Relations/assets/Relation%20Annotation%20Guideline.pdf

Table 6.2. Relation Annotations Statistics from the 2010 i2b2/VA Challenge Dataset.

Relation	Train	Validation	Test
TrIP	49	2	152
TrWP	24	0	109
TrCP	171	13	342
TrAP	800	85	1,732
TrNAP	52	10	112
TeRP	903	90	2,060
TeCP	149	17	338
PIP	659	96	1,448

The dataset is split into 170 training documents and 256 testing documents. In order to facilitate comparison against the state-of-the-art system for relation extraction on the 2010 i2b2/VA Challenge dataset (Peng et al., 2019), we reserve the same 16 documents from the training set on which to perform validation. Annotation statistics are provided in Table 6.2.

6.2 Methods

This section presents the Knowledge-Informed BERT (KIBERT) model for relation extraction from discharge summaries. KIBERT augments a pre-trained medical language model architecture with knowledge embeddings learned from the UMLS Metathesaurus and Semantic Network knowledge graphs. Semantic type embeddings are used to infuse background knowledge into the model to inform relation classification decisions. Moreover, a novel attention-masking mechanism is applied to allow the model to develop simultaneously (a) a simplified representation of sentence context indicating relations (as in BlueBERT); and (b) contextualized representations of concept mentions. The KIBERT model is depicted in Figure 6.1, contrasted with the BlueBERT model.

Both KIBERT and BlueBERT operate at the sentence level, considering a pair of medical concepts mentioned in the same sentence. The two models perform multi-class classification to predict the type of relation (if any) between the two concepts, generating a probability

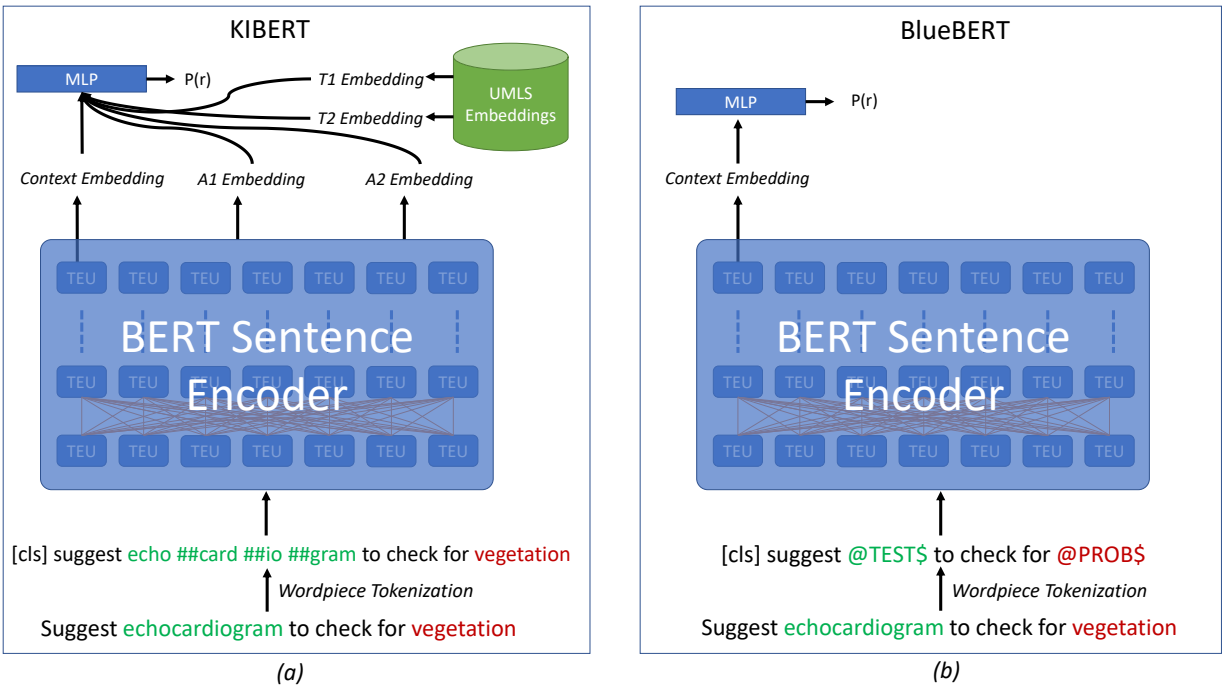

Figure 6.1. Neural architectures for identifying relation in clinical narratives. (a) The architecture of the Knowledge-Informed BERT (KIBERT) model. (b) The Architecture of BlueBERT.

distribution over relation types $P(r), r \in$ [TrIP, TrWP, TrCP, TrAP, TrNAP, TeRP, TeCP, PIP, NONE]. The BlueBERT model replaces the mentions of the two potentially related medical concepts with special tokens indicating the type of the concept, while KIBERT uses the full input sentence. In both models, the input sentences are then word-piece tokenized and fed to the same pre-trained BERT model (Peng et al., 2019). During word-piece tokenization, a special [CLS] token is prepended to the input sequence as in (Devlin et al., 2019). This [CLS] token is used by BERT to generate an aggregate embedding of the full sentence that is used as a context embedding by both models, referred to as the *context embedding*. In addition to the context embedding, KIBERT uses the token-level BERT outputs to generate embeddings for the two arguments of the potential relation to be classified, the A1 embedding and the A2 embedding. This process is described in more detail in the Section 6.2.1. While KIBERT uses the same pre-trained BERT model as BlueBERT, KIBERT

introduces a novel attention-masking strategy to allow the model to simultaneously generate (a) a context embedding that ignores concept mentions and (b) contextualized argument embeddings that consider the full input sequence. This attention masking strategy is detailed in Subsection 6.2.1. BlueBERT feeds the context embedding into a fully connected layer to produce $P(r)$. KIBERT combines the context and argument embeddings along with knowledge embeddings representing the type of each argument (i.e., problem, test, or treatment), described in Subsection 6.2.2. These five embeddings are fed into a two-layer multi-layer perceptron (MLP) with Gaussian Error Linear Unit activations (Hendrycks and Gimpel, 2016) to produce the distribution $P(r)$. Both models are trained using the cross-entropy loss function:

$$\mathcal{L} = -\sum_{s} \sum_{i \neq j \in s} \mathrm{I}[y_{ij} = \mathrm{argmax} P(r)_{ij}] \log(P(r)_{ij}) \tag{6.1}$$

where s is a sentence in the corpus, i and j are pairs of concepts in s, y_{ij} is the type of relation between i and j, and $\mathrm{I}[y_{ij} = \mathrm{argmax} P(r)_{ij}]$ is an indicator function that returns 1 if the most probable predicted relation type is correct and 0 otherwise.

6.2.1 Relation-context Transformer Encoder Attention Masking

Relation-context Transformer Encoder Attention Masking (RTEAM) is a novel attention-masking strategy whereby the same BERT model is used to simultaneously generate (1) a *context embedding* representing the simplified sentence context with medical concept mentions removed; and (2) fully-contextualized token-level representations of each medical concept mention. The performance of the BlueBERT model indicates that the strategy of simplifying the input sentence to focus on sentence context to uncover textual cues indicating relations is efficacious in this task. BlueBERT replaces the two medical concept mentions corresponding to the arguments of the potential relation under consideration with special tokens indicating the type of the replaced medical concept, depicted in Figure 6.1.

The decision of BlueBERT to ignore medical concept mentions is motivated by the non-standard language and complex lexicalizations of medical concepts in clinical narratives like discharge summaries. Lee et al. (2019) report that this strategy out-performs an alternative whereby concept mentions are not removed, but wrapped with pre-defined tags (e.g., <PROB></PROB>), indicating that the pre-trained BERT language model performs better when operating on this simplified input sentence. However, while context is certainly important, the text of medical concept mentions is also a valuable signal indicating relationships. Moreover, medical concepts mentioned in clinical records are often context dependent (e.g., the meaning of the token "*heart*" changes drastically if it is followed by "*surgery*" as opposed to "*disease*"). RTEAM allows KIBERT to learn fully-contextualized representations of medical concepts, while simultaneously learning a simplified context representation that ignores medical concepts, as in BlueBERT.

The Relation-context Transformer Encoder Attention Masking modifies the connections between Transformer Encoder layers in BERT depending on whether a token is (a) a part of a medical concept mention; or (b) a part of the sentence context. The KIBERT model with RTEAM is depicted in Figure 6.2. The BERT Transformer Encoder is comprised of 12 layers of Transformer Encoder Units (TEU), one for each token of the input sentence. Recall from Section 3.4.1 that in a traditional Transformer Encoder, each TEU is connected to every other TEU from the previous layer, informing each token's representation with the representations of every token in the full input sequence. RTEAM is used to limit the attention connections for tokens corresponding to relation context to simulate the simplified sentence representation used by BlueBERT, without removing the relation arguments from the input sentence. RTEAM is applied to each TEU associated with a context token (e.g., [CLS], *suggest, to, check, for* in Figure6.2) such that context tokens only attend to other context tokens in the input sentence, ignoring relation arguments. Figure 6.2 illustrates RTEAM masking between layers 1 and 2. For clarity, the RTEAM mask is shown for only

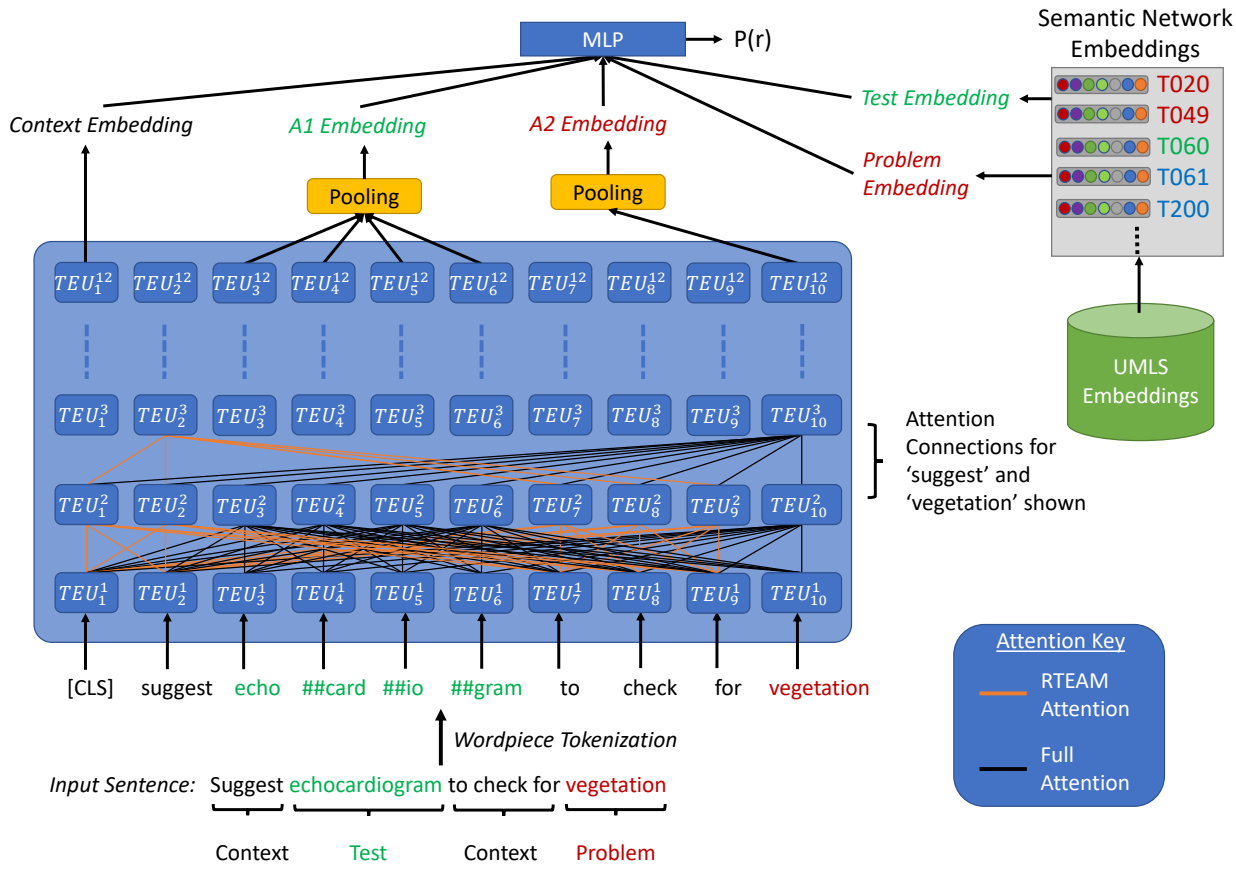

Figure 6.2. Knowledge-Informed BERT (KIBERT) for relation extraction in clinical narratives.

two units (TEU_2^3, TEU_{10}^3) in layer 3 to illustrate how the mask operates for context tokens and argument tokens. The context token, "*suggest*", is associated with TEU_2^3 while the argument token, "*vegetation*", is associated with TEU_{10}^3. Because it is associated with a context token, TEU_2^3 is only connected to the units from layer two corresponding to other context tokens (TEU_1^2, TEU_2^2, TEU_7^2, TEU_8^2, TEU_9^2). In contrast, TEU_{10}^3 is connected to every token in layer two because the full context may have important cues that inform the model's representation of the argument associated with TEU_{10}^3.

The Transformer Encoder is defined formally in Section 3.4.1 through Equations 3.11-19. Using RTEAM, the attention values, α_{ijh}, defined in Equation 3.13 is replaced by $\alpha'_{ijh} = \alpha_{ijh} \times C[i]$, where i, j are token indexes into the input sequence, h is an attention

head[2], and $C[i]$ is an indicator function that returns 1 if token i is a context token and 0 otherwise. The rest of the TEU functions exactly as described in the original Transformer Encoder (Vaswani et al., 2017), performing multi-headed attention over the connected cells from the previous layer, followed by a feed-forward layer. The effect of this attention masking is that the representation of each context token output by BERT is only informed by the other context tokens in the input sequence, ignoring complicated medical concept mentions such that the focus remains textual cues indicative of relations between pairs of medical concepts. However, KIBERT is able to also simultaneously generate fully-contextualized representations of relation argument tokens, attending to each token from the input sequence at each layer.

