2022

Temporal disambiguation of relative temporal expressions in clinical texts using temporally fine-tuned contextual word embeddings.

Amy L. Olex
Virginia Commonwealth University

DISSERTATION: TEMPORAL DISAMBIGUATION OF RELATIVE TEMPORAL EXPRESSIONS IN CLINICAL TEXTS USING TEMPORALLY FINE-TUNED CONTEXTUAL WORD EMBEDDINGS.

A Dissertation: submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at Virginia Commonwealth University.

by

AMY L. OLEX

Master of Science, Wake Forest University - September 2005 to May 2007

Director: Bridget T. McInnes,

Associate Professor, Department of Computer Science

Virginia Commonwealth University

Richmond, Virginia

May, 2022

Acknowledgements

Even as I write this I am having a hard time believing that I am finally here. Those who know me know my journey has been a long and winding up hill climb with many obstacles as I juggled a full time job, raising two children, and caring for my loving husband all while pursuing my PhD. However, I have finally arrived and I could not have done it without the support of so many people of whom I would like to recognize now.

First I would like to thank my dissertation committee, Dr. Bridget McInnes, Dr. Tamas Gal, Dr. Ozlem Uzuner, Dr. David Shepherd, and Dr. Kostadin Damevski for their guidance and invaluable feedback that helped me fine-tune my dissertation, highlight my contributions, and establish a path for future work. I specifically want to thank my advisor, Dr. Bridget McInnes, for her guidance and mentorship over the last 6 years. Her support and understanding enabled me to get up and keep going every time I felt beaten down (which was a lot). I really can't express in words how much she has meant to me over the years, but I am so grateful and honored to have been a student, and now a colleague, of hers!

I would also like to thank my classmates and fellow students of the McInnes Lab for always welcoming me and never hesitating to provide feedback even though I would fall out of touch quite often. Luke Maffey and Nick Morton get a special thank you to working with me on the very early versions of Chrono as a class project that eventually became my dissertation.

A thank you goes out to my colleagues at the Wright Center for Clinical and Translational Research for all their support over the years. Specifically, a big thank you to Dr. Moeller and Dr. Gal for your mentorship and for allowing me the freedom

to integrate my newly acquired NLP skills into my job while pursuing my PhD.

Dr. Jacquelyn Fetrow was my Master's thesis advisor and employer for 5 years following. Jacque, I credit a lot of where I am today to you. You raised me academically in an interdisciplinary environment, and taught me how to be a bridge across computational, biological, and clinical fields. It was your mentorship and training that helped me find my passion for working in an interdisciplinary field of research, and is the reason I have thrived in my current position at the CCTR. You also honed my scientific writing skills, which have continued to grow and have helped me be as productive as I am today. To this day I still think "What would Jacque do?" (i.e. WWJD) when I get stuck. Thank you for giving me such a strong foundation to stand on!

Stacy, you are not going to get out of these acknowledgements :) Stacy Howerton is my one and only best friend, and while we only talk about once a year I know she is always there for me, just as I will always be there for her. Thank you for your support and friendship over the years and for the years to come. It means the world to me to have your kindred spirit in my life!

Of course I won't forget to thank my family, who have been supporting me (and putting up with me) first-hand for many many years. When I first started to pursue my PhD my son, Ollie, was 2 years old and my daughter, Quinn, was 4 years old. How they have grown so much! I could not have accomplished all I have without my parent's support - Mom for always being there when I needed an ear, and my Dad for all your help with driving the kids around, helping with Mike, and making sure my yard didn't turn into a jungle.

Finally, I wanted to save the best for last and thank my loving husband Mike. We have been through quite a roller-coaster of a journey together with his diagnosis of MS, raising our two children, my full time job, and pursuit of a PhD. I have a tendency to operate with blinders on, like a lot, especially when I'm stressed. His loving support and brutal honesty kept me afloat (even though it often made me angry) and helped me juggle all my responsibilities. He is the one who never let go, who picked me up when I was crashing, who was by my side through the darkest of days, and he is the one who made me find balance in my life. Mike, you are the love of my life and I am grateful that you chose to go on this journey with me, and I will cherish every moment we have had and will have together, forever.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Abstract

DISSERTATION: TEMPORAL DISAMBIGUATION OF RELATIVE TEMPORAL EXPRESSIONS IN CLINICAL TEXTS USING TEMPORALLY FINE-TUNED CONTEXTUAL WORD EMBEDDINGS.

By Amy L. Olex

A Dissertation: submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at Virginia Commonwealth University.

Virginia Commonwealth University, 2022.

Director: Bridget T. McInnes,
Associate Professor, Department of Computer Science

Temporal reasoning is the ability to extract and assimilate temporal information to reconstruct a series of events such that they can be reasoned over to answer questions involving time. Temporal reasoning in the clinical domain is challenging due to specialized medical terms and nomenclature, shorthand notation, fragmented text, a variety of writing styles used by different medical units, redundancy of information that has to be reconciled, and an increased number of temporal references as compared to general domain texts. Work in the area of clinical temporal reasoning has progressed, but the current state-of-the-art still has a ways to go before practical application in the clinical setting will be possible. Much of the current work in this field is focused on direct and explicit temporal expressions and identifying temporal relations. However, there is little work focused on relative temporal expressions, which can be difficult to normalize, but are vital to ordering events on a timeline. This work introduces a new temporal expression recognition and normalization tool, Chrono, that normalizes temporal expressions into both SCATE and TimeML schemes. Chrono advances clinical timeline extraction as it is capable of identifying more vague and relative temporal expressions than the current state-of-the-art and utilizes contextualized word embeddings from fine-tuned BERT models to disambiguate temporal types, which achieves state-of-the-art performance on relative temporal expressions. In addition, this work shows that fine-tuning BERT models on temporal tasks modifies the contextualized embeddings so that they achieve improved performance in classical SVM and CNN classifiers. Finally, this works provides a new tool for linking temporal expressions to events or other entities by introducing a novel method to identify which tokens an entire temporal expression is paying the most attention to by summarizing the attention weight matrices output by BERT models.

CHAPTER 1

INTRODUCTION

Temporal reasoning is the ability to extract and assimilate temporal information to reconstruct a series of events such that they can be reasoned over to answer questions involving time (Figure 1). Medicine is one such area where temporal reasoning is vitally important to the care of patients. Temporal information in medicine plays a significant role in treatment decisions as the frequency of symptoms could mean the difference between being diagnosed with a given disease or not, or the need to undergo a procedure. For example, if a child tests positive for strep throat 7 times in a single year, the physician may consider performing a tonsillectomy [1]. Everything in medicine revolves around when things happen, such as when the first symptom appeared, when a lab test was performed, when medication or treatment was started, etc. Every aspect of a patient's medical data in their Electronic Health Record (EHR) contains some type of temporal component.

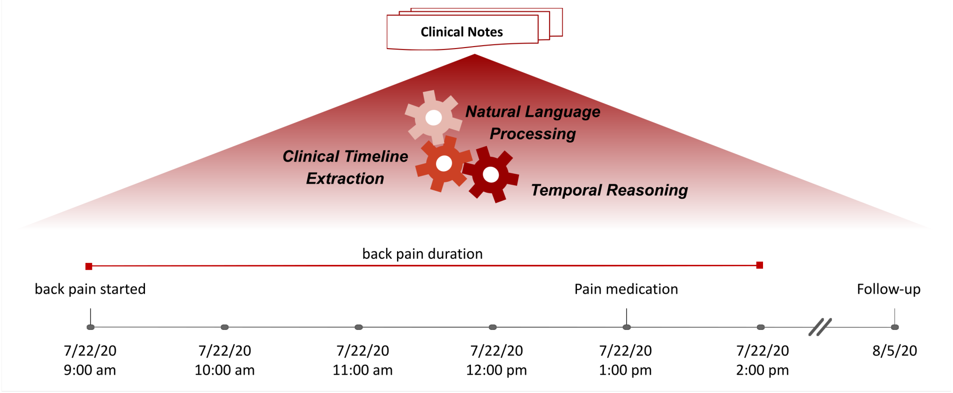


Fig. 1. Graphical representation of Temporal Reasoning on clinical notes.

One task performed by physicians prior to seeing a new patient is the review of that patient's medical history. This includes reconstructing the timeline of events leading up to a patient's present condition, but also reading and interpreting patient histories written by others to understand the patient's condition and how to best care for them. This requires knowing the progression of symptoms, when tests or medical procedures were or are to be performed, and when medications were taken and for how long. The most frequently read information when reconstructing a patient's history are the clinical notes [2]. This task can be time consuming and incomplete; however, it has been shown that visualizing longitudinal clinical data reduces the time it takes for medical professionals to assimilate a patient's information and assess their health status [3]. Having the ability to automatically extract and visualize a patient's medical timeline based on clinical notes would allow medical professionals the ability to grasp the patient's condition more quickly and completely without having to read through and digest potentially large numbers of clinical notes prior to providing care. To-date, there are a very small number of systems that can do this with unstructured clinical notes. The ability to automatically reconstruct a patient's medical history using both unstructured and structured EHR data would provide doctors a tool to help them assimilate vasts amounts of information about a patient's medical history quickly and efficiently, saving them time to focus on the patient instead of getting caught up on the patient's condition by browsing and reading numerous notes and reports. The advancement of Clinical Temporal Reasoning and Timeline Extraction in the field of Natural Language Processing is an area that aims to achieve this goal.

There has been a massive amount of work done on temporal information extraction over the past several decades [4, 5, 6, 7, 8, 9] with the ultimate goal of performing tasks that require temporal reasoning, which requires an events timeline. However, performance of current state-of-the-art timeline extraction pipelines are still not good enough to integrate into clinical practice leaving many areas of progress open to new and innovative ideas. Clinical texts from Electronic Health Records (EHRs) are a naturally temporally dense corpora, which means that clinical and medical texts derived from EHRs are rich in temporal information. However, clinical texts are difficult to parse due to a variety of idiosynchronicities such as medical jargon, shortened, non-grammatical sentences, department and even physician specific formatting, and medical abbreviations that can mean something different in different medical contexts.