The output of the TEU in the final layer corresponding to the leading [CLS] token is considered the resulting context embedding, $c = TEU_1^{12}$, as suggested by Devlin et al. (2019), as it was also considered BlueBERT. The output of each TEU corresponding to the relation arguments are pooled to generate the argument embeddings: $a_1 = \psi(TEU_i^{12}|i \in A1)$, and $a_2 = \psi(TEU_j^{12}|j \in A2)$ where ψ is the mean-pooling operation; while tokens $i \in A1$ correspond to the tokens in the first argument and tokens $j \in A2$ correspond to the tokens in the second argument of the relation. The context embedding is concatenated with the argument embeddings and type embeddings learned from the UMLS, as described in Section 6.2.2, and fed into the MLP as depicted in Figure 6.2.

6.2.2 Knowledge Embeddings for Identifying Relations in Discharge Summaries

Knowledge embeddings derived from the UMLS are used to represent the type of each relation argument in KIBERT. Recall from Section 6.1 each of the eight relation types in the 2010 i2b2/VA Challenge Dataset are constrained to connect two specific concept types. TrIP,

[2]Recall that BERT is a Transformer model that performs multi-headed attention as described in Section 3.4.1.The same attention mask is used for each attention head.

Table 6.3. UMLS Semantic Types used by KIBERT to represent Medical Problems, Tests, and Treatments in the 2010 i2b2/VA Challenge Dataset.

Code	Name	KIBERT Concept Type
T060	Diagnostic Procedure	Test
T061	Therapeutic or Preventive Procedure	Treatment
T200	Clinical Drug	Treatment
T127	Vitamin	Treatment
T020	Acquired Abnormality	Problem
T190	Anatomical Abnormality	Problem
T049	Cell or Molecular Dysfunction	Problem
T019	Congenital Abnormality	Problem
T047	Disease or Syndrome	Problem
T050	Experimental Model of Disease	Problem
T033	Finding	Problem
T037	Injury or Poisoning	Problem
T048	Mental or Behavioral Dysfunction	Problem
T191	Neoplastic Process	Problem
T046	Pathologic Function	Problem
T184	Sign or Symptom	Problem

TrWP, TrCP, TrAP, and TrNAP relations occur between treatments and problems; TeRP and TeCP relations occur between tests and problems; and PIP relations occur between pairs of medical problems. In BlueBERT, type information is provided in the form of a learned token embeddings that replace each argument mention. As KIBERT does not perform this type-token replacement, it does not have access to type information.

Semantic Type knowledge embeddings learned from UMLS present an ideal solution to this problem. The semantic type embeddings are learned to represent an amalgamation of the concept embeddings for each concept of that type as well as capture the semantics of the Semantic Network knowledge graph as described in Section 5.3. The UMLS defines 127 semantic types in the Semantic Network that represent types of medical concepts from the Metathesaurus. KIBERT uses some of these semantic types to represent medical problems, tests, and treatments, defined in Table 6.3.

The 'Diagnostic Procedure' semantic type is used to represent tests, while there are three semantic types used to represent treatments: 'Therapeutic or Preventive Procedure', 'Clinical Drug', and 'Vitamin'. There are twelve semantic types associated with medical problems, defined by the 'Disorder' semantic group. Semantic Groups[3] are groups of semantic types defined by the UMLS. While there is only a single semantic type associates with tests, there are multiple semantic types related to treatments and medical problems. Therefore, we considered the centroid of each semantic type embedding associated with a specific concept type as the type embedding in KIBERT.

It should be noted that in Section 5.3 knowledge embeddings representing medical concepts are presented along with embeddings representing semantic types. Such concept-level knowledge embeddings would provide fine-grained knowledge that is particularly relevant to the end-task of relation extraction. However, medical concepts in the 2010 i2b2/VA challenge dataset are not annotated with gold-standard concept codes. Therefore, without a perfect entity disambiguation solution, we were not able to accurately associate medical concept mentions with their corresponding UMLS concept embeddings. In Section 6.3, we explore the use of an off-the-shelf biomedical entity linker (Neumann et al., 2019) to assign each medical concept mention to a UMLS concept but it is shown to degrade performance.

6.2.3 Training Details

KIBERT is trained using cross-entropy defined in Equation 6.1. The predicted distribution over relations types, $P(r)$, is produced by concatenating the context embedding, c, the argument embeddings, a_1, a_2, and the type embeddings t_1, t_2 and feeding that into a two layer MLP with GELU (Hendrycks and Gimpel, 2016) activations followed by a softmax

[3]https://metamap.nlm.nih.gov/Docs/SemGroups_2018.txt

layer:

$$h_1 = \text{GELU}(W_{h1}[c, a_1, a_2, t_1, t_2] + b_{h1}) \tag{6.2}$$

$$h_2 = \text{GELU}(W_{h2}h_1 + b_{h2}) \tag{6.3}$$

$$P(r) = \text{softmax}(W_p h_2 + b_p) \tag{6.4}$$

where $W_{h1} \in \mathbb{R}^{256\times(3d+2k)}, W_{h2} \in \mathbb{R}^{128\times256}, W_p \in \mathbb{R}^{9\times128}$ are weight matrices, d is the hidden size of BERT, k is the size of the type embeddings, and $b_{h1} \in \mathbb{R}^{256}, b_{h2} \in \mathbb{R}^{128}, b_p \in \mathbb{R}^9$ are bias vectors. Note that the dimensionality of the distribution, $P(r)$, is 9, due to the extra 'NONE' class signifying no relation.

In order to facilitate a comparison with BlueBERT, we adopt the same training regime and hyperparameters, where applicable (Peng et al., 2019). KIBERT is trained for 10 epochs with a learning rate of 5e-5, 1 epoch of learning rate warmup, and linear decay with a rate of 0.99. A dropout of 0.1 is used in all layers, including BERT, during training. We use the NCBI BERT (base) model trained on PubMed and MIMIC that performs best with BlueBERT (Peng et al., 2019), based on the BERT (base) model (Devlin et al., 2019). The hidden size of this BERT model is $d = 768$ and the size of the semantic type embeddings was $k = 50$. Due to the increased capacity of the inputs to the prediction layer compared to BlueBERT, (i.e., the argument and type embeddings) we found that a two-layer GELU MLP was necessary to facilitate training, described in Equations 6.3-4.

6.3 Experimental Results and Discussions

KIBERT is evaluated on the 256 test documents of the 2010 i2b2/VA Challenge Dataset using the standard metrics of micro-averaged precision, recall and F_1 score used by the task organizers. KIBERT is evaluated against NCBI BlueBERT as well as four other baseline systems. Rink et al. (2011) was the winning submission of the original challenge, using an SVM-based approach operating on hand-crafted features generated from dependency parse

Table 6.4. Relation Extraction evaluation of KIBERT against five baselines on the 2010 i2b2/VA Challenge Dataset measured using Precision, Recall, and F_1 score. Top scores bolded.

System	Precision	Recall	F_1
D'Souza and Ng (2014)	72.9	66.7	69.6
He et al. (2019)	73.1	66.7	69.7
Luo et al. (2017)	68.7	73.7	71.1
Rink et al. (2011)	72.0	75.3	73.7
NCBI BlueBERT	**78.4**	74.4	76.4
KIBERT	75.0	**81.2**	**78.0**

patterns and biomedical ontologies. D'Souza and Ng (2014) present an ensemble approach also based on features. He et al. (2019) and Luo et al. (2017) present deep-learning models learned from scratch using convolutional neural networks, however neither approach attains the performance of the original baseline. The results are presented in Table 6.4

While BlueBERT attains the highest precision, KIBERT out-performs the baselines in recall (by 5.9 points) and F_1 score (by 1.6 points). Compared to BlueBERT, KIBERT has lower precision (3.4 points) but much higher recall (6.8 points) leading to a higher F_1 score. The dramatic increase in recall indicates that the KIBERT model is able to use the argument and type embeddings to detect more instances of correct relations than BlueBERT.

In order to determine the value of (a) the Relation-context Transformer Encoder Attention Masking strategy and (b) knowledge embeddings, we conduct an ablation analysis of KIBERT. Five knowledge embedding strategies are evaluated both with and without RTEAM. Specifically, we vary the knowledge embeddings in KIBERT representing types t_1, t_2, in the following ways:

1. **None**: The embeddings t_1, t_2 are not used. This is done to set a baseline from which to evaluate the efficacy if knowledge embeddings in KIBERT.

2. **Concept Embeddings**: The embeddings t_1, t_2 are replaced with knowledge embeddings of concepts from the UMLS Metathesaurus. In order to associate a concept

Table 6.5. Ablation analysis of KIBERT varying knowledge embedding and attention-masking strategies. Results measured in terms of F_1 score.

Knowledge Embedding Type	RTEAM masking	No Masking
None	76.5 (-1.5)	75.0 (-3.0)
Concept Embeddings	75.8 (-2.2)	74.4 (-3.6)
Semantic Type Embeddings	**78.0**	77.1 (-0.9)
Concept+Type Embeddings	76.2 (-1.8)	75.6 (-2.4)
Learned Type Embeddings	76.9 (-1.1)	75.9 (-2.1)

mention with a UMLS concept, we use the ScispaCy entity linker (Neumann et al., 2019). If no concept is found by ScispaCy, we default to the semantic type embedding defined in Section 6.2.2

3. **Semantic Type Embeddings**: This is the default KIBERT model. The embeddings t_1, t_2 are the semantic type embeddings defined in Section 6.2.2.

4. **Concept+Type Embeddings**: The embeddings t_1, t_2 are replaced with the Concept Embeddings and Semantic Type Embeddings described above concatenated together. In theory, such embeddings would contain both coarse-grained type information and fine-grained concept information.

5. **Learned Type Embeddings**: The embeddings t_1, t_2 are learned from scratch along with the rest of the parameters of KIBERT. This is done to evaluate the efficacy of knowledge embeddings by comparing them to embeddings with the same capacity learned from the data.

Results are presented for each knowledge embedding strategy both with and without RTEAM. The models trained without RTEAM use fully-connected self-attention, as is default in BERT and described in Section 3.4.1.

Table 6.5 presents this analysis evaluated using F_1 score. The analysis shows that both semantic type embeddings and RTEAM improve relation extraction performance of KIBERT. Without semantic type embeddings performance decreased by 1.5 points and without

RTEAM performance decreased by 0.9 points. Without both semantic type embeddings and RTEAM, performance degrades by 3.0 points. RTEAM improves results across the board, under any knowledge embedding strategy. This reinforces the results of Lee et al. (2019) and Peng et al. (2019) that focusing on sentence context is an effective strategy for relation extraction. However, the 76.5 F_1 score attained by the RTEAM model without knowledge embeddings shows that the RTEAM strategy is slightly preferable to ignoring concept mentions representing relation arguments entirely.

Table 6.5 reveals that semantic type embeddings are the preferred way to inject background knowledge into KIBERT for relation extraction between medical concepts in the 2010 i2b2/VA Challenge Dataset. Concept embeddings degrade performance below the baseline that does not use knowledge embeddings at all. Moreover, the combination of concept and type embeddings does not match the performance of type embeddings alone. We believe that this could be due to noisy entity linking misinforming the model during training. In order to properly evaluate the efficacy of UMLS concept embeddings for relation extraction, gold-standard concept codes are required for relation arguments. In future work, we plan to investigate strategies for *soft entity linking* whereby knowledge embeddings for the most likely concepts associated with a concept mention are combined using attention. The comparison with learned type embeddings shows that knowledge embeddings are indeed able to capture useful background knowledge and leverage that knowledge to inform relation extraction decisions.

6.4 Summary and Lessons Learned

In this chapter, the Knowledge-Informed BERT (KIBERT) model for relation extraction between medical concepts in clinical narratives is presented. KIBERT augments a state-of-the-art pre-trained Transformer model with (1) background knowledge; and (2) a novel attention masking strategy in order to improve results, setting a new baseline for the 2010

i2b2/VA Challenge Dataset. Background knowledge is incorporated in the form of knowledge embeddings learned from the knowledge graphs of the Unified Medical Language System (UMLS). Semantic Network embeddings from the UMLS are trained to capture the structure of the UMLS knowledge graphs, representing groups of medical concepts known as *semantic types*. KIBERT uses Semantic Type embeddings to represent the types of medical concepts participating in potential relations in discharge summaries. Moreover, a novel attention masking strategy is presented whereby the pre-trained Transformer model is used to simultaneously generate embeddings representing (a) simplified sentence context; and (b) fully-contextualized relation arguments. The results indicate that considering semantic type embeddings along with attention masking improves relation extraction performance.

This chapter demonstrates that deep learning methods for relation extraction are sufficiently robust operate well in a new genre of medical text while also showing that knowledge embeddings can be successfully used to inject knowledge when they take the form of semantic type embeddings. While semantic type embeddings are shown to improve results, such embeddings represent broad groups of concepts. More fine-grained knowledge in the form of *concept embeddings* is also available, but results indicate that more sophisticated methods are necessary to make use of this knowledge to improve relation extraction. Because relations were considered between arguments that represent broad groups of concepts from the UMLS, it is not surprising that semantic type embeddings are preferable for incorporating background (or ontological) knowledge. It is to be noted that because KIBERT takes advantage of knowledge embeddings, it achieves the best recall measures, indicating that it uncovers more relations than other methods. However, KIBERT also achieves lower precision than BlueBERT, indicating that the knowledge embeddings embolden an aggressive relation identification which, while less precise, leads to an improvement in overall performance.

CHAPTER 7

IDENTIFYING RELATIONS BETWEEN MEDICAL CONCEPTS IN STRUCTURED PRODUCT LABELS

In this chapter the methods introduced in this dissertation are extended to extract drug-drug interaction relations from medical reference text. Drug-drug interactions are a preventable cause of adverse events, the eighth leading cause of death in the United States (Demner-Fushman et al., 2018). For each prescription drug approved by the U.S. Food and Drug Administration (FDA), Structured Product Labeling (SPL) documents are produced, conveying in-depth information characterizing each prescription drug, including known drug-drug interactions. Contrary to their name, SPLs are comprised of unstructured natural language descriptions of the essential scientific information needed for the safe and effective use of a drug[1]. Together, the FDA and the National Library of Medicine (NLM) have a joint mandate to transform the natural language content of SPL documents, including drug-drug interactions, into a normalized, structured format that is readily accessible to downstream systems (Demner-Fushman et al., 2018). As such, Natural Language Processing (NLP) methods are necessary to extract information from the SPLs into a structured form. In an effort to address this problem, the 2019 Text Analysis Conference (TAC) Drug-Drug Interaction (DDI) Extraction from Drug Labels track was designed by the FDA and NLM for extraction of drug-drug interactions from SPL documents.