This work advances progress in timeline extraction by focusing on improving the recognition and normalization of relative temporal expressions in clinical text, which appear frequently and are vital to ordering events on a timeline. Specifically, this work introduces a new temporal expression recognition and normalization tool, Chrono, that is capable if identifying more vague and relative temporal expressions than the current state-of-the-art, and by being the first to utilize temporally fine-tuned contextualized word embeddings to disambiguate relative temporal expression temporal types. This dissertation is organized as follows: Chapter 2 provides the reader with needed background, including a brief history of clinical temporal reasoning, discussion of the types of temporal information found in clinical texts, and a review of temporal annotation schemes, annotated corpora, shared tasks, and the timeline extraction pipeline. Additionally, background on representational learning, pre-trained models, and contextualized word embeddings are included in Chapter 2 followed by related work. Chapter 3 introduces a new temporal expression recognition and normalization tool, Chrono, that normalizes temporal expressions into the SCATE schema, with Chapter 4 discussing changes made to Chrono to modify it for the clinical domain, and Chapter 5 discussing modifications made to parse temporal expressions into the popular TimeML schema. Chapter 6 provides details on implementing and evaluating a temporal disambiguation module for relative temporal expressions, including comparison to current state-of-the-art methods and an End-2-End evaluation. Finally, Chapter 7 discusses future work and Chapter 8 summarizes the contributions of this work to the field of Clinical Natural Language Processing and Temporal Reasoning.

CHAPTER 2

BACKGROUND

2.1 A Brief History of Temporal Reasoning

The history of temporal reasoning with NLP started in the 1950's and was focused on general domain texts. Much of the work through the early 2000's focused on the linguistics behind the temporal nature of discourse, how to represent temporal information (such as with intervals or discrete points in time), temporal named entity recognition, the development of annotation standards for temporal information, and the temporal ordering of events [9]. While there is much to discuss with respect to general domain temporal reasoning, this work is focused on clinical temporal reasoning, so we refer the reader to other reviews for a history of temporal reasoning in the general domain [4, 5, 6, 7].

Clinical temporal reasoning has been around since the 1980's, but had trouble gaining traction with main-stream temporal reasoning NLP researchers due to the lack of access to a gold standard clinical corpus. In 2012, the Informatics for Integrating Biology and the Bedside (i2b2) Challenge was released [10], followed by the release of the THYME (Temporal Histories of Your Medical Events) corpus in the Clinical TempEval challenge of 2015 [11, 12]. For the first time, the NLP community at large had access to temporally annotated clinical corpora as gold standards for determining algorithm performance. This fueled the progress in temporal reasoning on unstructured clinical texts in the areas of temporal expression recognition and normalization, clinical event identification, and temporal relation classification. From the shared tasks that utilized these corpora, rule-based systems, such as HeidleTime

[13] and SUTime [14], emerged as performing the best for temporal expression recognition and normalization while statistical machine learning methods outperformed rule-based for event recognition and temporal relation classification tasks [10]. However, despite the boost in progress, temporal reasoning over clinical texts still posed several challenges, including the normalization of vague, relative, or implied temporal phrases; clinical event co-reference resolution; deciphering acronym and anaphoric expressions; identifying candidate temporal relationships; relative time anchoring; the end-to-end construction of a medical timeline from multiple documents; and the incorporation of structured EHR data with clinical narrative information [9, 10, 8].

The rest of this chapter is organized as follows: in Section 2.2 the types of temporal information found in clinical texts is described along with the distinction of explicit and relative expression types. Next, Section 2.3 reviews how this information is annotated for computational use, and Section 2.4 reviews clinical temporal reasoning annotated corpora and shared tasks. In Section 2.5 we describe each step involved in the generation of a clinical timeline from unstructured text along with a review of current progress for each, and Section 2.6 reviews the few end-to-end clinical timeline extraction pipelines for unstructured clinical texts. Finally, Section 2.7 reviews contextualized word embeddings along with recent work that attempts to incorporate them into temporal tasks. A version of this chapter was published in the Journal of Biomedical Informatics [8].

2.2 Temporal Information

Temporal information in text conveys information about the passage of time or specific points in time. The basic units of temporal information are dates and/or times (e.g. April 4, 2020, 11:45pm), and all expressions of temporal information in text are ultimately distilled down to some combination of dates and times. However, temporal information is not always expressed in such an explicit fashion. There are several different ways temporal information can be conveyed in text: explicit, implicit, relative, vague, and non-consuming [7].

Explicit temporal expressions are exactly that, they relay the value of temporal information explicitly in the basic temporal units of dates and/or times, such as “February, 18, 2020” or “9am”. Explicit temporal expressions, also known as *absolute* expressions, can be complete (“February, 18, 2020 at 9am”) or incomplete (“9am”) [15]. These types of expressions are generally straightforward to identify and normalize as they already contain the information needed to map to the timeline.

Implicit temporal expressions can be either globally implicit (aka relying on some global knowledge base of historical events) or locally implicit (relying on information given elsewhere in the current document). With clinical data, globally implicit references could also be those referring to other medical events in a patient’s record that are not included in the document being processed with the temporal information. For example, a patient could have had knee surgery in the past, but the current clinical note being processed only indicates “the patient fell 2 months after knee surgery”. In order to figure out when the patient fell, we would need to know when the surgery took place, which is part of the global knowledge about this patient’s medical history. On the other hand, locally implicit references are those that refer to explicit temporal information elsewhere in the same document. For example, the phrase “the patient fell on March 3rd” may precede the statement “the patient reported severe back pain 2 days after she fell”. The phrase “2 days after” is in reference to when the patient fell, which was explicitly stated in the text previously.

Relative temporal expressions are those that are either anchored to some event or the document time. Phrases such as “3 days ago” may be relative to the document creation time, whereas phrases such as “2 hours after taking her medication” are relative to when the event “taking her medication” occurred whether or not the absolute time the patient took medication is known.

Vague temporal expressions convey estimates of when something happened, but is unable to be converted into specific dates and times. The phrase “in the early 1980’s” represents an unknown period of time presumably in the first few months of the year 1980; however, the start date, end date, and duration are vague and unknown. Similarly, clinical notes can contain phrases such as “the pain started about 2 months ago”. The phrase “about 2 months ago” is both relative to the document time as well as vague due to the word “about”.

Finally, Lim, et al. [7] described a *non-consuming temporal information*, which is temporal information that is not explicitly stated in the text, but is assumed to be provided or is general knowledge. This generally represents the document creation time, of which many of the relative references may be anchored to.

Being able to define the types of temporal information is great, but if the computer can’t process or utilize that data it is useless for NLP. Thus, it is necessary to define *temporal annotation schemes*, which format temporal information from text in a way that is more easily processed by computers for temporal reasoning tasks.

2.3 Temporal Annotation

Annotation schemes are used to normalize unstructured information in texts to a computer-readable format for downstream processing. Temporal annotation schemes are specifically designed to normalize temporal information and related events into a standard format that can be utilized for temporal reasoning tasks. This includes providing standardized formats for temporal expressions, events, and the many different types of relationships between them. How an annotation scheme is defined has a major influence on how temporal and event information in text is processed and interpreted. Temporal annotation schemes have evolved over the past 30 years from having a simple temporal value attribute for annotated events (Message Understanding Conferences of the 1990's), to specifically annotating temporal expressions with an expanding set of attributes and adding tags for events and temporal links (TIDES and TimeML), to specialization for the clinical domain (THYME-TimeML), and the development of an interval and semantic-based temporal schema (SCATE). Figure 2 provides a high-level summary of the evolution of temporal annotation schema with details being discussed in the following sub sections.

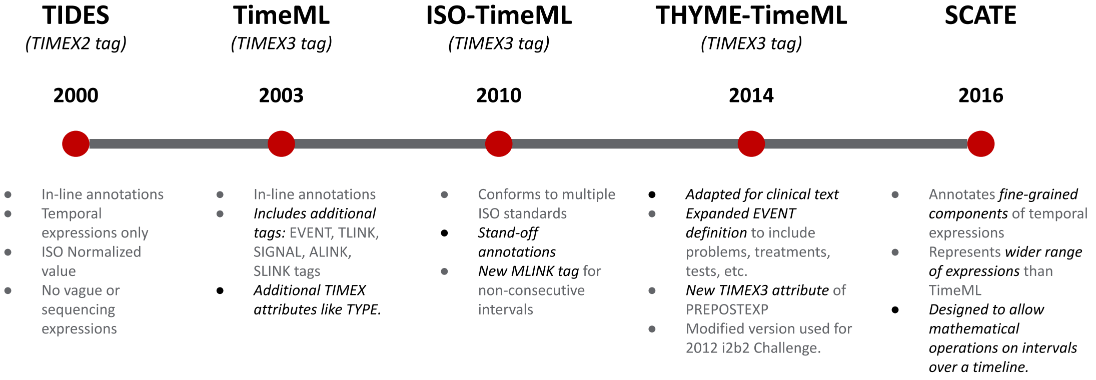

TIDES
(TIMEX2 tag)
2000
In-line annotations
Temporal expressions only
ISO Normalized value
No vague or sequencing expressions
TimeML
(TIMEX3 tag)
2003
In-line annotations
Includes additional tags: EVENT, TLINK, SIGNAL, ALINK, SLINK tags
Additional TIMEX attributes like TYPE.
ISO-TimeML
(TIMEX3 tag)
2010
Conforms to multiple ISO standards
Stand-off annotations
New MLINK tag for non-consecutive intervals
THYME-TimeML
(TIMEX3 tag)
2014
Adapted for clinical text
Expanded EVENT definition to include problems, treatments, tests, etc.
New TIMEX3 attribute of PREPOSTEXP
Modified version used for 2012 i2b2 Challenge.
SCATE
2016
Annotates fine-grained components of temporal expressions
Represents wider range of expressions than TimeML
Designed to allow mathematical operations on intervals over a timeline.

Fig. 2. Timeline of temporal annotation schema.

2.3.1 Translingual Information Detection, Extraction, and Summarization (TIDES)

TIDES was first developed in 2000 after the Defense Advanced Research Projects Agency (DARPA) sponsored Message Understanding Conference (MUC) challenges to provide more semantic details to the temporal expression (TIMEX) tags used in previous challenges. To differentiate, TIDES refers to temporal expression tags as TIMEX2 tags. This annotation focused on temporal expressions only, and are explicit in what can and can't be annotated with TIMEX2 tags. Any temporal expression that has enough information to be pinned to a timeline is considered markable in the

TIDES schema. Tag attributes capture semantic information about the expression as well as its ISO-8601 normalized value. Table 1 shows a few TIMEX2 tags for different temporal expressions (examples taken from [16]). Tag attributes include the normalized value of the expression (VAL), if a temporal modifier exists (MOD), the anchoring date and time (ANCHOR_VAL), the relative direction of the anchor value (e.g. before or after) with respect to VAL, and whether or not this TIMEX2 is annotating a set of temporal values, such as a frequency (SET). In general, vague expressions that do not have enough information to identify a spot on a timeline are not markable by TIDES [16]. This includes sequencing and ordering expressions like "subsequent", manner adverbs such as "immediately", non-quantifiable durations like "permanently", negatives and non-existent times (e.g. "no time"), the token "time" when it refers to a situation or occasion such as "at this time", and frequency expression without a quantifier (e.g. "frequently" or "too often").

2.3.2 TimeML

In 2003, a new scheme named TimeML [17] was released that defined a new TIMEX tag named TIMEX3. The TIMEX3 tag is based off of the the original TIMEX tag [18] and the TIMEX2 standard from TIDES [16]; however, TimeML also includes the additional tags EVENT, TLINK, SIGNAL, ALINK and SLINK. This new scheme addresses the issue of not annotating events and the relationships between temporal expressions and events, which are key components to understanding the temporal nature of a text. While the TIMEX3 tag is based off of previous TIMEX tags, it's composition makes it difficult to convert TIMEX2 annotations to TIMEX3 due to the added and altered attributes [19].

The TIMEX3 tag is used for explicit time expressions including dates, times, and durations. In the TimeML scheme, the TIMEX3 tag has more attributes than

I returned to work at *twelve o'clock January 3, 1984*. I returned to work at <TIMEX2 VAL="1984-01-03T12:00">twelve o'clock January 3, 1984</TIMEX2>.
There has been a lot of rain *the past three weeks*. There has been a lot of rain <TIMEX2 VAL="P3W" >the past three weeks</TIMEX2>.
She has been at work for *more than a month*. She has been at work for <TIMEX2 VAL="P1M" MOD="MORE_THAN" >more than a month</TIMEX2>.
Two years ago, the dance club drew about 100 students *each week*. <TIMEX2 VAL="1997">Two years ago</TIMEX2>, the dance club drew about 100 students <TIMEX2 SET="YES" VAL="1997-WXX">each week</TIMEX2>

Table 1. Example temporal expressions with TIMEX2 annotation.

TIMEX2 that provide more semantic information about each expression. These include the type (DATE, TIME, DURATION, or SET), beginPoint, endPoint, a quantifier such as “every”, a frequency such as “2X”, the document function (e.g. CREATION_TIME, PUBLICATION_TIME, EXPIRATION_TIME, etc), if this is a temporal function (true or false), a value, modifiers, and an anchor time. Example TIMEX3 annotations can be found in Table 2.

Other tags now included in the TimeML annotation scheme include an EVENTS tag, which annotates the events that are needing to be placed on a timeline. Previously, TIDES did not annotate events or relationships between temporal expressions and events. The SIGNAL tag is used to annotate function words that indicate how two temporal objects (TIMEX or EVENT) are related to each other (e.g. “when”, “in”, and “after”) where previously these tokens were not mark-able by TIMEX2. The TLINK tag indicates a temporal link/relationship for EVENT-TIMEX, EVENT-EVENT, or TIMEX-TIMEX pairs. Finally, the SLINK and ALINK tags are also relationships tags, but are used when an event is part of another event.

2.3.3 ISO-TimeML

In 2010, the TimeML scheme was modified to provide an interoperatable temporal annotation scheme that conforms to the ISO standards ISO 24610-1:2006 FSR, ISO DIS 24611 MAF, and ISO DIS 24612 LAF [20]. The primary change was the move from in-line annotations to stand-off annotations where the text being processes is not altered (Table 3). The ISO-TimeML standard also introduces a new MLINK, which is interpreted as a MEASURE and is associated with durations. Previously, durations were treated as consecutive intervals; however, this is not always the case. For example, in the phrase “Sam taught for 4 hours”, it is ambiguous if Sam taught a consecutive 4 hours or for a total of 4 hours with breaks in between. The new

I **returned** to work at *twelve o’clock January 3, 1984*. I <EVENT eid=“e1” class=“OCCURENCE”>returned</EVENT>to work at <TIMEX3 tid=“t1” type=“TIME” value=“1984-01-03T12:00”>twelve o’clock January 3, 1984</TIMEX3>.
There has been a lot of **rain** *the past three weeks*. There has been a lot of <EVENT eid=“e1” class=“OCCURENCE”>rain</EVENT><TIMEX3 tid=“t2” type=“DURATION” value=“P3W”>the past three weeks</TIMEX3>.
She has been at **work** for *more than a month*. She has been at <EVENT eid=“e1” class=“OCCURENCE”>work</EVENT>for <TIMEX3 tid=“t3” type=“DURATION” value=“P1M” mod=“MORE_THAN” >more than a month</TIMEX3>.
Two years ago, the dance club **drew** about 100 students *each week*. <TIMEX3 tid=“t4” type=“DURATION” value=“P3Y”>Two years ago</TIMEX3>, the dance club <EVENT eid=“e1” class=“OCCURENCE”>drew</EVENT>about 100 students <TIMEX3 tid=“t5” type=“SET” value=“P1W” quant=“EACH” freq=“1w”>each week</TIMEX2>

Table 2. Example temporal expressions with TimeML annotations.

I **returned** to work at *twelve o’clock January 3, 1984*. <EVENT eid=“e1” start=3 end=11 class=“OCCURENCE”> <TIMEX3 tid=“t1” start=23 end=53 type=“TIME” value=“1984-01-03T12:00”>
There has been a lot of **rain** *the past three weeks*. <EVENT eid=“e1” start=25 end=29 class=“OCCURENCE”> <TIMEX3 tid=“t2” start=30 end=50 type=“DURATION” value=“P3W”>
She has been at **work** for *more than a month*. <EVENT eid=“e1” start=17 end=21 class=“OCCURENCE”>work</EVENT> <TIMEX3 tid=“t3” start=25 end=42 type=“DURATION” value=“P1M” mod=“MORE_THAN” >
Two years ago, the dance club **drew** about 100 students *each week*. <TIMEX3 tid=“t4” start=1 end=15 type=“DURATION” value=“P3Y”> <EVENT eid=“e1” start=32 end=36 class=“OCCURENCE”> <TIMEX3 tid=“t5” start=56 end=65 type=“SET” value=“P1W” quant=“EACH” freq=“1w”>

Table 3. Example temporal expressions with TimeML-ISO stand-off annotations.

MLINK relation allows one to identify this ambiguity in the annotation. Additionally, the ISO-TimeML standard provides a mechanism for proper counts of recurring events, where the previous annotation for the phrase “Sam taught every Tuesday in December” was unclear if the event “taught” occurred once or multiple times, the new ISO-TimeML provides a distributive mechanism to properly count the number of teaching events as four.

2.3.4 THYME-TimeML

Due to the unique challenges of temporal information extraction in the clinical domain, Styler, et al. [12] developed the THYME-TimeML guidelines in 2014 for annotating temporal information from clinical texts. In THYME-TimeML, the definition as what qualifies as an event is expanded to include anything that would occur in a patients clinical timeline and is clinically relevant, such as diagnoses, tumor, illness, or procedure [12]. While the concept of EVENT was expanded, the attributes required were both simplified and expanded to be more relevant to clinical documents. For example, a new attribute for "contextual modality" was added that includes the values ACTUAL, HYPOTHETICAL, HEDGED, and GENERIC. Additionally, a new type of temporal expression is added to the TIMEX3 tag of PREPOSTEXP, which refers to the clinically relevant and temporally complex terms such as preoperative, postoperative, and intraoperative.

Due to the higher frequency of temporal information in a clinical note or document, the number of temporal relations that needed to be annotated was large. Styler, et al. were concerned about consistency in annotation and aimed to reduce the number of necessary TLINK annotations. They created the concept of a narrative container that was relative to the document creation time. Instead of creating TLINKS for all possible events, all events were placed into one of four narrative containers: "before the DOCTIME, before and overlapping the DOCTIME, just overlapping the DOCTIME or after the DOCTIME" [12]. Both EVENTs and TIMEXs can be used as anchors for a narrative container. The advantage of this approach is that events are placed within an explicit temporal bound and it is not necessary to create all possible TLINKs to identify whether an event comes before, after, or during another event. Thus, with this change the CONTAINS relation is now the most frequent out of the previous relation types of BEFORE, OVERLAP, BEGINS-ON, and ENDS-ON. Due to the clinical relevance of the THYME-TimeML scheme, a simplified version was used as the basis of the 2012 i2b2 temporal challenge [21] annotation framework.

2.3.5 SCATE

The SCATE Schema (Semantically Compositional Annotations for Temporal Expressions) was developed by Bethard, et al. [22] to address some of the shortcomings of the ISO-TimeML standard [20]. Specifically, ISO-TimeML has trouble representing time intervals that do not map to a specific calendar date, such as "2 summers ago", temporal expressions can only be relative to other times, and not events as in "four days postoperative", and the flattened structure of ISO-TimeML annotations removes the compositional structure of temporal expressions. Thus, SCATE was developed to annotate the fine-grained components of temporal expressions, to represent a wider variety of temporal expressions, allowing for events to act as anchors, and using mathematical operations over a timeline to define the semantics of each annotation. Figure 3 demonstrates the differences between SCATE annotation and that of the ISO-TimeML annotations.