The TAC-DDI track provides a set of SPLs with manually annotated drug-drug interactions meant to facilitate the development of NLP systems that can automatically recognize such interactions. Unlike EHR data, SPLs are comprised of well-formed, grammatical sentences, however, they present their own difficulties. Each SPL document contains information pertaining to a single prescription drug, termed the *Labeled Drug*. Since the

[1]https://open.fda.gov/data/spl/

entire SPL pertains to the Labeled Drug, it is rarely if ever mentioned explicitly, yet each drug-drug interaction indicated in an SPL involves the Labeled Drug. Therefore, existing relation extraction methods that expect relation arguments to be explicitly mentioned must be adapted to the task of DDI extraction from SPLs. For example, consider the following sentence from the SPL for Accupril: "*Patients taking concomitant* [*mTOR inhibitor*] *(e.g.,* [*temisirolimus*]*) therapy may be at increased risk for* [*angioedema*]." The sentence indicates two DDIs between (Accupril and mTOR inhibitors) and (Accupril and temisirolimus). Both DDIs impose the risk of "*angioedema*". Each of the interacting substances ("*mTOR inhibitors*", "*temisirolimus*") and the effect of the interactions, ("*angioedema*") are represented in biomedical ontologies like those discussed in Chapter 5. The goal of the track is to identify the unique set of drug-drug interactions for each prescription drug and link the interactions to existing knowledge sources by normalizing the interacting substances and the effects of each interaction to target ontologies.

In order to facilitate the development of systems capable of performing this goal, the organizers provide four subtasks[2]: (1) entity recognition; (2) sentence-level relation identification; (3) entity normalization; and (4) normalized relation identification. The entity recognition task is to identify entities that participate in drug-drug interactions with the Labeled drug including interacting substances (e.g., "*mTOR inhibitors*") and interaction effects (e.g., "*angioedema*"). The sentence-level relation identification task is to classify the type of interactions, if any, between the set of entities identified in Task (1) in the same sentence. The entity normalization Task is to normalize each entity into a concept from one of three structured biomedical ontologies. The normalized relation identification task is the end goal of the 2019 TAC DDI track: to identify the unique interactions between normalized entities contained in an SPL. Task (4) is the goal of the challenge, while Tasks (1-3) are provided to aid in the development of machine learning systems capable of performing Task (4).

[2]https://bionlp.nlm.nih.gov/tac2019druginteractions

In this chapter, we present a multi-task neural network for end-to-end drug-drug interaction extraction from Structured Product Labels, the Multi-Task Transformer for Drug-Drug Interaction (MTTDDI). Moreover, we introduce a sentence classification task in which sentences that contain a DDI are discriminated from sentences that do not contain a DDI. MTTDDI jointly performs sentence classification, entity recognition (Task 1) and relation identification (Task 2) using a pre-trained Transformer model trained on medical text (Peng et al., 2019). In this way, MTTDDI jointly identifies sentences that contain a DDI, recognizes the arguments of all DDIs in the sentence, and classifies the relations between each set of arguments in the sentence. The entity normalization task is performed using tf-idf search and the normalized relation identification task is inferred from the results of Tasks 1-3. Similar to the Multi-task BERT models introduced in Chapters 2–3, MTTDDI uses a transformer model to develop a shared multi-task representation that is fed to a series of prediction models that identify DDIs and the entities that participate in them. We show that training MTTDDI to perform sentence classification along with entity recognition and relation identification improves upon equivalent models which are trained separately.

This chapter is organizes as follows: Section 7.1 describes the dataset and the task of Normalized Drug-Drug Interaction Identification in detail. Section 7.2 provides a brief background for the task and drug-drug interaction identification in general. Section 7.3 presents the MTTDDI model and the end-to-end normalized drug-drug interaction identification pipeline. Finally, Section 7.4 presents experimental results and Section 7.5 concludes the chapter.

7.1 Normalized Drug-Drug Interaction in Structured Product Labels: The 2019 TAC DDI Dataset

The 2019 TAC DDI Dataset targets three drug-drug interactions expressed in Structured Product Labels: (1) Pharmacodynamic Interactions, (2) Pharmacokinetic Interactions, and

(3) Unspecified Interactions. Pharmacodynamic Interactions (PDI) indicate that a drug interacts with the Labeled Drug[3] to produce an effect on a patient taking both drugs. Conversely, Pharmacokinetic Interactions (PKI) indicate that a drug interacts with the Labeled Drug to produce an effect on the drugs themselves. Unspecified Interactions (UI) indicate that a drug interacts with the Labeled Drug in some otherwise problematic way that is warned against. Each interaction involves the Labeled Drug and three potential arguments: a *precipitant*, a *trigger*, and an *effect*. A precipitant is a drug, drug class, or some other substance that causes one of the three DDIs described above when combined with the Labeled Drug. A trigger is a string of words that indicates the presence and type of an interaction. PDIs and PKIs also have *effects* that are annotated in the dataset. PDI effects are spans of text that indicate a medical problem that results from a PDI while PKI effects are classified into 20 discrete classes defined by the National Cancer Institute Thesaurus (Golbeck et al., 2003). Table 7.1 provides example sentences indicating DDIs along with their interaction type, precipitants, effects, and triggers.

The first sentence, S1, indicates two PDIs precipitated by the mentions "*mTOR inhibitor*" and "*temsirolimus*", indicated by the trigger "*increased risk*" and having the effect "*angioedema*". Sentence S2 demonstrates a PKI between the Labeled Drug Accupril and the precipitant "*high-fat meal*" indicated by the disjoint trigger span "*absorption|diminished*". Disjoint mention spans are common in this dataset, comprising 35% of trigger mentions and 14% of all mention spans. Note that the effect of the PKI demonstrated by S2 is not a span from the sentence, but the NCI Thesaurus code C54356 indicating a decrease in drug level. Sentence S3 demonstrates an unspecified interaction between the Labeled Drug and four precipitants with overlapping, disjoint spans. Note that UIs do not have annotated effects.

The first task of the 2019 TAC DDI challenge is the **entity recognition task**: to identify precipitant and effect spans mentioned in SPLs. The primary difficulty of this task arises

[3]The Labeled Drug is the drug for which the Structured Product Label was created.

Table 7.1. Examples of Drug-Drug Interactions from the 2019 TAC DDI Dataset. PDI indicates a pharmacodynamic interaction, PKI indicates a pharmacokinetic interaction, and UI indicates an unspecified interaction. Disjoint spans are indicated by the '|' character.

Sentence	Type	Precipitant	Effect	Trigger
S1: *Patients taking concomitant mTOR inhibitor (e.g., temsirolimus) therapy may be at increased risk for angioedema.*	PDI	*mTOR inhibitor*; *temsirolimus*	*angioedema*	*increased risk*
S2: *The rate and extent of quinapril absorption are diminished moderately when ACCUPRIL tablets are administered during a high-fat meal.*	PKI	*high-fat meal*	C54356	*absorption*| *diminished*
S3: *Combined P-gp and strong CYP3A4 inhibitors and inducers: Avoid concomitant use*	UI	*P-gp*|*inducers*; *P-gp*|*inhibitors*; *strong CYP3A4 inhibitors*; *strong CYP3A4*| *inducers*	–	*Avoid*

from the irregular spans like the precipitant spans demonstrated in S3. Moreover, the entity recognition task is further complicated by the fact that only those entities which participate in DDIs are to be identified. Task 2, the **sentence-level relation identification task**, is to classify the interaction type between each set of identified precipitants, triggers, and effects identified in Task 1. Trigger spans are not evaluated as part of the challenge for either Task 1 or 2, however they provide important information relevant to each task since they indicate the presence and type of a DDI.

The third task is the **normalization task**: to normalize each identified precipitant and effect into a target ontology. There are three target ontologies: Medication Reference Terminology (MED-RT)[4] and Unique Ingredient Identifier[5] for precipitants, and SNOMED-CT (Donnelly, 2006) for PDI effects. There are two classes of precipitants termed *drug classes* and *interacting substances* by the task organizers. Drug classes are precipitant men-

[4]https://www.nlm.nih.gov/research/umls/sourcereleasedocs/current/MED-RT/index.html

[5]https://www.fda.gov/industry/fda-resources-data-standards/fdas-global-substance-registration-system

Table 7.2. Dataset statistics for the 2019 TAC DDI Dataset.

	Training	**Validation**	**Testing**
Structured Product Labels	**191**	**20**	**86**
Entities (Task 1)	**17,533**	**2,314**	**5,713**
Precipitants	9,562	1,355	2,889
Effects	2,804	321	944
Triggers	5,167	638	1,880
Sentence-Level DDIs (Task 2)	**11,044**	**1,584**	**3,911**
Pharmacodynamic Interactions	4,938	547	2,188
Pharmacokinetic Interactions	3,255	485	1,010
Unspecified Interactions	2,851	552	713
Normalized Entities (Task 3)	**11,026**	**1,463**	**3,115**
Drug Classes (MED-RT)	4,935	918	1,308
Interacting Substances (UNII)	3,740	475	1,063
Effects (SNOMED-CT)	2,351	255	744
Unmapped	1,340	213	718
Normalized DDIs (Task 4)	**8,541**	**1,025**	**3,107**
Pharmacodynamic Interactions	4,195	456	1,763
Pharmacokinetic Interactions	2,309	308	780
Unspecified Interactions	2,036	261	564

tions corresponding to classes or groups of drugs (e.g., "*P-gp inducers*"), while interacting substances include specific drugs or non-drug substances that cause an interaction with the Labeled Drug (e.g., "*temsirolimus*", "*high-fat meal*"). Precipitants which are drug classes are to be normalized to MED-RT while interacting substances are to be normalized to UNII. Because PKI effects are not tied to mentions, they are not evaluated as part of Task 3. It should be noted that roughly 12% of precipitant and PDI effect mentions do not have a valid mapping in a target ontology. These mentions are assigned a normalized code of NO MAP.

Task four is the **Normalized DDI identification task**: to identify the unique DDIs between the labeled drug and normalized precipitants and effects identified in the SPL. For PDIs and UIs, Task 4 is entailed by the results of Tasks 1-3, however for PKIs, the PKI effect code must be identified as well.

Table 7.2 lists the frequencies of entities (Task 1), sentence-level DDIs (Task 2), normalized entities (Task3), and normalized DDIs (Task 4) in the 2019 TAC DDI Dataset.[6] The dataset is comprised of 297 human-annotated SPL documents split into sets of 211 for training and 86 for testing. We reserve 20 randomly selected training documents for validation.

7.2 Background

Drug-Drug Interaction (DDI) occurs when two drugs are co-administered causing the effects of at least one of the administered drugs to change. Adverse reactions to DDIs are a major preventable cause of injury and death in the United States (Demner-Fushman et al., 2018), so the automatic extraction of DDIs is an important research area in regards to patient outcomes. In 2011, the 1st DDIExtraction challenge task was held for the identification of DDIs in biomedical texts (Segura-Bedmar et al., 2011) and was updated in 2013 as part of SemEval (Segura Bedmar et al., 2013). The DDIExtraction dataset focuses on the identification of pharmacological substances and DDIs between them in biomedical text from DrugBank and MedLine. Neural methods have shown promise for these tasks including convolutional networks (Liu et al., 2016) and hierarchical RNNs (Zhang et al., 2018). More recently, pretrained language model-based networks have set a new state-of-the-art result (Peng et al., 2019).

DDI extraction from SPLs is not as well studied. In 2018, the FDA and NLM partnered to host the first TAC DDI challenge track (Demner-Fushman et al., 2018) which consisted of a set of 22 training SPL documents and two test sets of 57 and 66 labels, respectively. The 2018 challenge defined the normalized DDI identification task and its three subtasks, however given the relatively small amount of training data, systems were not able to produce

[6] https://bionlp.nlm.nih.gov/tac2019druginteractions

results consistent with DDI identification in other domains, achieving a high score of 11.83 F_1. The top performing system consisted of a BiLSTM-CRF informed with syntactic features for entity recognition, a Piecewise Attention-LSTM for relation extraction, and learning-to-rank for entity normalization (Demner-Fushman et al., 2018). In this chapter, we leverage a state-of-the-art neural architecture and additional training data to set a new baseline for normalized DDI identification in structured product labels.

7.3 Multi-task Learning for Normalized Drug-Drug Interaction Extraction from Structured Product Labels

In this section we present the End-to-end Normalized DDI Identification Pipeline (ENDIP) for performing all four TAC-DDI tasks. ENDIP uses the Multi-task Transformer network for identifying Drug-Drug Interactions (MTTDDI) to identify DDI relations, the substances that precipitate DDIs, the effects of the DDIs, and their triggers. MTTDDI is a multi-task network, based on BERT (Devlin et al., 2019) that uses the pretrained transformer to develop a shared multi-task representation that is fed to a series of four prediction modules: (1) the sentence classifier; (2) the mention boundary detector; (3) the relation extractor; and (4) the Pharmacokinetic Effect (PKE) classifier.

ENDIP is depicted in Figure 7.1. ENDIP first ingests the three available training datasets via a preprocessing module. This module performs annotation propagation as in Dandala et al. (2018) and two-stage tokenization. Next, the MTTDDI model is used to identify sentences containing DDIs, and extract their mentions, relations, and PKI effect codes. The mentions and relations are post-processed by the Postprocessing module resulting in predicted mention spans (for Task 1) and predicted relations (for Task 2). The mentions are normalized into ontology codes by the Normalization module for Task 3. Finally, the normalized mentions and predicted relations are unified and filtered by uniqueness to derive the unique normalized DDI relations for Task 4.

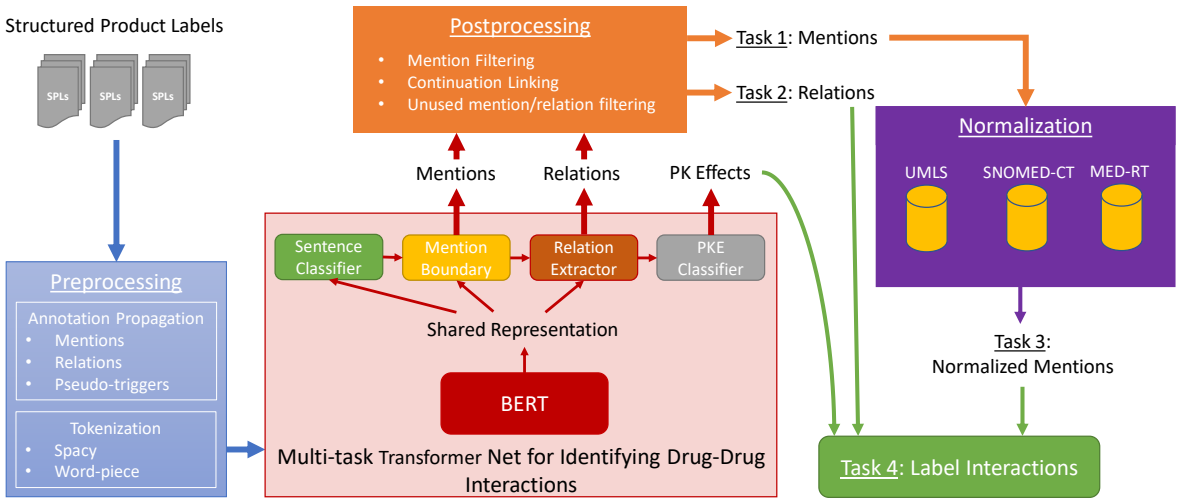

Figure 7.1. The End-to-end Normalized DDI Identification Pipeline (ENDIP).

7.3.1 Pre-processing

Pre-processing began with reading mentions and sentence-level interactions from SPL files, including triggers, precipitants, and effects. The presence of trigger annotations facilitates the representation of an N-ary DDI relations as a composition of binary relations. Interactions were represented as binary relations between triggers and precipitants and between triggers and effects. Therefore, ternary PDI relations are transformed into two binary relations, sharing the same trigger. In this way, the two ternary PDI relations in example S1 in Table 7.1 are expressed as three binary relations: {PDI(*increased risk*, *mTOR inhibitor*), PDI(*increased risk, temsirolimus*), PDI(*increased risk, angioedema*)}. The original ternary relations are recovered by joining precipitants and effects which are related to the same trigger in a relation of the same type. Since PKI and UI relations do not have effect mentions, they are simply represented as binary relations between triggers and precipitants. Pharmacokinetic interaction effects are treated as an attribute of the corresponding PKI relation since they had no mention associated with them.

The mention span annotations in the 2019 TAC DDI dataset are slightly below gold-standard due to the fact that they are produced semi-automatically via the following proce-

dure: mention spans indicating precipitants, effects, and triggers are identified in a sentence and then *only the first* substring matching the identified span is annotated as the gold mention span. This is problematic when the same span is mentioned more than once in the same sentence, e.g., "*Until data on possible interactions between verapamil and* $[disopyramide]_P$ *are obtained, disopyramide* $[should\ not\ be\ administered\ ...]_T$". In this example the second mention of *disopyramide* should be annotated as a precipitant since it is clearly related to the trigger span. Therefore, mention propagation is performed. Mention propagation consists of propagating the annotations of any mention participating in a relation to all other matching mention strings in the same sentence as in (Dandala et al., 2018). In the same way, relation propagation is performed to link newly created mentions spans to the same DDI arguments as the original span. During post processing, only the first predicted span is kept.

Sentence spans are provided by the task organizers. Tokenization was performed in two steps: at the word level, and at the word-piece level. First ScispaCy (Neumann et al., 2019) is used to extract word tokens from each sentence in a SPL. Then, word piece tokenization was performed on each token utilizing the word-piece vocabulary of BERT. We adopted the C-IOBES tagging scheme for mention prediction due to the prevalence of disjoint spans in this corpus. In C-IOBES tagging, each word-piece token is assigned a tag in {O,I,B,E,S,C-I,C-B,C-E-C-S} depending on if it outside of a mention, inside of a mention, the beginning of a mention or a single token representing a mention. The C- tags denote that a token is a part of a continuation span, e.g., "*diminished*" from the trigger "*absorption/diminished*" of S2 in Table 7.1. Continuation spans are attached to the closest leading head span of the same type. C-IOBES boundary tagging was performed on each sentence, where separate C-IOBES tags were assigned to each word piece token for triggers, precipitants, and effects. It should be noted that under this tagging scheme, there is some information loss in regards to overlapping spans of the same type. For instance, the four precipitants in example S2 from Table 7.1 are collapsed into a two spans: "*P-gp/strong CYP3A4 inhibitors*" and "*P-gp/inducers*". However, such overlapping spans account for less than 4% of the data.

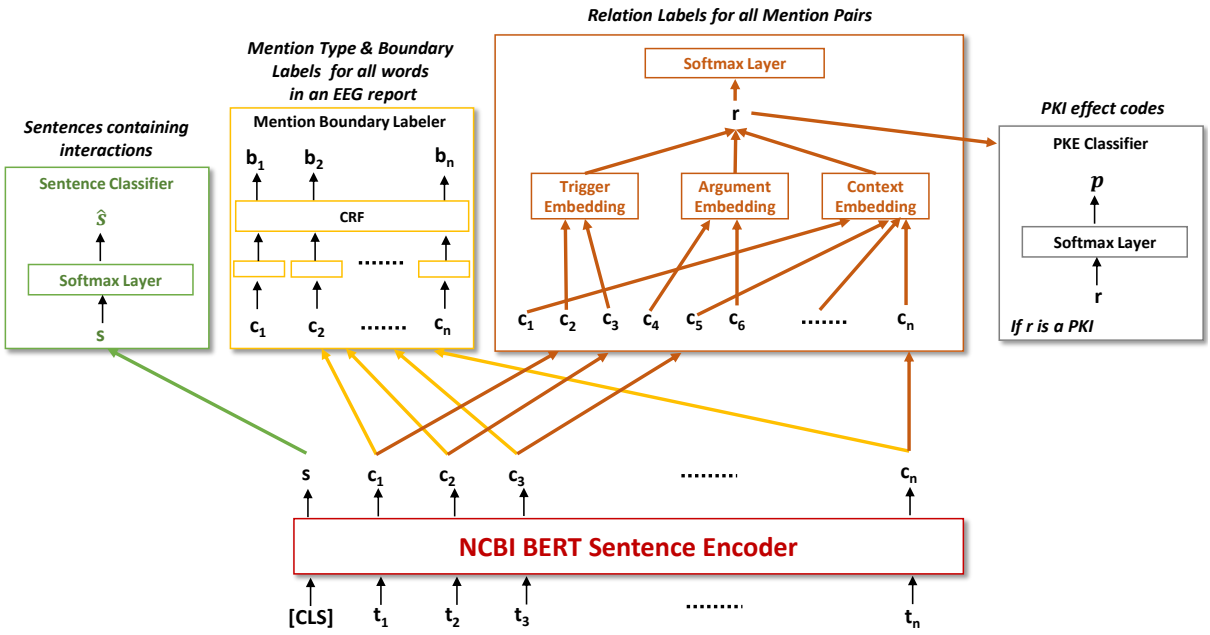

Figure 7.2. The Multi-Task Transformer Network for Identifying Drug-Drug Interactions.

7.3.2 The Multi-Task Transformer Network for Identifying Drug-Drug Interactions

The Multi-Task Transformer network for identifying Drug-Drug Interactions (MTTDDI), depicted in Figure 7.2, is a multi-task neural network built on the pretrained BERT transformer model to perform end-to-end drug-drug interaction identification from FDA drug labels. MTTDDI consists of five modules: (1) the BERT sentence encoder; (2) the sentence classifier; (3) the mention boundary labeler; (4) the relation extractor; and (5) the PKE classifier. MTTDDI operates on the sentence level, determining if the sentence contains a drug-drug interaction using the sentence classifier. If so, MTTDDI applies the mention boundary labeler to identify the arguments of the DDI and the relation extractor to classify the type of the relation. If the relation is found to be a pharmacokinetic interaction, the PKE classifier is applied to classify the effect of the interaction from one of the pre-defined classes in the NCI Thesaurus. Each module is trained jointly (including fine-tuning the

BERT model) using the following loss function:

$$\mathcal{L} = \mathcal{L}_S + \mathcal{L}_B + \mathcal{L}_R + \mathcal{L}_{PKE} \tag{7.1}$$

where $\mathcal{L}_S, \mathcal{L}_B, \mathcal{L}_R, \mathcal{L}_P$ are the loss functions of the sentence classifier, the mention boundary labeler, the relation extractor and the PKE classifier, respectively.

The BERT sentence encoder generates (1) a sentence embedding and (2) contextualized token embeddings for each token in the input sentence. The sentence embedding is passed to the sentence classifier while the contextualized token embeddings are passed to the mention boundary labeler and the relation extractor. The input to the BERT sentence encoder is a sequence of word-piece tokens with a special leading '[CLS]' token used for sentence classification as described in Devlin et al. (2019). As described in Section 3.4, a bi-directional transformer encoder such as BERT uses every token in the input sequence to inform the representation of every other token in the sequence using multi-headed attention. As such, the purpose of the '[CLS]' token is to extract from the full input sequence the information necessary to perform the sentence classification task, without biasing towards any specific token or learning to pool the contextualized token embeddings explicitly.

Formally, given a sequence of n word-piece tokens, $t_1, t_2, \ldots, t_n$ the BERT sentence encoder produces a sentence embedding, s, and a sequence of contextualized token embeddings, $c_1, c_2, \ldots, c_n$, as depicted in Figure 7.2. The sentence embedding s is fed to the sentence classifier while the contextualized token embeddings are shared among the mention boundary labeler and the relation extractor as in SACAR (Section 3.4).

The BERT model used in MTTDDI was pretrained in two phases, similar to the model described in Section 2.6.1, first on English Wikipedia and Books Corpus (Devlin et al., 2019) then on PubMed abstracts (Peng et al., 2019). This model was chosen due to the similarity of the language used in SPL medical reference documents and scientific articles contained in PubMed. The parameters of BERT are fine-tuned, updated during training using the

training signal provided by each of the four prediction models. In this way, the embeddings produced by BERT extract information relevant to each task from the input sentence.

Sentence Classifier

The sentence classifier module is used to determine if a given sentence contains a drug-drug interaction. While not one of the four tasks provided by TAC DDI, we found that filtering out sentences identified by the sentence classifier as not containing a DDI beneficial, experimentally. The sentence classifier module consists of a single sigmoid layer operating on the [CLS] embedding produced by the BERT sentence encoder:

$$\hat{s} = \sigma(W_S^\intercal s + b_S) \tag{7.2}$$

where σ is the sigmoid function, $W_S \in \mathbb{R}^d$ is a weight vector, and $b_S \in \mathbb{R}$ is a bias value, and $d = 768$ is the dimensionality of s, the hidden size of BERT. The sentence classifier is trained using sigmoid cross-entropy:

$$\mathcal{L}_\mathcal{S} = \sum_i y_i \log(\hat{s}) + (1 - y_i) \log(1 - \hat{s}) \tag{7.3}$$

Mention Boundary Labeler

The Mention Boundary Conditional Random Field (MB-CRF) uses a Conditional Random Field (CRF) (Lafferty et al., 2001) to generate the most likely C-IOBES boundary tag sequences for a sentence in order to identify trigger, precipitant and effect mentions in the sentence. In order to accomplish this, the MBD passes each contextualized token embedding c_i produced by BERT through a fully connected layer to produce a vector of potentials for each possible tag, $\tilde{b_i} \in \mathbb{R}^9$. This is identical to the MB-CRF module described in Section 2.6.4 with extra potentials to accommodate the larger tagset of this task. Three separate fully connected linear layers are trained to produce potentials for triggers, precipitants, and effects,

respectively, which are then fed to the same CRF:

$$\tilde{b_j^p} = W_p c_j + b_p \tag{7.4}$$

$$\tilde{b_j^e} = W_e c_j + b_e \tag{7.5}$$

$$\tilde{b_j^t} = W_t c_j + b_t \tag{7.6}$$

Again as in Section 2.6.4, the MB-CRF is trained to minimize the negative log likelihood of the correct tag sequence, given the sequence of potentials:

$$\mathcal{L}_B = \sum_i -\log P(b^p|\tilde{b_1^p}, \ldots \tilde{b_n^p}) - \log P(b^e|\tilde{b_1^e}, \ldots \tilde{b_n^e}) - \log P(b^t|\tilde{b_1^t}, \ldots \tilde{b_n^t}) \tag{7.7}$$

where b^p, b^e, and b_t are the true tag sequence for precipitants, effects, and triggers for sentence i, respectively.

Relation Extractor

The relation extractor is used to extract relations between each pair of mentions in the same sentence and to classify the type of relation between them, if any. Each potential relation is indicated by a trigger, therefore this task is cast as binary relation extraction between triggers and one other relation argument, either a precipitant or an effect. In order to predict a relation between a trigger and another argument, the model requires a representation for (a) the trigger, (b) the argument (either a precipitant or an effect), and (c) the context of the containing sentence. Therefore, the relation extractor distills the shared contextualized token embeddings provided by BERT into: (a) a trigger embedding; (b) an argument embedding; and (c) a context embedding for each trigger-argument pair. The trigger embedding represents the relation trigger, while the argument embedding represents either a precipitant or an effect, depending on what was is present in the sentence. The context embedding represents the context in which the two relation arguments occur – i.e., it is derived from the rest of the sentence. Formally, the trigger embedding is calculated

using max-pooling, $\tau = \phi(c_i^t, \ldots, c_j^t)$ where ϕ is the max-pooling operation and $c_i^t, \ldots, c_j^t$ are the tokens corresponding to the trigger mention. Similarly, the argument embedding is calculated as $\alpha = \phi(c_i^a, \ldots, c_j^a)$ where $c_i^a, \ldots, c_j^a$ are the tokens corresponding to the argument mention and the context embedding is calculated as $\delta = \phi(c_i^s, \ldots, c_j^s)$ where $c_i^s, \ldots, c_j^s$ are the tokens from the sentence that appear in neither argument.

The three embeddings are concatenated to derive the relation embedding representing the potential relation between trigger t and argument a: $r(t, a) = [\tau, \alpha, \delta]$. r is passed to a fully connected softmax layer and the relation extractor is trained using cross-entropy:

$$q(t, a)_k = \frac{e_k^{W_R r + b_R}}{\sum_l e_l^{W_R r + b_R}} \tag{7.8}$$

$$\mathcal{L}_R = -\sum_i \sum_{(t,a) \in s_i} \sum_k \mathrm{I}[y(t, a) = k] \log(q(t, a)_k) \tag{7.9}$$

where $q(t, a)$ is the probability distribution over relation types between trigger t and argument a, $W_R \in \mathbb{R}^{4 \times 3d}$ is a weight matrix, $b_R \in \mathbb{R}^4$ is a bias vector, and $y(t, a)$ is the true relation type between trigger t and argument a.