2.4 Clinical Temporal Reasoning Shared Tasks and Corpora

Shared Tasks provide a centralized and structured platform for advancing specific areas of research in NLP. An overview of the shared tasks in the general and clinical domains that include some type of temporal component is shown in Figure 4. Temporal Reasoning and Information Extraction first appeared in NLP shared tasks in the 1990's with the Message Understanding Conferences (MUC 6, 7 and 8) [23, 24, 25]. These early tasks, however, were not focused on temporal information extraction, but rather identifying the temporal value of an entity, if present, as part of

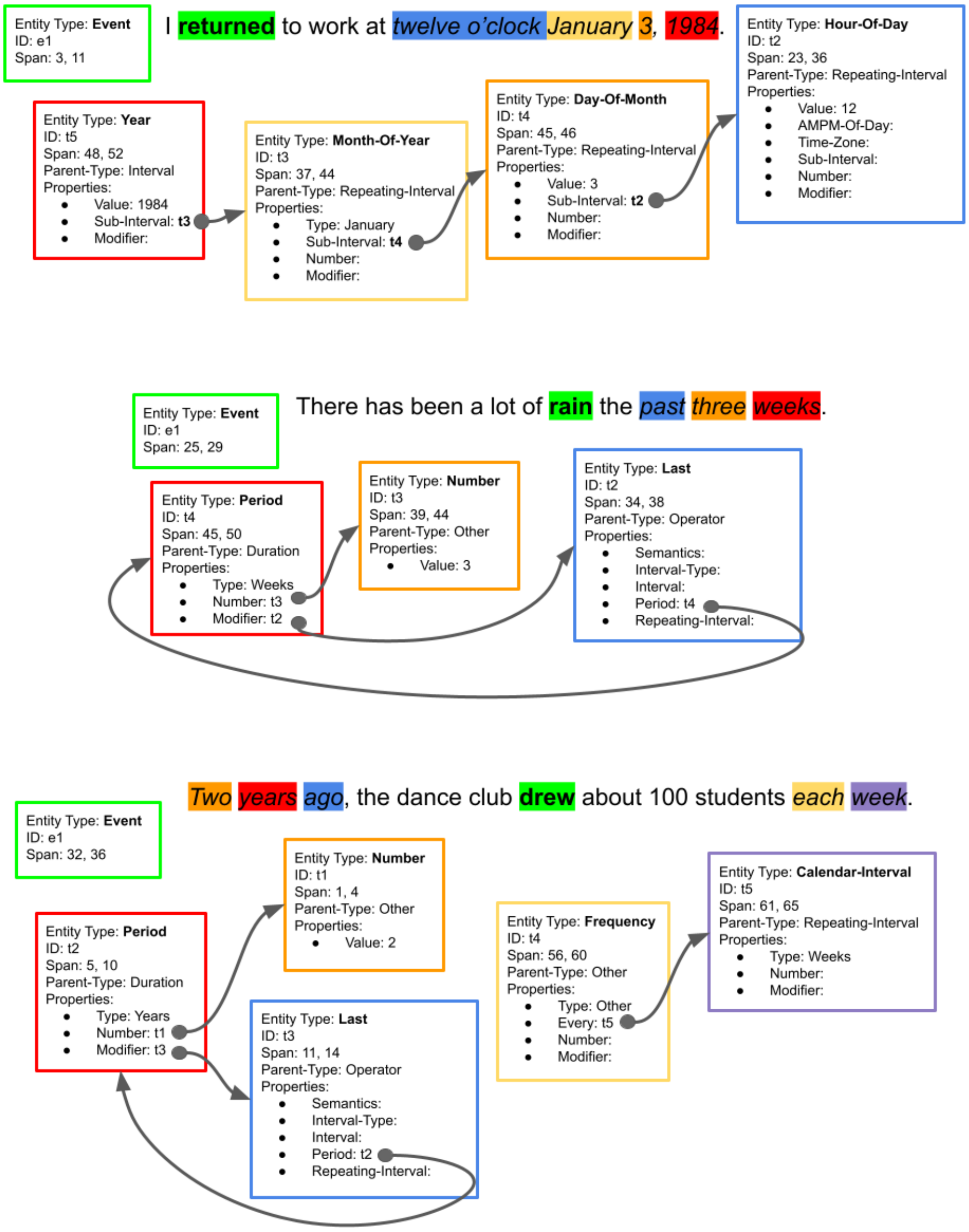


Fig. 3. Example SCATE annotations.

a named entity identification task. Temporal information extraction did not become the focus of a shared task until 2004 in the ACE TERN challenge [26], followed by a series of TempEval challenges [27, 28, 29]. The TempEval challenges focused on identifying and normalizing temporal expressions and identifying a variety of temporal relations, and were foundational in developing temporal reasoning systems in the general domain. Lim, et al. [7] provides a good overview of these challenges and their contributions to the field. In this section, we focus on the temporal challenges and corpora in the clinical domain.

2.4.1 2012 i2b2 Temporal Challenge and Corpus

Clinical temporal information extraction and reasoning shared tasks did not appear until 2012 with the Informatics for Integrating Biology and the Bedside (i2b2) Clinical Temporal Relations Challenge [21]. This task provided the NLP community with 310 de-identified discharge summaries from Partners Healthcare and the Beth Israel Deaconess Medical Center with gold standard temporal annotations, and included 3 tracks: 1) temporal expression (TIMEX) and clinically relevant event identification (EVENT), 2) temporal relation identification (TLINK), and 3) end-to-end system. Unlike previous general domain temporal shared tasks, the i2b2 challenge narrowed events to those that were clinically relevant, namely clinical concepts (i.e. problems, treatments, and tests), clinical departments, evidentials, and occurrences. Temporal expressions were annotated with the ISO-TimeML scheme and included dates, times, durations, and frequencies with absolute values normalized to the ISO-8601 standard. Temporal relations consisted of all possible relations between two TIMEXs, two EVENTs, or a TIMEX and EVENT. Relation types consisted of BEFORE, AFTER, SIMULTANEOUS, OVERLAP, BEGUN_BY, ENDED_BY, DURING, and BEFORE_OVERLAP. Ultimately, 18 teams officially participated in the challenge; however, in the years after several other systems were published utilizing the corpus and further advancing the field [30, 31, 32, 33, 34, 35, 36, 37, 38].

2.4.2 TempEval Challenges and THYME Corpus

From 2015 to 2017, Bethard, et al. [11, 39, 40] hosted a series of Clinical TempEval challenges as part of SemEval following the task structure of previous TempEval shared tasks, only moving to the clinical domain. These challenges used the Temporal Histories of Your Medical Events (THYME) corpus [12], which is composed of 1,254 de-identified clinical notes and pathology reports from brain and colon cancer patients seen at the Mayo Clinic. The 2015, 2016, and 2017 challenges consistent of the same 9 tasks grouped into 3 categories: 1) TIMEX identification, 2) EVENT identification, and 3) TLINK identification. TIMEX types included DATE, TIME, DURATION, QUANTIFIER, PREPOSTEXP, and SET. Identified EVENTs were classified as ASPECTUAL, EVIDENTIAL, or N/A, and required properties to be annotated such as polarity and contextual modality. TLINKs were broken down into 2 main categories: DOCTIMEREL or CONTAINS. DOCTIMEREL relations were those where an EVENT has a BEFORE, OVERLAP, BEFORE-OVERLAP, or AFTER relation with the document creation time. CONTAINS relations were narrative container relations between EVENTs and TIMEXs. Participating systems were evaluated either as end-to-end systems, or were given TIMEX and EVENT annotations and judged on TLINK identification. Due to issues in getting participants access to the data in 2015, the same challenge was run again in 2016 with 4 times as many participating teams. In both instances, teams using supervised machine learning approaches excelled in the TIMEX and EVENT identification tasks using a variety of features, indicating that these tasks are close to solved. However, the TLINK tasks proved to be challenging to all systems, especially the narrative container relations. In 2017, the same tasks were run again, however the aim was to address how well systems dealt with domain adaptation. Participating systems were given the colon cancer

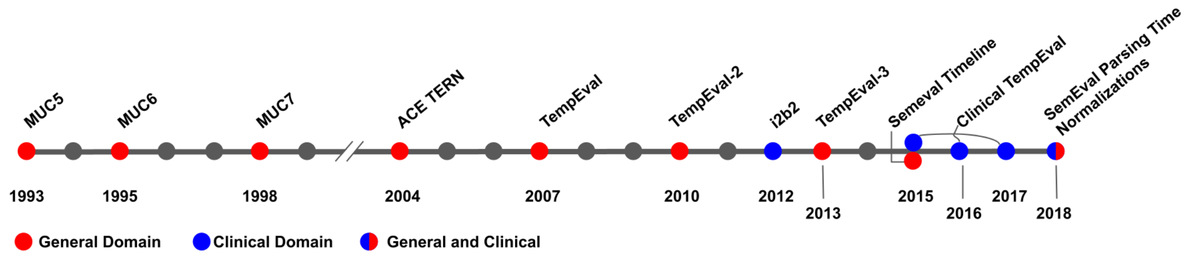


Fig. 4. Timeline of Shared Tasks that include Temporal Components

clinical notes and pathology reports for training, but were tested on the brain cancer cohort. All systems were reported to have a significant performance drop when tested on a new domain, which indicates that there is much work to be done in creating a generalizable timeline extraction system for clinical data.

The most recent temporal reasoning shared task was the SemEval 2018 Task 6: Parsing Time Normalizations [41] using the THYME corpus and the general domain AQUAINT News wire corpus. The goal of this task was to normalize fine-grained temporal information and relationships into the Semantically Compositional Annotations for Temporal Expressions (SCATE) scheme developed by Bethard, et al. [22]. This scheme aims to improve upon the current TIMEX3/TimeML [17] standard by representing a wider variety of temporal expressions, allowing for events to act as anchors, and using mathematical operations over a timeline to define the semantics of each annotation. Two tracks were assessed: 1) Parsing text to time entities (event parsing and temporal relations were not assessed), and 2) production of time intervals. While over 40 teams registered for the challenge, only one team submitted results for comparison with the organizer's baseline–our system Chrono [42], described below in Chapter 3.

2.5 Timeline Extraction

Timeline extraction is a high level temporal reasoning task that relies on the accurate performance of lower level Temporal Information Extraction (TIE) tasks. In order to reconstruct a useful and non-redundant medical timeline that can be used in clinical settings, the following steps must be implemented:

1. Temporal Expression Identification and Normalization
2. Clinical Event Identification
3. Temporal Relation Identification
4. Clinical Event Co-Reference Resolution
5. Temporal Event Ordering
6. Timeline Visualization

In the following subsections we define and review the current state of each of these step in the Clinical NLP realm along with a review of existing end-to-end clinical timeline extraction systems and the state of evaluating these systems.

2.5.1 Temporal Expression Recognition and Normalization (TERN)

A *temporal expression* (TimeML tag TIMEX3), referred to as a TIMEX, is a phrase that conveys information about time. *Temporal expression recognition* is the task of identifying which span of text contains a TIMEX. Temporal expressions can be annotated as one of four types in the TimeML schema: DATE, TIME, DURATION, and FREQUENCY. Temporal expressions can either be explicit ("February, 18, 2020"), relative ("after the surgery"), implicit ("on Labor Day"), vague ("about 6 months ago"), or a combination ("about 2 weeks after Labor Day"). Non-consuming temporal information is generally not annotated as a temporal expression and is instead used to anchor relative expressions to implicit information such as the document creation time. Each TIMEX is composed of one or more *temporal entities*, such as day-of-week, hour-of-day, month-of-year, etc. For example, the TIMEX “February 18, 2020” contains 3 temporal entities: month-of-year, day-of-month, and year. A TIMEX can also contain *modifiers* that indicate whether the temporal phrase is referring to the past, present or future (e.g. “last month” vs “this month” vs “next month”).

Temporal expression normalization is the task of interpreting the value of an identified TIMEX and storing it in a computer-readable format. This format is generally the ISO-8601 standard, which is supported by the current temporal annotation schemes discussed in Section 2.3. In general, explicit dates and times, such as “May 4, 2020”, are straight-forward to normalize into the ISO standard. However, relative, implicit and vague TIMEXs, henceforth referred to as a RelIV-TIMEX, are difficult to process due to the need for additional implicit information, which may be easily identifiable to a human reader, but not apparent to a computer. In addition, RelIV-TIMEXs can refer to a single point in time (e.g. “she had surgery two weeks prior to admission”), or a span of time (e.g. “she has been having knee pain since two weeks prior to admission”). Using the TimeML schema, RelIV-TIMEXs referring to a single point in time are classified as a DATE type, and those referencing a span of time are classified as a DURATION. Knowing the difference between these two types of expressions is important because it determines how that expression is going to be normalized. Thus, for RelIV-TIMEXs there are 3 phases to the TERN task: recognition, type classification, and normalization.

Initial progress in TERN came from the general domain with the TempEval-2 and 3 challenges in 2010 and 2013 [28], where TIMEXs were to be recognized and normalized, including the identification of the type and value attribute of the TIMEX3 tag in the TimeML annotation scheme. In both of these challenges, the rule-based

HeidelTime [13] tool achieved the best performance in identifying TIMEXs and their type. Other high performing systems included the data-driven ClearTK [43] system, which used SVMs, and the rule-based SUTime [14]. However, while all of these tools performed well on identifying a TIMEX span and type, it is clear from all the teams in both challenges that assigning the correct value of the TIMEX was a difficult problem indicated by the drop in performance.

TERN moved to the clinical domain with the 2012 i2b2 Temporal challenge [21]. Similar to the TempEval challenges, i2b2 utilized the TIMEX3, EVENT, and TLINK tags from the ISO-TimeML scheme; however, in addition to the type and value of TIMEX3 attributes, the i2b2 challenge also required the modifier attribute to be set. The top performing system for the TIMEX task was a rule-based system from Mayo Clinic [44]. All of the other systems were either rule-based or hybrid methods, with several integrating the top TempEval performer, HeidelTime, into their workflow. Interestingly, the HeidelTime team also participated in the i2b2 challenge; however, the tool trained for general-domain temporal information extraction performed poorly on the clinical data. This is likely due to the added challenges of parsing text that is temporally dense and contains many more frequency-based expressions than general domain text [45]. However, teams that integrated HeidelTime in with additional rules to compensate for the added clinical challenges performed better [21]. While the top performing systems did well on identifying the span, type, and modifier for the TIMEX3 tag, they all still saw a drop in performance when it came to identifying the normalized value. Since the i2b2 challenge, rule-based systems are by far the preferred method for the task of recognizing and normalizing TIMEXs from clinical text [46, 33, 38, 47, 48] followed by hybrid [49, 50] and purely data-driven [51, 52] approaches.

2.5.2 Clinical Event Identification

Event identification (TimeML tag EVENT) is the task of identifying things that happen and are of importance in a text. In temporal information extraction, events also have some type of temporal component, such as when or for how long a thing happened. In the general domain, an event is defined as a situation, action, or state of being [5], and are generally represented by syntactically inflected verbs or event nominals such as “killed” and “crash” in the statement “she was killed by the crash” (example taken from [28]). However, in the clinical domain we are interested in clinically relevant events, which expands the definition of “event”. The 2010 and 2012 i2b2 challenges defined a clinical event as a clinical concept (i.e. problem, treatment, or test), clinical departments, evidentials (i.e. identifying the source of information), and occurrences (i.e. events like “admission” or “transfer” that happen to a patient) [53, 21]. Clinical event identification is similar to the classic Named Entity Recognition (NER) NLP task, except we are only looking for certain medically-related types of entities. In both the 2010 and 2012 i2b2 challenge tasks on clinical event identification systems implementing conditional random fields (CRF) for event span detection, and support vector machines (SVMs) for event attribute classification performed the best for this task.

While the i2b2 challenges define events as tests, treatments, and medical problems, this may not cover all relevant events depending on the task at hand. For this reason, Dehghan [49] expanded the definition of event to also include health and quality of life indicators, which are relevant when constructing a timeline of important events for a patient with a disease that causes long-term mental and emotional health issues such as childhood CNS cancer. Thus, it is important to realize that the definition of event is task-specific, so building a catch-all classifier may be challenging.

2.5.3 Temporal Relation Identification

Temporal relations (TimeML tag TLINK), are relationships between two entities that have a temporal component. Entities are either EVENTs or TIMEXs. TLINKs can exist between a TIMEX and an EVENT, between two EVENTs, between two TIMEXs, and between each EVENT and the document creation time (DCT). For example, in the phrase "a full hip replacement was performed on 6/22/92" the event is "hip replacement" and the TIMEX is "6/22/92", where the TIMEX refers to when the event happened. EVENT-EVENT relations can also occur, for example, in the phrase "she became sick after visiting the store" there are 2 events, but no explicit temporal references. However, the event "became sick" is temporally linked to the event "visiting the store" with a relative TLINK.

Until the 2012 i2b2 challenge, not much attention had been paid to temporal relation extraction in the clinical domain [10]. The majority of the work was done in the general domain [6, 7, 4], and temporal relation extraction in the clinical domain posed new challenges. This included having to deal with the large increase of temporal expressions due to the increased temporal density of clinical notes versus general domain text [45], which creates more candidate relations needing to be filtered to the medically relevant ones. It also includes dealing with implicit relations and relative relations where no explicit temporal expression can be found. Finally, clinical text is highly redundant [2, 54] and requires sophisticated co-reference resolution techniques in order to build out non-redundant timelines of events.

The TLINK track of the 2012 i2b2 challenge spurred a variety of machine learning and hybrid approaches to identifying TLINKs [21]. Hybrid approaches utilizing rule-based pair selection, CRFs, and SVMs performed the best; however, significant challenges still remained. While systems identified relations between entities and the document creation time (DCT) accurately, all systems had difficulty narrowing the candidate entity pairs, anchoring relative temporal expressions, and identifying inter-sentence TLINKs. These still remain areas in need of improvement. Since the i2b2 challenge there has been some progress in TLINK identification for clinical texts. In the following subsections we review some early attempts to handle inter-sentence TLINKs, strategies to target specific types of TLINKs, neural network approaches, and the contributions of BERT to temporal relation extraction.

2.5.4 Early Inter-Sentence Temporal Relation Identification Strategies

Inter-sentence TLINKs are difficult as they also require some level of coreference resolution. Cheng, et al. [30] found their supervised machine learning classifier using MaxEnt performed poorly on these, so implemented a rule-base approach for inter-sentence TLINKs. D'Souza [32] found that many of the inter-sentence TLINKs were not actually annotated in the i2b2 corpus, so proceeded to augment the i2b2 annotations to compensate. Lin, et al. [37] contributions were the first open source state-of-the-art end-to-end system that performed comparably to the top i2b2 system. It used the same features as previously published systems and implemented a multi-layered approach by identifying course-grained relations followed by intermediate and fine-grained.

2.5.4.1 Targeting Specific Temporal Relation Types

Many of the systems derived from the i2b2 challenge tried to identify both the explicit and implicit temporal relations simultaneously. This resulted in a lot of relations that need to be identified. Lee, et al. [34] argued that systems need to be good at identifying explicit temporal relations before identifying the implicit relations as they are based on the explicit temporal relations. Thus, Lee and colleagues [34, 35, 36] focused their work on defining, annotating, and identifying "direct" temporal relations, which are defined as explicit relations between an event and timex from within the same sentence, using the i2b2 corpus. While their initial system based off of the top performing Vanderbilt system from the i2b2 challenge performed poorly [34], subsequent attempts that included the augmentation of machine learning features to include document format features, dependency features, and semantic roles along with previously used lexical, syntactic, and contextual features improved performance [35]. More recently, Guan, et al. [36] has produced the latest state-of-the-art in direct TLINK identification using RoBERTa [55], which is a multi-layer transformer encoder modified from the original BERT [56]. By training on more data, longer sequences, removing the next sentence prediction objective, and altering the masking pattern, Guan, et al. [36] utilized BERT to predict the temporal relation presence and type using the direct temporal relation annotated corpus from Lee, et al. [34]. However, similar to previous work, only intra-sentence relations are targeted that meet specific lexical-syntactic requirements for a direct temporal relation (defined in [34]).

Viani, et al. [57] also reduced the types of TLINKs identified in their work, but narrowed them to a specific medical condition instead of using syntactic properties like Lee, et al. [34]. Viani, et al. implemented a rule-based system to identify TLINKs that were relevant to the symptom onset timeline of patients diagnosed with schizophrenia. Because their system rule-set was built for a specific clinically oriented task for a specific medical condition, the developed rules and methodology are not generalizable, and many of the rules depend on their hospital system's specific format of clinical notes. Even with these highly specific constraints on the types of TLINKS targeted, and the manual annotation of events and TIMEXs for the corpus used, Viani, et al.'s performance had an accuracy of 0.67, which is not much better than the more generalizable system produced by Lee and Guan [34, 35, 36].

2.5.4.2 Neural Network Approaches

Prior to 2017, temporal expression relation extraction was primarily implemented via a rule-base or hybrid approach with Support Vector Machines (SVMs) commonly chosen as the machine learning component using hand-crafted linguistic features [37, 44, 58, 59, 60, 61, 62, 31, 33]. However, in 2017 neural network architectures took the stage for the temporal relation extraction task. Dligach, et al. [63] and Tourille, et al. [64] were among the first to utilize neural networks for temporal relation extraction. Dligach, et al. pitted the Long Short Term memory (LSTM) units and Convolutional Neural Network (CNN) architectures against each other using the THYME corpus from the 2015 Clinical Temp Eval Challenge [11] for identifying EVENT-TIMEX and EVENT-EVENT intra-sentence temporal relationships. Input was minimally processed raw tokens and/or POS tags with XML markup of the temporal expressions and events. CNNs were shown to outperform the LSTM architecture for the Clinical Temp Eval 2015 challenge. However, shortly thereafter, Tourille, et al. [64] implemented a Bidirectional-LSTM (Bi-LSTM) that included character-level features for both intra- and inter-sentence temporal relations on the THYME data set with comparable performance on intra-sentence relations and superior performance with both inter- and intra-sentence temporal relations. Lin 2018 [65] took the BiLSTM from Tourille, et al. further by including a self-training RNN framework that utilized out-of-domain word embeddings to create a silver standard for training. This self-learning was also applied to the top performing, SVM-based THYME system. Lin, et al. discovered that SVM-based systems are unable to learn from self training using a silver standard and identified that the SVM was just calling more instances as positive. The self training did improve the RNN-biLSTM and RNN-GRU models, and including the out-of-domain features improved cross-domain performance as well.