Pharmacokinetic Effect Classifier

The pharmacokinetic effect (PKE) classifier is used to predict the effect code of each pharmacokinetic interaction. The PKE classifier consists of a single softmax layer that operates on the relation embedding of the candidate relation, provided by the relation extractor. The PKE classifier is trained using softmax cross-entropy jointly along with the other four modules:

$$q^{pke}(t, a)_p = \frac{e_k^{W_P r + b_P}}{\sum_l e_l^{W_P r + b_P}} \tag{7.10}$$

$$\mathcal{L}_P = -\sum_i \sum_{(t,a) \in s_i} \mathrm{I}[y(t, a) = \mathrm{PKI}] \sum_p \mathrm{I}[y_p = p] \log(q^{pke}(t, a)_p) \tag{7.11}$$

where $q^{pke}(t, a)$ is the probability distribution over PKE types between trigger t and precipitant a, $W_P \in \mathbb{R}^{20 \times 3d}$ is a weight matrix, $b_P \in \mathbb{R}^4$ is a bias vector, $\mathrm{I}[y(t, a) = \mathrm{PKI}]$ indicates

that the loss is only considered for PKI relations, and y_e is the true PKE code for the PKI relation between trigger t and precipitant a.

7.3.3 Postprocessing

Postprocessing began with reading predictions from MTTDDI and cleaning up predicted C-IOBES tags by removing malformed predicted spans. Predicted spans were transformed into predicted mentions, where continuation spans were linked to the closest leading mention of the same type. First occurrences of mention text in the sentence were the only mentions kept, and interactions were reconstructed from predicted binary relations with the same head trigger. Predicted mentions which participated in no interactions were also removed.

7.3.4 Normalization

Each mention was normalized into one of three target ontologies: SNOMED-CT for PDI effects, MED-RT for drug classes, and UNII for interacting substances. Normalization was performed using string matching against atoms from (a) the vocabularies themselves and (b) the Unified Medical Language System (Lindberg et al., 1993). For each vocabulary, an index was constructed using the 2019AA UMLS release augmented with primary names from the source vocabularies themselves. For MED-RT, only drug classes and their atoms were extracted. Effects were searched against SNOMED and precipitants were searched against MED-RT first, then UNII if no match to MED-RT was found. In this way, the task of determining if a mention was a drug class was obviated.

7.4 Experimental Results and Discussions

ENDIP is evaluated against the submissions to the 2019 TAC DDI challenge in Tasks 1, 2, 3, and 4. Each task is evaluated using precision, recall, and F_1 score. In accordance with the standards set by the task organizers, Tasks 1 and 2 are evaluated using micro-averaged

statistics, while Tasks 3 and 4 are evaluated using statistics macro-averaged over each SPL in the test set.

The competing systems are summarized in the track overview (Goodwin et al., 2019) and described below:

1. **IBMResearch** (Mahajan et al., 2019) developed a series of BERT-based models targeting Tasks 1 and 2 using the cased 24-layer BERT-large model[7]. IBMResearch's submission is comprised of five separately trained BERT models for detection of (1) precipitants; (2) effects; (3) PDI triggers; (4) PKI triggers; and (5) UI triggers. Each of the five models perform boundary detection as token classification as in MTTDDI, however they do not use a CRF and train a separate model for each concept type. The effect boundary detector is only applied to sentences in which a precipitant has already been found. Task 1 is accomplished using the precipitant boundary detector and the effect boundary detector. Post-processing using a set of hand-crafted dependency parse patterns is used to extract overlapping disjoint entities, which is shown to be effective in this dataset. Task 2 is accomplished using the three typed trigger boundary detectors. If a trigger is detected in a sentence, the precipitants in the same sentence are predicted as participating in an interaction of the same type as the trigger. Precipitants and effects in the same sentence are linked together in a PDI. In this way, IBMResearch do not perform explicit relation detection. Final predictions for both tasks are made using an ensemble created via 5-fold cross validation. Pharmacokinetic effect codes are not predicted as Task 4 is not attempted by IBMResearch.

2. **SRCB** represents Ricoh Software Research Center (Beijing) (Ding et al., 2019) which have participated in Tasks 1, 2, and 3. For Tasks 1 and 2, SRCB develop a series

[7]https://github.com/google-research/bert

of BERT-based models using the cased BioBERT-large model[8]. For Task 1, SRCB trained models using additional universal transformer layers atop BioBERT (the number of layers is not specified) and perform data augmentation. Two models are trained for precipitants and effects, respectively. For data augmentation, SRCB used the full set of FDA SPLs downloaded from the web (including the labels present in the test set), use their current model to predict precipitants and effects in the unlabeled data, then retrain in the next epoch with the full dataset using both gold- and silver-standard labels. Prediction was performed using an ensemble created via 11-fold cross validation. Task 2 was performed by the SRCB system as *sentence-pair* classification. This sentence pair is comprised of (1) an input sentence from an SPL; and (2) a "*support sentence*" describing the type of interaction to be predicted, PDI, PKI, or UI. Support sentence creation was not described in detail (Ding et al., 2019). Task 3 was performed using index search as in ENDIP, however the search is augmented with multiple string kernels including Jaccard distance, Longest Common Subsequence, Levenshtein distance, and combinations thereof. Pharmacokinetic effect codes are not predicted as Task 4 is not attempted by SRCB.

3. **INK_BC**, a system developed by Shadong University of Finance and Economics, participated in Tasks 1 and 2 with "*a hybrid approach combining context and n-gram models*" as reported in Goodwin et al. (2019). No other details of this approach were provided.

The results for all Tasks are presented in Table 7.3 and compared against the top performing system for each task from the 2018 TAC DDI challenge. It should be noted that the 2018 TAC DDI challenge was conducted using a smaller training set and a different test set, so the results are not directly comparable.

[8]https://github.com/naver/biobert-pretrained

Table 7.3. Evaluation of ENDIP against participating systems in entity recognition (Task 1), relation extraction (Task 2), entity normalization (Task 3), and normalized relation identification (Task 4) of the 2019 TAC DDI measure using Precision, Recall, and F_1 score. Top scores in each task are bolded. The top scoring results from the 2018 challenge are also reported.

	Task 1			**Task 2**		
System	**Precision**	**Recall**	**F_1**	**Precision**	**Recall**	**F_1**
IBMResearch	**73.40**	**58.94**	**65.38**	**58.29**	**42.31**	**49.03**
SRCB	70.93	56.52	62.91	54.70	40.84	46.77
ENDIP	68.60	54.09	60.48	49.03	40.39	44.29
INK_BC	18.15	28.73	22.25	3.62	4.51	4.02
2018 Best	*43.8*	*49.9*	*46.7*	*54.4*	*32.8*	*40.9*
	Task 3			**Task 4**		
System	**Precision**	**Recall**	**F_1**	**Precision**	**Recall**	**F_1**
SRCB	**70.88**	**58.49**	**62.39** –	–	–	–
ENDIP	61.13	46.35	50.90	**31.89**	**28.63**	**28.84**
2018 Best	*31.9*	*24.0*	*26.4*	*17.4*	*9.7*	*11.8*

For Task 1, IBMResearch outperformed the other systems, attaining an F_1 score of 65.38 vs. 62.91 for SRCB and 60.48 for ENDIP. In Task 2, IBMResearch again outperform the other submissions, attaining an F_1 score of 49.03 vs. 46.77 for SRCB, and 44.29 for ENDIP. For Task 3, only SRCB and ENDIP participated, with SRCB submitting a more sophisticated string matching system than that of ENDIP. ENDIP is the lone system that is capable of performing Task 4. Compared with the top system from 2018, ENDIP achieves a 144% greater F_1 score (17.04 absolute), setting a new baseline for this domain.

The IBMReserch, SRCB, and ENDIP models are all based on BERT, with IBMResearch's model being the simplest, yet most effective adaptation. IBMResearch's model is comprised of a simple softmax layer atop BERT, while SRCB use universal transformer layers. However, both IBMResearch and SRCB use larger BERT models than ENDIP, indicating that the extra capacity of the larger models is useful for this task. Moreover, the pretrained BERT models used by each system differ; IBMResearch use the original BERT model (Devlin et al., 2019) trained on open-domain text while SRCB use BioBERT (Lee et al., 2019)

trained on scientific articles. In experiments using our validation set, we found that the original BERT model performed slightly worse than the NCBI BERT model used in ENDIP, indicating that the capacity of the model is more important than the pre-training corpus, for this challenge. The performance of IBMResearch over the other models indicates that the dependency pattern post-processing performed by IBMResearch is also important for Task 1. IBMResearch are the only team whose system is theoretically able to produce the correct precipitant spans for overlapping, non-contiguous mentions like the precipitants in example S3 in Table 7.1 due to their post-processing step. Future work may benefit from the incorporation of such dependency information at learning time, allowing models to learn their own patterns like those hand-crafted by IBMResearch.

Since the relation extraction of Task 2 is dependent on correctly identifying relation arguments in Task 1, the performance of Task 2 is bounded by the performance of Task 1. As such, a comparison of relation extraction methods is not possible for this task as the differences in Task 2 evaluation scores could be due entirely to differences in Task 1 performance. However, it should be noted that IBMResearch, SRCB, and the ENDIP systems perform similarly, indicating that the approaches are commensurate.

Both IBMResearch and SRCB train separate models for each of the three boundary detection problems in task 1, in contrast to the multi-task network of ENDIP. In order to assess the efficacy of the multi-task paradigm of the MTTDDI model used in ENDIP, we compare ENDIP against an alternate configuration whereby precipitant, trigger, and effect recognition are performed by dedicated models, as in the approaches of IBMResearch and SRCB. Specifically, we train three separate versions of the MTTDDI network that ignore both sentence and relation classification, focusing only on boundary detection. Each model is trained to focus on a single boundary detection task. We refer to this ENDIP configuration using multiple single-task learners for boundary recognition as ENDIP-ST (Single Task). ENDIP-ST is compared against the full multi-task version of ENDIP, referred to as ENDIP-MT (Multi-Task), in Table 7.4

Table 7.4. Evaluation of the multi-task paradigm of ENDIP on of the 2019 TAC DDI measure using Precision, Recall, and F_1 score. ENDIP-MT uses a single end-to-end mutli-task network, while ENDIP-ST is comprised of several single-task learners.

	Task 1			**Task 2**		
System	**Precision**	**Recall**	**F_1**	**Precision**	**Recall**	**F_1**
ENDIP-MT	**68.60**	54.09	**60.48**	**49.03**	40.39	44.29
ENDIP-ST	65.27	**55.94**	60.24	47.80	**43.19**	**45.38**
	Task 3			**Task 4**		
System	**Precision**	**Recall**	**F_1**	**Precision**	**Recall**	**F_1**
ENDIP-MT	**61.13**	46.35	**50.90**	**31.89**	**28.63**	**28.84**
ENDIP-ST	58.61	**47.53**	50.69	29.31	28.08	27.58

Table 7.4 shows that the adoption of separate single-task learners for boundary detection improves recall in Tasks 1-3 but decreases precision for all four tasks. For relation extraction, ENDIP-ST sets a new state-of-the art in recall at 43.19 (IBMResearch achieved 42.31). The increase in recall does not, however, translate to Task 4, where the multi-task model is still superior. Moreover, the multi-task model has superior precision in each task and superior F_1 score in Tasks 1, 2, and 4. This indicates that the multi-task paradigm is indeed useful for identifying normalized drug-drug interactions in medical reference text. The superiority of multi-task learning in terms of precision could be explained by the model using information from related tasks to identify and ignore potential false positives. By focusing on several aspects of the input sentence, the model is able to identify otherwise hidden features that preclude incorrect classification. Moreover, the fact that ENDIP-MT out-performs ENDIP-ST in Task 4 but not Task 2 indicates that the higher precision approach is preferable for the end-task of normalized relation identification.

7.5 Summary and Lessons Learned

In this chapter, an end-to-end pipeline for identifying normalized drug-drug interactions from medical reference text is presented. The End-to-end Normalized Drug-drug interaction

Identification Pipeline (ENDIP) leverages a multi-task neural model based on the BERT pre-trained Transformer model. ENDIP is evaluated using the 2019 TAC DDI dataset, which provides three sub-tasks in addition to normalized drug-drug interaction. The sub-tasks, entity recognition, sentence-level relation identification, and entity normalization, facilitate the training of models capable of performing normalized relation extraction. ENDIP sets a new baseline for normalized DDI extraction from Structured Product Labels, achieving results comparable to much larger state-of-the-art models on the entity recognition and sentence-level relation identification subtasks. Experiments show that learning to perform relation extraction along with entity recognition in a multi-task network improves results over models learned in isolation. Moreover, the chapter illustrates that deep learning methods for relation identification are robust across genres of medical text.

CHAPTER 8

POSSIBLE FUTURE DIRECTIONS

In this chapter, possible avenues for future work are presented. First, the work presented in this dissertation could be extended to yet another genre of medical text: scientific articles. Relation extraction from scientific articles has important applications including clinical decision support, knowledge discovery, and search (Nasar et al., 2018). In order for automatic systems to stay up-to-date with current research, knowledge must be continuously extracted from newly published articles. With respect to systems like SACAR (Section 3.4) and KIBERT (Section 6.2), relation extraction from scientific articles poses new problems which require extensions. For instance, an interesting problem in relation extraction from scientific articles is zero-shot learning of new relation types. Scientific articles differ widely in subject and scope, necessitating different relation schemata for each sub-genre (e.g., cancer research, quantum physics, natural language processing). Zero-shot learning methods could be investigated to apply learned models to new sub-genres of scientific articles inducing new target relations for each sub-genre.

Another promising area of future work is the unification of the MT-BGCN (Section 2.6) and KIBERT (Section 6.2) models to form a single multi-task information extraction architecture for discharge summaries. Recall from Section 6.2 that KIBERT extends the BlueBERT model. The BlueBERT model alters the input sentence, removing medical concept mentions which precludes the possibility of jointly predicting medical concept mention boundaries with the same network. KIBERT does not have this drawback, and therefore is able to be incorporated with MT-BGCN to form a single multi-task network capable of predicting concepts, assertions, and relations. Moreover, the inclusion of syntactic information from the dependency parse could prove to be beneficial for the task of relation extraction as has been shown in previous work (Rink et al., 2011).

A third possible direction of future work is the extension of the multi-task methods presented in this dissertation to biomedical entity disambiguation. Biomedical entity disambiguation is the task of assigning a medical concept mention in a health narrative to a canonical code from a structured vocabulary (e.g., the UMLS, ICD-10). For instance, the MedMentions dataset (Mohan and Li, 2019) contains medical concept mention annotations from over 4,000 PubMed abstracts with an associated UMLS Concept Unique Identifier (CUI) code for each concept mention. Likewise, several shared tasks have been conducted that include an entity disambiguation component (Suominen et al., 2013; Pradhan et al., 2014; Elhadad et al., 2015). Learning to perform entity disambiguation in the same network as concept detection and relation extraction could prove to be beneficial since the semantics of each task are deeply interconnected. Moreover, I hypothesize that UMLS knowledge embeddings could also be used to improve entity disambiguation in such a multi-task network as was shown to be the case for relation extraction in Chapter 6.

CHAPTER 9

CONCLUSIONS

Biomedical text such as clinical narratives and medical reference documents are rich with important medical information relevant throughout medical informatics. However, this information is largely contained in unstructured text, unable to be incorporated into downstream systems. In this dissertation, deep learning methods for extracting meaningful information in the form of structured relations were investigated across three genres of health narratives: EEG reports, discharge summaries, and FDA structured product labeling documents. EEG reports are generated by neurologists during an electroencephalogram to document the clinically relevant information gleaned during the examination and the clinical interpretation thereof. Discharge summaries are generated upon discharge from a hospital stay, documenting the important events and findings that characterize a patient's stay. Structured Product Labeling documents are created for general use by the United States Food and Drug Administration (FDA) for each prescription drug, conveying important information characterizing the drug. In each of these genres, deep learning methods were presented in this dissertation for identifying relations between medical concepts mentioned in unstructured narrative text. In order to identify relations between medical concepts, the medical concepts themselves must first be identified. To that end, medical concept detection methods were presented in Chapter 2.

Chapter 2 introduced the task of medical concept detection in EEG reports, defining five categories of important medical concepts mentioned in the reports: (1) EEG activities, (2) EEG events, (3) medical problems, (4) tests, and (5) treatments. A deep learning architecture for identifying each of these medical concepts in EEG reports was presented, showing promising results. Moreover, a schema defining 18 attributes that characterize medical concepts mentioned in EEG reports was presented. A separate deep learning architecture for classifying the attributes of each concept was also presented. Chapter 2 also addressed the

task of identifying medical concepts and the physician's *belief values* about those concepts in discharge summaries. The documenting physician's belief value, referred to as an *assertion*, characterizes a medical concept mention, e.g., determining if the concept is mentioned as having occurred, not occurred, possibly occurred, etc. In Chapter 2, a state-of-the-art neural language model was adapted to jointly perform concept detection and assertion classification, leading to state-of-the-art results in both tasks.

Having presented methods for medical concept detection in Chapter 2, Chapter 3 addressed the task of extracting relations between the identified concepts. A novel relation schema was defined including four relation types. Due to the nature of the EEG report, relations between medical concepts often span sentences and even sections. Therefore in Chapter 3, two neural networks performing long-distance relation extraction were presented. The first network, EEG-RelNet, processes entire EEG reports one sentence at a time, gathering information about potential relations between concepts using a set of *memory* vectors. Upon processing the entire document, these memory vectors are used to predict relations between each pair of concepts mentioned anywhere in the report. While experimental results are promising, EEG-RelNet requires medical concepts and their attributes to be identified a priori, which is inefficient. The second neural architecture presented in Chapter 3 is an end-to-end multi-task network that jointly predicts medical concepts, their attributes, and relations between them in the same network.

While the neural methods presented in Chapters 2 and 3 are performant, they require large amounts of labeled data in order to be trained. In order to produce this labeled data, manual annotations were required. Due to the highly specialized narratives of EEG reports, substantial expertise is required to generate these annotations. Therefore, in Chapter 4 Active Learning (AL) methods were investigated. Three Active Learning frameworks were developed for training the neural methods for information extraction in EEG reports presented in Chapters 2 and 3. The first framework targeted the annotation of concepts and

attributes, while the second framework targeted relations. Both of these AL frameworks use traditional heuristics to select documents for manual annotation from a pool of unlabeled documents. These heuristics are referred to as the active learning *selection policy*. The third framework used the multi-task model presented in Chapter 3 to address two drawbacks of the first two AL frameworks: (1) the inefficiency of having separate frameworks operating on the same unlabeled document pool; and (2) the ineffectiveness of traditional heuristics selecting new documents for multiple annotation tasks. The third framework selects unlabeled documents that are informative for concept detection, attribute classification, and relation extraction, addressing drawback (1). To accomplish this, the third AL framework learns an active learning selection policy from data, addressing drawback (2).

Chapter 5 addressed the task of biomedical knowledge graph embedding. Knowledge graph embedding methods were presented for performing inference using relations extracted from EEG reports. Moreover, Chapter 5 presents a novel approach for creating knowledge embeddings from the expert-curated knowledge graph defined by the Unified Medical Language System (UMLS) (Lindberg et al., 1993). The UMLS is a large biomedical ontology that unifies disparate biomedical vocabularies, defining two comprehensive knowledge graphs: the Metathesaurus and the Semantic Network. The Metathesaurus defines relations between medical concepts while the Semantic Network defines relations between Semantic Types. Semantic Types are conceptual groups of medical concepts, e.g., diagnostic procedure, clinical drug, and disease or syndrome. Methods for embedding both the Metathesaurus and Semantic Network knowledge graphs into the same embedding space are presented, resulting in knowledge embeddings for medical concepts and semantic types encoded in the UMLS. Because deep learning architectures cannot make direct use of structured ontological knowledge, UMLS knowledge embeddings provide an intriguing method for representing biomedical concepts in downstream deep learning architectures. In Chapter 5, the UMLS knowledge embeddings were applied to an existing clinical prediction model, improving results. Moreover, the knowledge embedding method presented in Chapter 5 was extended to

perform ontology alignment, attaining promising results. In Chapter 6, UMLS knowledge embeddings learned in Chapter 5 are applied to the task of relation extraction in clinical narratives.

Chapter 6 presented a neural network that used semantic type embeddings generated in Chapter 5 to improve relation extraction performance in discharge summaries. Semantic type embeddings were used to represent the type of each medical concept participating in a potential relation, providing valuable background knowledge to inform relation classification decisions. Experimental results showed that knowledge embeddings result in drastically improved recall compared with the state-of-the-art, indicating that the embeddings allowed the model to recognize more relations than previous methods.

In Chapter 7, relation extraction from FDA Structured Product Labeling (SPL) documents was presented. Three types of Drug-Drug Interaction relations were identified between the drug the SPL is about (i.e., the *Labeled Drug*), an interacting substance, and a resulting effect. An end-to-end multi-task neural pipeline was developed to identify medical concept mentions, DDI relations between them, and normalize the DDIs into one of four target ontologies, setting a new baseline for normalized DDI extraction from Structured Product Labels. Experimental results indicate that multi-task learning improves results on the end-task of normalized DDI extraction over equivalent single-task models trained separately.

Overall, this dissertation presents novel deep learning architectures for relation extraction across three genres of health narratives, setting new baselines in each genre. Novel methods for representing medical knowledge as dense vectors readily accessible to deep learning systems were presented and applied to relation extraction in discharge summaries, advancing the state-of-the-art. Possible directions for future work include merging the tasks of information extraction and knowledge embedding into the same, multi-task network. Currently, state of the art text encoders leverage vast amounts of text data, but are not directly informed by knowledge. A network capable of producing a knowledge-informed text encoding could prove to be the next frontier in deep text representation.

REFERENCES

Abadi, M., P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. (2016). Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pp. 265–283.

Achichi, M., M. Cheatham, Z. Dragisic, J. Euzenat, D. Faria, A. Ferrara, G. Flouris, I. Fundulaki, I. Harrow, V. Ivanova, et al. (2017). Results of the ontology alignment evaluation initiative 2017. In *OM: Ontology Matching*, pp. 61–113.

Aronson, A. R. (2001). Effective mapping of biomedical text to the umls metathesaurus: the metamap program. In *Proceedings of the AMIA Symposium*, pp. 17. American Medical Informatics Association.

Ashburner, M., C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, et al. (2000). Gene ontology: tool for the unification of biology. *Nature genetics 25*(1), 25–29.

Ba, J. L., J. R. Kiros, and G. E. Hinton (2016). Layer normalization. In *arXiv preprint arXiv:1607.06450*.

Bachman, P., A. Sordoni, and A. Trischler (2017). Learning algorithms for active learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 301–310. JMLR. org.

Bansal, T., A. Neelakantan, and A. McCallum (2017). Relnet: End-to-end modeling of entities & relations. In *NIPS Workshop on Automated Knowledge Base Construction (AKBC)*.

Bengio, Y. and Y. LeCun (Eds.) (2015). *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Bengio, Y., P. Simard, P. Frasconi, et al. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks 5*(2), 157–166.

Beniczky, S., L. J. Hirsch, P. W. Kaplan, R. Pressler, G. Bauer, H. Aurlien, J. C. Brøgger, and E. Trinka (2013). Unified eeg terminology and criteria for nonconvulsive status epilepticus. *Epilepsia 54*, 28–29.

Bhatia, P., B. Celikkaya, and M. Khalilia (2019). Joint entity extraction and assertion detection for clinical text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 954–959.

Bordes, A., S. Chopra, and J. Weston (2014). Question answering with subgraph embeddings. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 615–620.

Bordes, A., N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko (2013). Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pp. 2787–2795.

Bordes, A., J. Weston, R. Collobert, and Y. Bengio (2011). Learning structured embeddings of knowledge bases. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*.

Brown, P. F., P. V. Desouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai (1992). Class-based n-gram models of natural language. *Computational linguistics 18*(4), 467–479.

Cai, L. and W. Y. Wang (2018). Kbgan: Adversarial learning for knowledge graph embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, Volume 1, pp. 1470–1480.

Chapman, W. W., W. Bridewell, P. Hanbury, G. F. Cooper, and B. G. Buchanan (2001). A simple algorithm for identifying negated findings and diseases in discharge summaries. *Journal of biomedical informatics 34*(5), 301–310.

Chapman, W. W., P. M. Nadkarni, L. Hirschman, L. W. D'Avolio, G. K. Savova, and O. Uzuner (2011). Overcoming barriers to nlp for clinical text: the role of shared tasks and the need for additional creative solutions.

Chen, D. and C. Manning (2014). A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 740–750.

Chen, Y., R. J. Carroll, E. R. M. Hinz, A. Shah, A. E. Eyler, J. C. Denny, and H. Xu (2013). Applying active learning to high-throughput phenotyping algorithms for electronic health records data. *Journal of the American Medical Informatics Association 20*(e2), e253–e259.

Cho, K., B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio (2014). Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734.

Choi, E., M. T. Bahadori, J. A. Kulas, A. Schuetz, W. F. Stewart, and J. Sun (2016). Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, USA, pp. 3512–3520. Curran Associates Inc.

Cimino, J. J. (1998). Desiderata for controlled medical vocabularies in the twenty-first century. *Methods of information in medicine 37*(4-5), 394.

Cui, L., A. Bozorgi, S. Lhatoo, G.-Q. Zhang, and S. S Sahoo (2012, 11). Epidea: Extracting structured epilepsy and seizure information from patient discharge summaries for cohort identification. *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium 2012*, 1191–200.

Cui, L., S. S. Sahoo, S. D. Lhatoo, G. Garg, P. Rai, A. Bozorgi, and G.-Q. Zhang (2014). Complex epilepsy phenotype extraction from narrative clinical discharge summaries. *Journal of biomedical informatics 51*, 272–279.

Dai, Z., Z. Yang, Y. Yang, J. Carbonell, Q. Le, and R. Salakhutdinov (2019, July). Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, pp. 2978–2988. Association for Computational Linguistics.

Dandala, B., D. Mahajan, and A. Poddar (2018). Ibm research system at tac 2018: Deep learning architectures for drug-drug interaction extraction from structured product labels. In *Text Analysis Conference*.

Das, R., A. Neelakantan, D. Belanger, and A. McCallum (2017). Chains of reasoning over entities, relations, and text using recurrent neural networks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, Volume 1, pp. 132–141.

De Bruijn, B., C. Cherry, S. Kiritchenko, J. Martin, and X. Zhu (2011). Machine-learned solutions for three stages of clinical information extraction: the state of the art at i2b2 2010. *Journal of the American Medical Informatics Association 18*(5), 557–562.

Dehghani, M., S. Gouws, O. Vinyals, J. Uszkoreit, and Ł. Kaiser (2018). Universal transformers. In *6th International Conference on Learning Representations, ICLR 2018, Conference Track Proceedings*.

Demner-Fushman, D., W. W. Chapman, and C. J. McDonald (2009). What can natural language processing do for clinical decision support? *Journal of biomedical informatics 42*(5), 760–772.

Demner-Fushman, D., K. W. Fung, P. Do, R. D. Boyce, and T. R. Goodwin (2018). Overview of the tac 2018 drug-drug interaction extraction from drug labels track. In *proceedings of the Text Analysis Conference (TAC 2018)*.

Demner-Fushman, D., W. J. Rogers, and A. R. Aronson (2017). Metamap lite: an evaluation of a new java implementation of metamap. *Journal of the American Medical Informatics Association 24*(4), 841–844.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186.

Ding, L., L. Liang, Y. Tong, S. Jiang, and B. Dong (2019). A bert-based model for drug-drug interaction extraction from drug labels. In *Text Analysis Conference*.

Dligach, D., T. Miller, and G. Savova (2013, September). Active learning for phenotyping tasks. In *Proceedings of the Workshop on NLP for Medicine and Biology associated with RANLP 2013*, Hissar, Bulgaria, pp. 1–8. INCOMA Ltd. Shoumen, BULGARIA.

Donnelly, K. (2006). Snomed-ct: The advanced terminology and coding system for ehealth. *Studies in health technology and informatics 121*, 279.

D'Souza, J. and V. Ng (2014). Ensemble-based medical relation classification. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pp. 1682–1693.