However, Lin, et al. is still only using intra-sentence relations, and results were not very different from Tourille, et al.

2.5.4.3 The Era of BERT

In 2018, researchers from Google AI introduced Bidirectional Encoding Representations from Transformers (BERT) [56] as a new pre-trained language representation model (see Section 2.7 for an introduction). Representational Learning in NLP aims to learn informative numerical representations of words. These learned representations can take the place of, or augment, manually defined features that are fed into machine and deep learning models for prediction tasks [66]. Since the debut of BERT, it has been used in many NLP applications, including in the area of temporal reasoning. In 2019, Lin, et al. [67] converted their Recurrent Neural Network (RNN) self-training system [65] to utilize BERT and account for cross sentence relations by implementing a window approach instead of sentence by sentence. The BERT base was fine-tuned on MIMIC [68] and PubMed, and self-training using silver standard annotations was implemented similar to their work in 2018 [65] except with the window method instead of sentence-based. Improvements in performance, however, were mild, increasing the F1 score on the colon cancer test set to 0.684 from the previous best of 0.629. Lin, et al. also assessed the performance on the development set of the inter-sentence relations to get an F1 at 0.33, but was not able to surpass the performance of Tourille, et al. [64] with an F1 of 0.482, indicating much work is still needed in this area. Interestingly, they found that the window-based method does not work well with the previous BiLSTM method due to the unique characteristics of BiLSTMs. One of the disadvantages of the BERT model is its computational complexity due to having to encode the same sequence n x (n-1)/2 times. In 2020, Lin, et al. [69] converted this method to use a one-pass encoding mechanism inspired by Wang, et al. [70], which achieves the same performance but with a significant reduction in computational complexity, reducing the training time by several hours to days depending on the data set size. Finally, in 2020 Dupuis, et al. [71] utilized a clinically fine-tuned BERT model [72] to classify the anchor time relation type on a subset of relative and incomplete TIMEXs [15], but did not surpass the results from Sun et al. [15], which utilized classic SVM classifiers with bag-of-word features as input. Thus, utilizing BERT for clinical temporal tasks has proven to be a challenge.

Overall, while there has been progress in identifying temporal relations from refining focus to explicit relations, to implementing neural networks, and using information contained in deep neural networks such as BERT, there is still work to be done. Strategies to improve performance in the area of inter-sentence relations and relations involving implicit temporal expressions are needed.

2.5.5 Clinical Event Co-Reference Resolution

Coming from a non-linguistics background, deciphering the precise meaning of the term *co-reference* is surprisingly challenging as its definition is intertwined with the concept of *anaphora* in the NLP and computational linguistics literature. Recent reviews on the topic state that anaphora and co-reference resolution are two distinct yet overlapping tasks in NLP [51, 73]. In the following subsections the concepts of co-reference and anaphora from an NLP and linguistics point of view will be briefly reviewed followed by an explanation of how co-reference resolution is important in clinical timeline generation, and the progress of co-reference resolution in the clinical NLP domain. For a more in-depth discussion of Co-Reference Resolution and Anaphora, please read Sukthanker, et al.[73] and/or Tourille, et al. [51] as this level of detail is out of scope for this work.

2.5.5.1 Anaphora vs Co-Reference

In linguistics, an *anaphore* is a word or phrase where prior contextual knowledge is needed for correct interpretation. Resolving anaphoric expressions relies on previously introduced entities or concepts within the same narrative text, and requires little global or outside knowledge [73]. Examples 1 and 2[1] show two types of anaphoric expressions, where the anaphore is *italicized* and the antecedent or anchor is in **bold**

Sally left the theater, then *she* got dinner. (1)

Every speaker had to present *his* paper. (2)

Example 1 demonstrates a specific type of anaphore called a *referring anaphoric expression.* A referring anaphoric expression is a phrase or sentence where the anaphore is referring to a previously introduced entity (the antecedent/anchor), and the relationship is that of identity [51] (i.e. they are both referring to the same physical entity). This is referred to as "co-reference" in linguistics. Referring noun phrases that include pronouns, nominals, and proper names are common examples of referring anaphoric expressions. In Example 1, one must have read the first part of the sentence to understand to whom "she" is referring. Thus, in linguistics the task of *co-reference resolution* refers to the resolution of *referring anaphoric expression.* Note that resolving Example 2 is not co-reference resolution, but rather anaphoric resolution because the anaphore "his" in not equivalent to "Every speaker", but it's interpretation does depend on this antecedent. If these were equivalent, then the statement would read "Every speaker is responsible for presenting every speaker's paper.", which is not the correct interpretation (example taken from [73]).

[1] Example from [73]

The task of *co-reference resolution* in the NLP field overlaps that in linguistics, but it is not equivalent. NLP defines co-reference resolution as the task of finding equivalency classes among identified events or entities within a document, or across several related documents [74, 51, 73]. This is essentially event or entity normalization, and differs from the linguistics definition of co-reference and anaphora in a few ways. First, two identified entities can co-refer to the same physical entity but one does not have to be the antecedent of the other, and the expression does not have to be an anaphore. For example, noun phrases that use proper names can co-refer to the same person, but they are not anaphoric because you do not need the surrounding context for interpretation. Second, NLP co-reference resolution may require domain knowledge outside the local context for correct interpretation, unlike anaphoric resolution, which depends on the local narrative context. For example, when processing clinical records, reference to a patient's surgery may require global domain knowledge of the patient's medical history to know what type of surgery a phrase like "patient's surgery" is referring. The mentions of a surgery in one document co-refer to mentions of the same surgery in another document, which is not anaphoric. In this work we adopt the definitions of Sukthanker, et al. [73] and Tourille, et al. [51] for Anaphora and Co-Reference Resolution, where Anaphora Resolution is the task of identifying anaphores and their antecedents, and Co-Reference Resolution is the task of identifying equivalency classes among identified entities or events within a single, or across multiple, documents.

2.5.5.2 Progress in Clinical Event Co-Reference Resolution

Progress in the field of Clinical Event Co-Reference Resolution has stemmed from the general domain, but has been slow. In the general domain, several strategies have emerged to perform co-reference resolution, primarily focusing on noun and pronoun phrases within a single document: mention-pair models, entity-based models, mention-ranking models, and tree-based models. The *mention-pair strategy* is similar to the pairwise temporal relation extraction strategy where each entity (i.e. mention) in the text is paired with all other entity/event mentions creating an anaphore-antecedent candidate, and the relationship of whether or not they co-refer is determined. However, this strategy has many of the same pitfalls of the pairwise temporal relation task, including an unbalanced data set and the possibility for contradictions to arise due to each mention-pair being evaluated independently of all the others [75, 51]. Instead of looking at each anaphore-antecedent candidate individually, *entity-based models* attempt to group entity mentions with clusters of noun phrases that are likely to co-refer to the same entity creating a co-reference chain, which attempts to address the unbalanced training instance problem. However, both of these approaches still suffer from potential contradiction because they evaluate candidate pairs independently. The *mention-ranking model* attempts to resolve the issue of evaluating a pair of mentions independently by creating a ranked list with all possible antecedent candidates for a given anaphore and their probability. In this way one can consider all candidates at once to choose the best pair [75]. Finally, the *tree-based method* aims to build a dependency tree of entities where the root is the antecedent. This enforces that each mention/anaphore is only associated with one antecedent within a document [51]. All of these strategies have been applied to the general domain on a single document basis and assume there is narrative structure for extracting classification features. Ng, et al. [76] and Tourille, et al. [51] provide comprehensive reviews of the progress in general domain co-reference resolution.

In the clinical domain, Co-Reference Resolution includes the resolution of personal pronouns and other clinically relevant entities and events like problems, treatments, and tests. These types of entities can be difficult to resolve because very different lexical expressions can be used for the same entity or event, and identical lexical expressions can be used to reference different entities/events as they occurred at different times. Also, the wide variety of the ways problems, treatments, and tests can be expressed is much larger than the ways person references and pronouns are used. There was little progress in clinical co-reference Resolution until the 2011 i2b2 Co-Reference Challenge [77], which expanded the focus beyond noun phrases containing anaphores and antecedents to clinically relevant persons, entities, and events. Using discharge summaries, entities were defined as one of PERSON, PROBLEM, TREATMENT, or TEST. Entities could only be co-referential if they were the same entity type, which helped to resolve the negative bias problem for training data. For example, the entities "temporal artery biopsy" and "that testing" in Quotation 3[2] are co-referent of the same medical procedure, and thus belong to the same equivalency class, whereas the entity "she" clearly belongs to a different class and does not have to be evaluated with the medical procedure entities. This entity type restriction was used by many of the participating systems to build separate modules for the resolution of each entity type. Overall, 20 systems participated in the challenge, and consisted of rule-based, hybrid, and supervised learning approaches. Uzuner, et al. [77] notes that all systems, regardless of strategy, that incorporated external domain knowledge, such as the Unified Medical Language System (UMLS) [78] and Wikipedia, showed improved performance. Xu, et al. [79] built the top performing system which uses a binary SVM to identify all of the PERSON mentions from the PROBLEM, TREATMENT, and TEST mentions, then trains SVM classifiers using a large list of contextual and world knowledge features to create PERSON co-reference chains (primarily relating to the patient), and a separate classifier to identify the other

[2] Example from [77]

types of co-reference chains. In the i2b2 challenge, Xu, et al. obtained an F1 score of 0.915, which is very good for this task. While the 2011 i2b2 Co-Reference Challenge encouraged progress in the area of clinical co-reference resolution, it focused on discharge summaries where clinical text is written in a more narrative fashion, and only required co-reference to be determined within a single document. Further work is needed to address co-reference resolution across multiple documents and in documents that have less of a narrative structure.