Elhadad, N., S. Pradhan, S. Gorman, S. Manandhar, W. Chapman, and G. Savova (2015). Semeval-2015 task 14: Analysis of clinical text. In *proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pp. 303–310.

Faria, D., C. Pesquita, E. Santos, M. Palmonari, I. F. Cruz, and F. M. Couto (2013). The agreementmakerlight ontology matching system. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pp. 527–541. Springer.

Fraser, K. C., I. Nejadgholi, B. De Bruijn, M. Li, A. LaPlante, and K. Z. E. Abidine (2019). Extracting umls concepts from medical text using general and domain-specific deep learning models. In *arXiv preprint arXiv:1910.01274*.

Friedman, C. (1997). Towards a comprehensive medical language processing system: methods and issues. In *Proceedings of the AMIA annual fall symposium*, pp. 595. American Medical Informatics Association.

García-Durán, A., A. Bordes, N. Usunier, and Y. Grandvalet (2015). Combining two and three-way embeddings models for link prediction in knowledge bases. *CoRR abs/1506.00999*, 1.

Glorot, X., A. Bordes, and Y. Bengio (2011). Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 315–323.

Golbeck, J., G. Fragoso, F. Hartel, J. Hendler, J. Oberthaler, and B. Parsia (2003). The national cancer instituteâĂŹs thésaurus and ontology. *Web Semantics: Science, Services and Agents on the World Wide Web 1*(1), 75–80.

Goodfellow, I. J., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio (2014). Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, Cambridge, MA, USA, pp. 2672–2680. MIT Press.

Goodwin, T. and S. M. Harabagiu (2013). Graphical induction of qualified medical knowledge. *International Journal of Semantic Computing 7*(04), 377–405.

Goodwin, T. and S. M. Harabagiu (2016). Multi-modal patient cohort identification from eeg report and signal data. In *AMIA Annual Symposium Proceedings*, Volume 2016, pp. 1794–1803.

Goodwin, T. R. (2018). *Medical Question Answering and Patient Cohort Retrieval*. Ph. D. thesis, The University of Texas at Dallas.

Goodwin, T. R., D. Demner-Fushman, K. W. Fung, and P. Do (2019). Overview of the tac 2019 track on drug-drug interaction extraction from drug labels. In *Text Analysis Conference*.

Goodwin, T. R. and S. M. Harabagiu (2016). Medical question answering for clinical decision support. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pp. 297–306. ACM.

Graves, A. (2016). Adaptive computation time for recurrent neural networks. In *arXiv preprint arXiv:1603.08983*.

Guo, S., Q. Wang, B. Wang, L. Wang, and L. Guo (2015). Semantically smooth knowledge graph embedding. In *ACL (1)*, pp. 84–94.

Guo, Z., Y. Zhang, and W. Lu (2019). Attention guided graph convolutional networks for relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 241–251.

Hahn, U., E. Beisswanger, E. Buyko, and E. Faessler (310). Active learning-based corpus annotation–the pathojen experience. In *AMIA Annual Symposium Proceedings*, Volume 2012, pp. 301. American Medical Informatics Association.

He, B., Y. Guan, and R. Dai (2019). Classifying medical relations in clinical text via convolutional neural networks. *Artificial intelligence in medicine 93*, 43–49.

He, K., X. Zhang, S. Ren, and J. Sun (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034.

He, K., X. Zhang, S. Ren, and J. Sun (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778.

Henaff, M., J. Weston, A. Szlam, A. Bordes, and Y. LeCun (2016). Tracking the world state with recurrent entity networks. In *arXiv preprint arXiv:1612.03969*.

Hendrycks, D. and K. Gimpel (2016). Bridging nonlinearities and stochastic regularizers with gaussian error linear units. In *International Conference on Learning Representations*.

Hersh, W. (2008). *Information retrieval: a health and biomedical perspective*. Springer Science & Business Media.

Hochreiter, S. and J. Schmidhuber (1997). Long short-term memory. *Neural computation 9*(8), 1735–1780.

Huang, D. W., B. T. Sherman, and R. A. Lempicki (2009). Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. *Nucleic acids research 37*(1), 1–13.

Huang, Z., W. Xu, and K. Yu (2015). Bidirectional lstm-crf models for sequence tagging. In *arXiv preprint arXiv:1508.01991*.

Ikhwantri, F., S. Louvan, K. Kurniawan, B. Abisena, V. Rachman, A. F. Wicaksono, and R. Mahendra (2018). Multi-task active learning for neural semantic role labeling on low resource conversational corpus. In *Proceedings of the Workshop on Deep Learning Approaches for Low-Resource NLP*, pp. 43–50.

Jameel, S., Z. Bouraoui, and S. Schockaert (2017). Member: Max-margin based embeddings for entity retrieval. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, New York, NY, USA, pp. 783–792. ACM.

Ji, G., S. He, L. Xu, K. Liu, and J. Zhao (2015a). Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Volume 1, pp. 687–696.

Ji, G., S. He, L. Xu, K. Liu, and J. Zhao (2015b). Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Volume 1, pp. 687–696.

Ji, Z., Q. Wei, A. Franklin, T. Cohen, and H. Xu (838). Cost-sensitive active learning for phenotyping of electronic health records. In *2019 AMIA Informatics Summit Proceedings*, Volume 2019, pp. 829. American Medical Informatics Association.

Kibbe, W. A., C. Arze, V. Felix, E. Mitraka, E. Bolton, G. Fu, C. J. Mungall, J. X. Binder, J. Malone, D. Vasant, et al. (2014). Disease ontology 2015 update: an expanded and updated database of human diseases for linking biomedical knowledge through disease data. *Nucleic acids research 43*(D1), D1071–D1078.

Kind, A. J. and M. A. Smith (2008). Documentation of mandated discharge summary components in transitions from acute to subacute care. In *Advances in patient safety: New directions and alternative approaches (vol. 2: Culture and redesign)*. Agency for Healthcare Research and Quality (US).

Kingma, D. P. and J. Ba (2015). Adam: A method for stochastic optimization. See (Bengio and LeCun, 2015).

Konyushkova, K., R. Sznitman, and P. Fua (2017). Learning active learning from data. In *Advances in Neural Information Processing Systems*, pp. 4225–4235.

Lafferty, J. D., A. McCallum, and F. C. Pereira (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pp. 282–289. Morgan Kaufmann Publishers Inc.

Le, Q. and T. Mikolov (2014). Distributed representations of sentences and documents. In *International conference on machine learning*, pp. 1188–1196.

LeCun, Y., Y. Bengio, and G. Hinton (2015). Deep learning. *nature 521*(7553), 436–444.

Lee, J., W. Yoon, S. Kim, D. Kim, C. So, and J. Kang (2019). Biobert: a pre-trained biomedical language representation model for biomedical text mining. In *Bioinformatics (Oxford, England)*.

LePendu, P., M. A. Musen, and N. H. Shah (2011). Enabling enrichment analysis with the human disease ontology. *Journal of biomedical informatics 44*, S31–S38.

Levandowsky, M. and D. Winter (1971). Distance between sets. *Nature 234*(5323), 34.

Lewis, D. D. and W. A. Gale (1994). A sequential algorithm for training text classifiers. In *SIGIRâĂŹ94*, pp. 3–12. Springer.

Li, P., Z. Yuan, W. Tu, K. Yu, and D. Lu (2019). Medical knowledge extraction and analysis from electronic medical records using deep learning. *Chinese Medical Sciences Journal 34*(2), 133–139.

Lin, Y., Z. Liu, M. Sun, Y. Liu, and X. Zhu (2015). Learning entity and relation embeddings for knowledge graph completion. In *AAAI*, pp. 2181–2187.

Lin, Z., M. Feng, C. N. dos Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio (2017). A structured self-attentive sentence embedding. In *5th International Conference on Learning Representations, ICLR 2017, Conference Track Proceedings*.

Lindberg, D. A., B. L. Humphreys, and A. T. McCray (1993). The unified medical language system. *Yearbook of Medical Informatics 2*(01), 41–51.

Lipscomb, C. E. (2000). Medical subject headings (mesh). *Bulletin of the Medical Library Association 88*(3), 265.

Liu, M., W. Buntine, and G. Haffari (2018a). Learning how to actively learn: A deep imitation learning approach. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1874–1883.

Liu, M., W. Buntine, and G. Haffari (2018b). Learning how to actively learn: A deep imitation learning approach. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1874–1883. Association for Computational Linguistics.

Liu, S., B. Tang, Q. Chen, and X. Wang (2016). Drug-drug interaction extraction via convolutional neural networks. In *Computational and mathematical methods in medicine*, Volume 2016. Hindawi.

Luo, Y. (2017). Recurrent neural networks for classifying relations in clinical notes. *Journal of Biomedical Informatics 72*, 85–95.

Luo, Y., Y. Cheng, Ö. Uzuner, P. Szolovits, and J. Starren (2017). Segment convolutional neural networks (seg-cnns) for classifying relations in clinical notes. *Journal of the American Medical Informatics Association 25*(1), 93–98.

Luo, Y., Ö. Uzuner, and P. Szolovits (2016). Bridging semantics and syntax with graph algorithmsâĂŤstate-of-the-art of extracting biomedical relations. *Briefings in bioinformatics 18*(1), 160–178.

Mahajan, D., A. Poddar, and L. Yen-Ting (2019). A hybrid model for drug-drug interaction extraction from structured product labeling documents. In *Text Analysis Conference*.

Maldonado, R., T. R. Goodwin, and S. M. Harabagiu (2017). Active deep learning-based annotation of electroencephalography reports for cohort identification. *AMIA Summits on Translational Science Proceedings 2017*, 229.

Maldonado, R., T. R. Goodwin, and S. M. Harabagiu (2018). Memory-augmented active deep learning for identifying relations between distant medical concepts in electroencephalography reports. *AMIA Summits on Translational Science Proceedings 2018*, 156.

Maldonado, R., T. R. Goodwin, M. A. Skinner, and S. M. Harabagiu (2017). Deep learning meets biomedical ontologies: knowledge embeddings for epilepsy. In *AMIA Annual Symposium Proceedings*, Volume 2017, pp. 1233. American Medical Informatics Association.

Maldonado, R. and S. M. Harabagiu (2019). Bootstrapping adversarial learning of biomedical ontology alignments. In *AMIA Annual Symposium Proceedings*, Volume 2019, pp. 1. American Medical Informatics Association.

Maldonado, R., M. Yetisgen, and S. M. Harabagiu (2019). Adversarial learning of knowledge embeddings for the unified medical language system. In *AMIA Informatics Summit Proceedings*, Volume 2019. American Medical Informatics Association.

Marcheggiani, D. and I. Titov (2017). Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1506–1515.

Mikolov, T., I. Sutskever, K. Chen, G. Corrado, and J. Dean (2013). Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, USA, pp. 3111–3119. Curran Associates Inc.

Miller, A., A. Fisch, J. Dodge, A.-H. Karimi, A. Bordes, and J. Weston (2016). Key-value memory networks for directly reading documents. In *arXiv preprint arXiv:1606.03126*.

Miotto, R., L. Li, B. A. Kidd, and J. T. Dudley (2016). Deep patient: an unsupervised representation to predict the future of patients from the electronic health records. *Scientific reports 6*, 26094.

Mohan, S. and D. Li (2019). Medmentions: A large biomdedical corpus annotated with umls concepts. In *Proceedings of the 2019 Conference on Automated Knowledge Base Construction*, AKBC 2019. AKBC.

Mungall, C. J., C. Torniai, G. V. Gkoutos, S. E. Lewis, and M. A. Haendel (2012). Uberon, an integrative multi-species anatomy ontology. *Genome biology 13*(1), R5.

Nair, V. and G. E. Hinton (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814.

Nasar, Z., S. W. Jaffry, and M. K. Malik (2018). Information extraction from scientific articles: a survey. *Scientometrics 117*(3), 1931–1990.

Neumann, M., D. King, I. Beltagy, and W. Ammar (2019). Scispacy: Fast and robust models for biomedical natural language processing. In *arXiv:1902.07669*.

Nickel, M., V. Tresp, and H.-P. Kriegel (2011). A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pp. 809–816. Omnipress.

Noachtar, S., C. Binnie, J. Ebersole, F. Mauguiere, A. Sakamoto, and B. Westmoreland (1999). A glossary of terms most commonly used by clinical electroencephalographers and proposal for the report form for the eeg findings. the international federation of clinical neurophysiology. *Electroencephalography and clinical neurophysiology. Supplement 52*, 21.

Noachtar, S., C. Binnie, J. Ebersole, F. Mauguière, A. Sakamoto, and B. Westmoreland (2004). A glossary of terms most commonly used by clinical electroencephalographers and proposal for the report form for the eeg findings. *Klinische Neurophysiologie 35*(1), 5–21.

Okazaki, N. (2007). Crfsuite: a fast implementation of conditional random fields (crfs).

Onder, G., C. Pedone, F. Landi, M. Cesari, C. Della Vedova, R. Bernabei, and G. Gambassi (2002). Adverse drug reactions as cause of hospital admissions: results from the italian group of pharmacoepidemiology in the elderly (gifa). *Journal of the American Geriatrics Society 50*(12), 1962–1968.

Paciorkowski, A. R., L. L. Thio, and W. B. Dobyns (2011). Genetic and biologic classification of infantile spasms. *Pediatric neurology 45*(6), 355–367.

Parikh, A., O. Täckström, D. Das, and J. Uszkoreit (2016, November). A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas, pp. 2249–2255. Association for Computational Linguistics.

Paulus, R., C. Xiong, and R. Socher (2018). A deep reinforced model for abstractive summarization. In *International Conference on Learning Representations*.

Peng, Y., S. Yan, and Z. Lu (2019). Transfer learning in biomedical natural language processing: An evaluation of bert and elmo on ten benchmarking datasets. In *Proceedings of the BioNLP 2018 workshop*, pp. 58–65.

Pennington, J., R. Socher, and C. Manning (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543.

Peters, M. E., M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer (2018). Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pp. 2227–2237.

Pradhan, S., W. Chapman, S. Man, and G. Savova (2014). Semeval-2014 task 7: Analysis of clinical text. In *Proc. of the 8th International Workshop on Semantic Evaluation (SemEval 2014*. Citeseer.

Radford, A., J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever (2019). Language models are unsupervised multitask learners. In *OpenAI Blog*.

Rajkomar, A., E. Oren, K. Chen, A. M. Dai, N. Hajaj, M. Hardt, P. J. Liu, X. Liu, J. Marcus, M. Sun, P. Sundberg, H. Yee, K. Zhang, Y. Zhang, G. Flores, G. E. Duggan, J. Irvine, Q. Le, K. Litsch, A. Mossin, J. Tansuwan, D. Wang, J. Wexler, J. Wilson, D. Ludwig, S. L. Volchenboum, K. Chou, M. Pearson, S. Madabushi, N. H. Shah, A. J. Butte, M. Howell, C. Cui, G. Corrado, and J. Dean (2018). Scalable and accurate deep learning with electronic health records. *npj Digital Medicine 1*, 1–10.

Ravì, D., C. Wong, F. Deligianni, M. Berthelot, J. Andreu-Perez, B. Lo, and G.-Z. Yang (2016). Deep learning for health informatics. *IEEE journal of biomedical and health informatics 21*(1), 4–21.

Reichart, R., K. Tomanek, U. Hahn, and A. Rappoport (2008). Multi-task active learning for linguistic annotations. In *Proceedings of ACL-08: HLT*, pp. 861–869.

Rink, B., S. Harabagiu, and K. Roberts (2011). Automatic extraction of relations between medical concepts in clinical texts. *Journal of the American Medical Informatics Association 18*(5), 594–600.

Roberts, K. and S. M. Harabagiu (2011). A flexible framework for deriving assertions from electronic medical records. *Journal of the American Medical Informatics Association 18*(5), 568–573.

Ross, S., G. Gordon, and D. Bagnell (2011). A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635.

Rosse, C. and J. L. Mejino Jr (2003). A reference ontology for biomedical informatics: the foundational model of anatomy. *Journal of biomedical informatics 36*(6), 478–500.

Roy, N. and A. McCallum (2001). Toward optimal active learning through sampling estimation of error reduction. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pp. 441–448. Morgan Kaufmann Publishers Inc.

Sadleir, L., M. Connolly, D. Applegarth, G. Hendson, L. Clarke, C. Rakshi, and K. Farrell (2004). Spasms in children with definite and probable mitochondrial disease. *European journal of neurology 11*(2), 103–110.

Sahoo, S. S., S. D. Lhatoo, D. K. Gupta, L. Cui, M. Zhao, C. Jayapandian, A. Bozorgi, and G.-Q. Zhang (2013). Epilepsy and seizure ontology: towards an epilepsy informatics infrastructure for clinical research and patient care. *Journal of the American Medical Informatics Association 21*(1), 82–89.

Sahoo, S. S., S. D. Lhatoo, D. K. Gupta, L. Cui, M. Zhao, C. Jayapandian, A. Bozorgi, and G.-Q. Zhang (2014). Epilepsy and seizure ontology: towards an epilepsy informatics infrastructure for clinical research and patient care. *Journal of the American Medical Informatics Association 21*(1), 82–89.

Savova, G. K., J. J. Masanz, P. V. Ogren, J. Zheng, S. Sohn, K. C. Kipper-Schuler, and C. G. Chute (2010). Mayo clinical text analysis and knowledge extraction system (ctakes): architecture, component evaluation and applications. *Journal of the American Medical Informatics Association 17*(5), 507–513.

Segura Bedmar, I., P. Martínez, and M. Herrero Zazo (2013). Semeval-2013 task 9: Extraction of drug-drug interactions from biomedical texts (ddiextraction 2013). In *SemEval*. Association for Computational Linguistics.

Segura-Bedmar, I., P. Martınez, and D. Sánchez-Cisneros (2011). The 1st ddiextraction-2011 challenge task: Extraction of drug-drug interactions from biomedical texts. In *The 1st DDIExtraction Challenge Task on Drug-Drug Interaction Extraction.*, pp. 1–9.

Settles, B. (2009). Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences.

Settles, B., M. Craven, and S. Ray (2008). Multiple-instance active learning. In *Advances in neural information processing systems*, pp. 1289–1296.

Seung, H. S., M. Opper, and H. Sompolinsky (1992). Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 287–294. ACM.

Sha, Y. and M. D. Wang (2017). Interpretable predictions of clinical outcomes with an attention-based recurrent neural network. In *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology,and Health Informatics*, ACM-BCB '17, New York, NY, USA, pp. 233–240. ACM.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell system technical journal 27*(3), 379–423.

Si, Y., J. Wang, H. Xu, and K. Roberts (2019, 07). Enhancing clinical concept extraction with contextual embeddings. *Journal of the American Medical Informatics Association 26*(11), 1297–1304.

Sioutos, N., S. de Coronado, M. W. Haber, F. W. Hartel, W.-L. Shaiu, and L. W. Wright (2007). Nci thesaurus: a semantic model integrating cancer-related clinical and molecular information. *Journal of biomedical informatics 40*(1), 30–43.

Smith, S. J. M. (2005). Eeg in the diagnosis, classification, and management of patients with epilepsy. *Journal of Neurology, Neurosurgery & Psychiatry 76*(suppl 2), ii2–ii7.

Socher, R., D. Chen, C. D. Manning, and A. Ng (2013). Reasoning with neural tensor networks for knowledge base completion. In *Advances in neural information processing systems*, pp. 926–934.

Stubbs, A., C. Kotfila, H. Xu, and Ö. Uzuner (2015). Identifying risk factors for heart disease over time: Overview of 2014 i2b2/uthealth shared task track 2. *Journal of biomedical informatics 58*, S67–S77.

Sukhbaatar, S., J. Weston, R. Fergus, et al. (2015). End-to-end memory networks. In *Advances in neural information processing systems*, pp. 2440–2448.

Sun, W., A. Rumshisky, and O. Uzuner (2013). Evaluating temporal relations in clinical text: 2012 i2b2 challenge. *Journal of the American Medical Informatics Association 20*(5), 806–813.

Sun, Z., W. Hu, Q. Zhang, and Y. Qu (2018). Bootstrapping entity alignment with knowledge graph embedding. In *IJCAI*, pp. 4396–4402.

Suominen, H., S. Salanterä, S. Velupillai, W. W. Chapman, G. Savova, N. Elhadad, S. Pradhan, B. R. South, D. L. Mowery, G. J. Jones, et al. (2013). Overview of the share/clef ehealth evaluation lab 2013. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pp. 212–231. Springer.

Sutton, R. S., D. McAllester, S. Singh, and Y. Mansour (1999). Policy gradient methods for reinforcement learning with function approximation. In *Proceedings of the 12th International Conference on Neural Information Processing Systems*, NIPS'99, Cambridge, MA, USA, pp. 1057–1063. MIT Press.

Tatum IV, W. O. (2014). *Handbook of EEG interpretation*. Demos Medical Publishing.

Trivedi, R., B. Sisman, X. L. Dong, C. Faloutsos, J. Ma, and H. Zha (2018). Linknbed: Multi-graph representation learning with entity linkage. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 252–262.

Tsuruoka, Y., Y. Tateishi, J.-D. Kim, T. Ohta, J. McNaught, S. Ananiadou, and J. Tsujii (2005). Developing a robust part-of-speech tagger for biomedical text. In *Panhellenic Conference on Informatics*, pp. 382–392. Springer.

Tsuruoka, Y. and J. Tsujii (2005). Bidirectional inference with the easiest-first strategy for tagging sequence data. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pp. 467–474. Association for Computational Linguistics.

Uzuner, Ö., B. R. South, S. Shen, and S. L. DuVall (2011). 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association 18*(5), 552–556.

Vapnik, V. (1999). *The Nature of Statistical Learning Theory*. Springer Science & Business Media.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin (2017). Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008.

Verga, P., E. Strubell, and A. McCallum (2018). Simultaneously self-attending to all mentions for full-abstract biological relation extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 872–884.

Wang, Y., L. Wang, M. Rastegar-Mojarad, S. Moon, F. Shen, N. Afzal, S. Liu, Y. Zeng, S. Mehrabi, S. Sohn, et al. (2018). Clinical information extraction applications: a literature review. *Journal of biomedical informatics 77*, 34–49.

Wang, Z., J. Zhang, J. Feng, and Z. Chen (2014). Knowledge graph embedding by translating on hyperplanes. In *AAAI*, pp. 1112–1119. Citeseer.

Weston, J., A. Bordes, O. Yakhnenko, and N. Usunier (2013). Connecting language and knowledge bases with embedding models for relation extraction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1366–1371.

Wu, F., J. Song, Y. Yang, X. Li, Z. Zhang, and Y. Zhuang (2015). Structured embedding via pairwise relations and long-range interactions in knowledge base. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Wu, Y., M. Jiang, J. Xu, D. Zhi, and H. Xu (2017). Clinical named entity recognition using deep learning models. In *AMIA Annual Symposium Proceedings*, Volume 2017, pp. 1812. American Medical Informatics Association.

Wu, Y., M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. In *arXiv preprint arXiv:1609.08144*.

Xu, K., J. L. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio (2015). Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pp. 2048–2057. JMLR.org.

Yang, B., W. Yih, X. He, J. Gao, and L. Deng (2015). Embedding entities and relations for learning and inference in knowledge bases. See (Bengio and LeCun, 2015).

Yang, Z., Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le (2019). Xlnet: Generalized autoregressive pretraining for language understanding. In *arXiv preprint arXiv:1906.08237*.

Yang, Z., D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy (2016). Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1480–1489. Association for Computational Linguistics.

Young, T., D. Hazarika, S. Poria, and E. Cambria (2018). Recent trends in deep learning based natural language processing. *ieee Computational intelligenCe magazine 13*(3), 55–75.

Zhang, Y., W. Zheng, H. Lin, J. Wang, Z. Yang, and M. Dumontier (2018). Drug–drug interaction extraction via hierarchical rnns on sequence and shortest dependency paths. *Bioinformatics 34*(5), 828–835.

Zhou, X., Q. Zhu, P. Liu, and L. Guo (2017). Learning knowledge embeddings by combining limit-based scoring loss. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 1009–1018. ACM.

Zhu, H., I. C. Paschalidis, and A. M. Tahmasebi (2018). Clinical concept extraction with contextual word embedding. In *NIPS Machine Learning for Health Workshop*.

Zhu, Y., R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 19–27.

BIOGRAPHICAL SKETCH

Ramon Maldonado was born in Corpus Christi, Texas. After completing his schoolwork at Mary Carroll High School in Corpus Christi in 2008, Ramon enrolled in Rice University, transferring to The University of Texas at Dallas in 2012 to major in Computer Science. He received a Bachelor of Science from The University of Texas at Dallas in 2014 and a Master of Science in 2018.

CURRICULUM VITAE

Ramon Maldonado

April 1, 2020

Contact Information:

Department of Computer Science
The University of Texas at Dallas
800 W. Campbell Rd.
Richardson, TX 75080-3021, U.S.A.

Email: ramon.m.maldonado@utdallas.edu
Website: www.hlt.utdallas.edu/∼ramon

Educational History:

B.S., Computer Science, The University of Texas at Dallas, 2014
M.S., Computer Science, The University of Texas at Dallas, 2018

Professional Recognitions and Honors:

American Medical Informatics Association Annual Symposium Distinguished Paper Award Winner, 2019
American Medical Informatics Association Clinical Research Informatics Distinguished Paper Award Winner, 2018
American Medical Informatics Association Clinical Research Informatics Distinguished Paper Award Nominee, 2017
Lars Magnus Ericsson Graduat Fellow, UTD, 2014-2016, 2018
Graduated *cum laude*, UTD, 2014

Publications:

Maldonado, Ramon and Sanda M. Harabagiu. "*Bootstrapping Adversarial Learning of Biomedical Ontology Alignments*." Proceedings of the American Medical Informatics Association (AMIA) Annual Symposium, 2019.

Maldonado, Ramon and Sanda M. Harabagiu. "*Active Deep Learning for the Identification of Concepts and Relations in Electroencephalography Reports*." Journal of Biomedical Informatics (JBI), Volume 98: 103265.

Maldonado, Ramon, Yetisgen, Maliha, and Sanda M. Harabagiu. "*Adversarial Learning of Knowledge Embeddings for the Unified Medical Lanuage System*." Proceedings of the American Medical Informatics Association (AMIA) Informatics Summit, 2019.

Maldonado, Ramon, Sullivan, Mark D., Yetisgen, Meliha, and Sanda M. Harabagiu. "*Hierarchical Attention-Based Prediction Model for Discovering the Persistence of Chronic Opioid*

Therapy from a Large Clinical Dataset." Proceedings of the American Medical Informatics Association (AMIA) Annual Symposium, 2018.

Maldonado, Ramon, Goodwin, Travis R., and Sanda M. Harabagiu. "*Memory-Agumented Active Deep Learning for Identifying Relations Between Distant Medical Concepts in Electroencephalography Reports*." Proceedings of the American Medical Informatics Association (AMIA) Informatics Summit, 2018.

Maldonado, Ramon, Taylor, Stuart J., and Sanda M. Harabagiu, "*UTDHLTRI at TREC 2018: Complex Answer Retrieval*." Text Retrieval Conference, 2018.

Maldonado, Ramon, Goodwin, Travis R., and Sanda M. Harabagiu. "*Deep Learning Meets Biomedical Ontologies: Knowledge Embeddings for Epilepsy*." Proceedings of the American Medical Informatics Association (AMIA) Annual Symposium, 2017.

Maldonado, Ramon, Taylor, Stuart J., and Sanda M. Harabagiu, "*UTDHLTRI at TREC 2017: Complex Answer Retrieval*." Text Retrieval Conference, 2017.

Goodwin, Travis R., Maldonado, Ramon and Sanda M. Harabagiu. "*Automatic Recognition of Symptom Severity from Psychiatric Evaluation Records*." Journal of Biomedical Informatics (JBI), Volume 75, Special Issue, pp.S71-S84, Vol. 9, Issue 2.

Maldonado, Ramon, Goodwin, Travis R., and Sanda M. Harabagiu. "*Active Deep Learning-Based Annotation of Electroencephalography Reports for Cohort Identification*." Proceedings of the American Medical Informatics Association (AMIA) Joint Summit on Translational Science (TBI), 2017.

Maldonado, Ramon, Goodwin, Travis R., Harabagiu, Sanda M., and Michael A. Skinner. "*The Role of Semantic and Discourse Information in Learning the Structure of Surgical Procedures*." International Conference on Healthcare Informatics (ICHI), pp. 223-232. IEEE, 2015.