She was scheduled to receive a temporal artery biopsy,
but she never followed up on that testing. (3)

2.5.5.3 The Timeline Relation

With regards to timeline generation, co-reference resolution is vital to building non-redundant timelines, especially when multiple related documents are processed. Clinical documents offer unique challenges to the task of co-reference resolution as compared to general domain texts. For example, clinical documents are highly repetitive and redundant due to the copying and pasting of content from previous documents or entries by clinicians to ensure the most relevant information is easily found [80, 54]. This creates a greater number of entities that must be placed in an equivalence class. Also, the narrative structure of clinical documents is limited and frequently not present due to domain knowledge of how notes are written and should be interpreted, and clinician shorthand. Both of which can change across different clinicians, departments, and medical facilities. This means the same event or entity could be referenced in multiple ways that can differ dramatically across documents such as a patient's history to a radiology report that both refer to a similar medical event. On the flip side, event mentions that have identical lexical and syntactic properties could be referring to completely different real-word events; thus, context and lexical-syntactic properties alone are insufficient to determine if the events are equivalent. Finally, co-references can not only span sentences, but also document sections and across different documents in the patient's medical record. This results in different contexts surrounding each mention of an entity or event that need to be normalized and resolved before the entity or event can be placed in an equivalence class and then properly placed on a timeline.

Until recently, co-reference resolution had been performed without consideration of the temporal information. A few systems in the 2011 i2b2 challenge utilized times and dates as features to help distinguish events with similar contextual, lexical, and/or syntactical properties [79, 81, 82]. However, recent progress in Clinical Event Co-Reference Resolution has determined that temporal information is vital to resolving event co-references both within a document and across multiple documents since event mentions can only co-refer if they happened in the same time frame [83, 51, 84, 49]. Tourille [51] even argues that in the task of extracting a clinical timeline, event co-reference and temporal information extraction should be performed jointly as they are complementary and connected.

Since the 2011 i2b2 Co-Reference Challenge, only a few papers have tackled incorporating co-reference resolution into timeline extraction. In 2014 Raghavan, et al. [83] addressed the task of identifying co-referring entities/events across multiple documents by modeling it as a multiple sequence alignment. Clinical events from each document were ordered into a sequence, then contextual and temporal features were used to align events across documents. To this day, Raghavan, et al. is this only one to address cross-document co-reference resolution. Around the same time, Dehghan, et al. [85] performs event co-reference resolution by using the lexical similarity of the context surrounding the event using a SoftTFIDF score. Any event pair score that is greater than 0.8 is assigned a new TLINK with the category Overlap, then a transitive closure of all TLINKS is performed. Both of these methods integrate the event co-reference resolution task with the identification of temporal relations, and they do not consider co-reference relationships in the PERSON category as defined by the 2011 i2b2 challenge. As the main purpose is to build a timeline of events, PERSON co-references are irrelevant in this context as we are only concerned with medical events that happened to the patient. Finally, the most recent work that integrates temporal information into clinical co-reference resolution is Tourille, et al. [51] where an event's temporal relation to the document creation time is included as a feature for a neural entity-based mention-ranking method.

It is clear from the limited set of published works since 2011 that clinical event co-reference resolution is far from a solved problem. With the goal of clinical event timeline extraction, co-reference resolution must be able to identify equivalence classes of events across multiple documents written in different styles and at different times over a patient's medical history. As of yet, only one group has attempted cross-document co-reference resolution [83], leaving much to be explored in this area. It is also clear that accurate clinical co-reference resolution requires the integration of temporal information. The type of temporal information that is the most useful (i.e. specific dates/time, relative relations, or relationships to a document creation time) is up for debate, but current evidence indicates that good clinical co-reference resolution needs temporal information and complete timeline extraction needs clinical co-reference resolution.

2.5.6 Temporal Event Ordering

The task of temporal ordering sounds, well, easy. Given explicit dates it is a simple matter of placing events chronologically on a timeline to induce order. However, when dealing with non-explicit, fuzzy, and relative temporal expressions, inducing order is a bit more difficult. The addition of timelines from multiple documents or data sources makes ordering events even more challenging, and hinges on the ability to perform event co-reference resolution within and across documents. In this work, we define the task of *temporal event ordering* as the problem of identifying the order of events without encountering any sequencing conflicts. When a set of temporal relations are viewed as a directed graph, a *sequencing conflict* occurs when a cycle is created; thus, any valid set of temporal relations should form a directed acyclic graph. This problem is closely related to identifying temporal relations, and in fact is a direct result of modeling the temporal relation task as a pair-wise classification task.

Many temporal relation algorithms are based on supervised machine learning approaches that model the relation task as a pair-wise classification problem, where the algorithm must classify relations between all pairs of events and temporal expressions, regardless of whether or not they are actually related [37, 44, 59, 60, 61, 62, 31]. Some try to reduce the number of relations by filtering to candidate relations [58], but this still leaves a lot of relations for classification. These algorithms do not consider ordering of events when assigning relationship types like BEFORE or AFTER. Thus, when you try to put these events in order on a timeline you could run into an ordering cycle that need resolution. For example, suppose our temporal relation algorithm identified the following relations between events A, B, and C: A before B, B before C, and C before A. Through inference of the first 2 relations, we can infer A happened before C; however, the direct relation "C before A" states the opposite. Now the question is, which is right? When did event A happen with respect to event C? Resolving these cycles is part of the Event Ordering problem.

To solve the ordering cycle paradox, Bramsen, et al. [86] modeled pair-wise relations as an acyclic directed graph (DAG). To ensure no ordering cycles were created, they implemented a greedy strategy by retaining only the relations with the highest confidence from their relation classifier. After each new relationship is added, the DAG is expanded using transitive closure. If a cycle is detected, only the highest confidence relations are kept to maintain the DAG. While this strategy may not find a global maximum, the authors claim it is close enough. However, no attempt is made to see if this resolution strategy generates clinically correct timelines.

Raghavan, et al. [87] takes a step back and looks at classifying events into course-grained time-bins based on their relation to the admission time (way before admission, before admission, on admission, after admission, after discharge) to create a course-grain ordering of events. A Conditional Random Field (CRF) is trained on temporal expression and narrative structure features such as sections, contextual bigrams, dictionary features, and explicit dates within the same sentences to place events into these time bins, which can then be used to identify a more fine-grained ordering of events if needed. Sarkar, et al. [46] take a similar approach to order events in Medical Case Reports where explicit temporal expressions are sparse, so one has to rely on relative temporal expressions. They first identify temporal breaks (i.e. current, past, future) in the narrative using a CRF. Ordering of events is then rule-based and is performed for each of the temporal contexts by ordering the sentences instead of individual events. Sarkar, et al. is the first to explicitly focus on relative temporal expressions for event ordering in the clinical domain.

Modeling the temporal relation identification task as a pair-wise classification problem results in inconsistencies, like ordering cycles, that need to be handled prior to creating a useful timeline. It also requires processing n-squared relationships where most have unknown relations or are not related at all, creating an unbalanced classification problem. Jeblee, et al. [88] approaches the problem of temporal relation identification through a different lens by forming it a list-wise ranking problem instead of a pair-wise classification problem. They argue this method alleviates the temporal inconsistencies from pair-wise methods and is easier for humans to evaluate. A linear neural network is used with contextual and event attributes as input features to rank events in a list-wise fashion. Viewing the temporal relation problem as a ranked list has the potential to more easily capture relative event relations as well, but this aspect was not addressed in this work and is noted as future work by the authors.

Temporal event ordering in the clinical domain is tightly associated with temporal relation extraction, but has not been paid much attention in the Clinical NLP field. Temporal inconsistencies are introduced, computational complexity increased, and human evaluation is difficult when modeling the temporal relationship identification task as a pair-wise classification problem. Modeling the temporal relation task as a list-wise event ranking problem resolves the issue with temporal inconsistencies and is easier for humans to comprehend and evaluate; however, current progress only uses simple relations and cannot model more complex or fine-grained event relations. Thus, event ordering is still an area of needed research and is tightly tied to the strategies used for temporal relation identification and classification. Methods are needed that can reduce the computational complexity of relationships identification and provide ordered sequences of events that are detailed and easily assimilated by medical professionals.

2.5.7 Timeline Visualization

Visualizing longitudinal clinical data has been shown to reduce the time it takes for a medical professional to assimilate a patient's health status and identify new insights that can better inform care decisions [3]. Timelines are most commonly visualized on a straight line with single points indicating when certain events occurred. However, medical timelines are more complicated due to the multiple layers of information that need to be represented, such as patient symptoms, procedures, and medications. In addition, durations of symptoms and medications are important to accurately visualize and cannot be represented as a single point on a line. While we may develop a highly sophisticated and accurate medical timeline extraction tool, it will be useless to clinicians in the field unless the information it has extracted can be graphically visualized in such a way that medical professionals can easily navigate events and quickly process and interpret the information to gain new insights about a patient's condition.

Medical timeline visualization is an active research field. In 2004, Martins, et al. introduced KNAVE-II [89], a tool built on a distributed framework for visualization of and interaction with time-oriented clinical data. Pulling data from multiple clinical data sources and a medical knowledge base, KNAVE-II aids clinicians in answering time-oriented clinical questions about a patient. Timeline was developed by Bui, et al. [90] and uses EHR structured data from multiple sources to provide a longitudinal, problem-centric view (such as cancer) of a patient's clinical data using multiple tracks for different data types. HARVEST [91] primarily utilizes the admission and discharge dates of patient visits to build a timeline that is displayed as a single linear track; however, it also analyzes clinical notes using NLP methods by normalizing disorders to UMLS (Unified Medical Language System) concepts [78] followed by a topic analysis with a TF-IDF (Term Frequency-Inverse Document Frequency) matrix to identify the most informative disorders in a patient's record compared to all other patients. Glicksberg, et al. [92] argued that the interoperability between developed timeline tools and EHR or data warehouse systems is a big problem for deploying a tool to multiple health care systems. They propose using a Common Data Model, such as

OMOP[3], as the backdrop to a timeline tool, and developed a simplistic timeline visualization based off of structured EHR data. While many of the timeline visualizations only focus on a single patient, Kilmov, et al. [93] expanded the KNAVE-II system to visualize and perform temporal reasoning over multiple patients for analysis and comparison. Likewise, Gotz, et al. [94] developed a tool to aid clinical researchers in identifying temporal patterns present in multiple patients, where users define the concepts and outcomes they are looking for and then the patient's matching those with the timeline of relevant events is visualized for exploration and analysis.

All of the aforementioned tools are built to utilize EHR or data warehouse structured data and do not incorporate information in clinical notes that could be extracted by NLP. This means there may be a lot more clinically relevant information present in an EHR that is not easily accessible by medical professionals, even with current timeline visualization tools. Park, et al. [95] argue that context and causality of events are also important to represent on a timeline. They propose V-model, a temporal information visualization tool that can effectively visualize event causality, non-explicit temporal information, and multiple levels of temporal granularity for events. V-model exclusively uses clinical narrative texts to summarize patient histories on a timeline; however, it has only been evaluated on a small cohort of single documents.

While there are a variety of clinical timeline visualization tools available, there is still a need for an inter-operable tool that can integrate multiple data types, including unstructured text, originating from multiple clinical systems into a comprehensible and intuitive visualization. This tool would need to be flexible, allowing for a global or problem-centric view of a patient record. The field is making progress on this front using structured EHR data; however, the incorporation of information for unstruc-

[3] https://ohdsi.github.io/CommonDataModel/

tured texts is lacking.

2.6 End-to-End Timeline Extraction Systems

In this section we review end-to-end clinical timeline extraction systems currently published in the literature. We classify a system as an end-to-end timeline extraction system if it includes event identification, temporal expression extraction and normalization, temporal relationship extraction, and either event ordering or event co-reference resolution for either a single document or over multiple documents. Any system that did not include methods or a tool to perform each step was not considered as end-to-end. This includes systems that used manually annotated temporal expressions or events [57, 83], or those that used a pre-annotated gold standard to obtain these entities [96, 97]. It also includes systems that only utilize the structured temporal data in the EHR instead of extracting it from clinical notes [98, 99, 100, 101], and all timeline extraction systems that only deal with general domain text [102, 103, 104, 105]. Additional timeline extraction steps considered as a bonus are timeline visualization and performance evaluation (see Section 2.6.2 below).

2.6.1 End-to-End System Review

Four published systems were found that met the end-to-end criteria for clinical timeline extraction (Table 4) [106, 107, 49, 48]. Three of these systems use rule-based components [106, 107, 48], with the other using a hybrid approach by incorporating machine learning into the event and temporal expression recognition, and co-reference resolution steps [49]. Each system takes a slightly different approach to timeline extraction that is influenced by the type of timeline needing to be identified, such as differing definitions of what a medical event is, the type of temporal expressions targeted, the number of documents needing to be reasoned over, and the underlying framework.

Zhou, et al. [106] and Jung, et al. [107] were the first to develop end-to-end timeline extraction systems that were both extensions of previously built NLP systems. In 2005, Zhou, et al. published the earliest complete timeline extraction system to integrate temporal information from clinical narrative reports into the existing MedLEE system. MedLEE is a rule-based NLP system that incorporates linguistic and clinical domain knowledge (e.g. UMLS, SNOMED, etc.) to annotate and structure clinical information into a frame-based representation. MedLEE's temporal tagger extracts temporal information into a Temporal Constraint Structure that defines the beginning and end of events. This data is then fed into the main NLP system to link events to temporal information. In post-processing the structured information and rules of discourse are used to model the timeline of events in a clinical narrative as a Simple Temporal Problem (STP). STPs are represented as a constraint graph, which effectively orders the clinical events based on their temporal constraints, and allows for non-explicit temporal relations to be inferred. At the time of publication, only a few basic rules had been implemented to perform co-reference resolution across multiple documents. MedLEE was designed to capture all medical events and thus, does not revolve around a specific use-case.

The next complete timeline extraction NLP system did not surface until 2011. Jung, et al. [107] extended a general purpose NLP system [108] using deep natural language understanding (NLU), to extract medical concepts and related temporal information from cancer patient consultation notes. The core NLU system relied on previously developed, sophisticated parsers that employed several statistical and symbolic sub-components, like POS tagging, and output information in Logical Form (LF), which is a frame-like semantic representation of the parsed text. Clinical domain and ontological knowledge was integrated into the LF representation from SNOMED and the UMLS. Clinical concepts, events, and temporal expressions were extracted using manually defined LF pattern-based rules. Relationship identification between events and concepts (where an event can contain many concepts) and between events and temporal expressions is rule-based and derived from the LF graph. Referential temporal expressions are resolved using LF information or the document time, and negations are considered to determine if a concept is present or not in the patient's medical history prior to visualization with the Simile Timeline Widget.

The systems built by both Zhou, et al. and Jung, et al. were intended to be general purpose timeline extraction systems for clinical data, and both only focused on *single document narrative texts*. Thus, the methodology of each was geared towards parsing discourse structure, which may not be applicable to the often fragmented, highly specialized, and diverse types of clinical notes commonly found in the EHR. While Zhou, et al. implemented some basic co-reference resolution across multiple documents, Jung, et al. does not consider this step. Additionally, both systems only focus on using explicit temporal expressions, and neither system implements a metric for a formal performance evaluation.

In 2014, Deghan [49] developed a clinical timeline extraction system, referred to as "mining patient journeys", that addressed some of the short-comings of the systems from Zheng et al. and Jung et al. Namely, Deghan utilized multiple narrative documents (clinical and patient narratives), performed co-reference resolution across a patient record (i.e. across multiple documents), and evaluated the system's performance, both on each individual component as well as for the resulting timeline. However, the system developed by Deghan was built for the specific use case of identifying treatment timelines for survivors of childhood brain cancer.

Briefly, Dehghan used a hybrid approach with the EVENT, TIMEX, and TLINK components trained on the 2012 i2b2 standard. EVENTs included the i2b2 problems, treatments, and tests, in addition to health-related quality of life (HrQoL) concepts that were relevant to survivors of childhood brain cancer. A hybrid approach was taken to identify EVENTs and TIMEXs using Conditional Random Field models with lexical, syntactic, and orthographic features as input, along with rule-based approaches, such as regular expressions for formatted dates/times and dictionary lookup for HrQoL concepts. All EVENTs identified were mapped to UMLS concepts in order to utilize the knowledge base to easily identify high-level concept categories, which are used in co-reference resolution and timeline evaluation. An initial set of TLINKs were identified with a rule-base approach, then SputLink [109] was used to perform a transitive closure to identify all implied relationships, and to resolve any inconsistencies in temporal ordering. Only TLINKs that could be resolved to a point on a timeline were used in the final timeline construction phase (i.e. all relative or non-anchored TLINKs were removed). Co-Reference resolution was performed within each document and across multiple documents in a patient's record. The intra-document co-reference resolution was based only on lexical similarity by using a SoftTFIDF score [110] on all candidate event pairs within a single document. Inter-document co-reference resolution was performed in a similar manner, but was limited to events assigned to high-level categories related to the specific use-case, such as surgery or radiotherapy. Finally, event ordering and visualization were done using applications referred to as "PathCluster" and "PathVisualization", which grouped events into high-level processes and assigned them to a time bin with a 6-month interval.

Dehghan's approach to extract patient journeys is based on abstraction of low-level details to high-level processes and concepts. Ultimately, EVENTs are abstracted to higher-level concepts using the UMLS, and are then binned into 6-month intervals. The 6-month interval was chosen based on the use case as 6-months was the approximate follow-up time for the patient's under study. Thus, within a 6-month time bucket, the exact order of events doesn't matter. Using a manually generated gold standard, evaluation of the full timeline at the abstracted, high-level (i.e. "Oncology Treatment", "Oncology Diagnosis", etc.) resulted in 100% Precision and Recall; however, this high level information is not helpful to a treating physician. When details, such as type of chemotherapy, were required to match the gold standard timeline, performance decreased. While Dehghan's system performs well for a specific use case, it still relies on narrative texts, and much of the co-reference resolution and performance evaluation are dependant on the high-level concepts defined by the specific use case. Additionally, the 6-month time-frame for binning events may not work well with diseases that are faster moving and have densely populated clinical notes. This system would most likely require detailed manual intervention in order for it to extract a timeline for a different type of disease or medical use-case.

The most recent clinical timeline extraction system is by Najafabadi, et al. in 2020 [48], and contains mostly rule-based components built over the UIMA framework [111]. This system was designed to extract the evolution of a lung cancer patient's health status over time starting at the date of diagnosis, and is the first timeline extraction system for Spanish clinical notes. Najafabadipour, et al. define EVENTs as diagnosis concepts and tumor stages, and extracts them using the C-liKES system [112] and TNM Annotator [113], respectively. TIMEXs are identified and normalized using the rule-based Temporal Tagger tool [114]. These annotations are then input into the Temporal Reasoning System, which identifies TLINKs using the UDPipe tool [115] to first build a dependency parse tree using a simple single layer neural network followed by a set of rules using syntax dependencies to determine if an EVENT is linked to a TIMEX based on the parse tree. The annotated EVENTs, TIMEXs, and TLINKs are stored in a structured database, which is used as input to the Timeline Constructor step. EVENT ordering and co-reference resolution are carried out during timeline construction. Capturing the evolution of changing health status requires a slightly different approach as you need to capture the change in *status* of a medical concept (e.g. tumor stage) that is associated with clinical events. This led Najafabadipour, et al. to develop a unique inter-document co-reference resolution method that required 2-3 criteria be met for two events to be termed co-referential. First, events needed to be semantically similar. Unlike Deghan [49] who only used lexical cues, Najafabadipour, et al. looks at the similarity between the assigned semantic classification of events. Second, the associated time for an event should be the same or contiguous after sorting. Finally, if an event is associated with a value (like tumor stage), a third criteria must be met where co-referential events must have the same value. This definition identifies continuing states of a patient, designed to identify the evolution of the patient's health status over time. Once events are ordered and identified as co-referential, the earliest event time is kept for that event and propagated to all co-referring instances. System generated timelines were compared to over 800 manually, expert-curated timelines. Errors were generated under one of two condition: 1) the system identified a different number of events, or 2) events had different date expressions from those identified by the experts. Overall, the system performed well under these metrics, but could be improved through annotation of negations and probabilistic terms. Najafabadipour, et al.'s approach to timeline generation is specifically built for tracking the evolution of lung cancer patient status over time. This view on clinical timelines is novel and may make sense for many clinical conditions as a patient's status does change over time, and that information can be highly informative.

From the four end-to-end timeline extraction systems we can see that the methodology is influenced by the assumed type of text being processed and the use-case that is utilized. Because a patient's medical history is buried in multiple notes with multiple note types and grammar that is not always going to follow traditional rules, future timeline extraction systems should be flexible in processing this diverse data, as well as able to deal with the high level of redundancy in the EHR by integrating this data into a single contiguous timeline through robust co-reference resolution. In addition, timeline extraction systems need to focus on the ability to extract timelines relevant to any number of conditions or diseases that have any number of temporal pathways (e.g. a slowly progressing chronic non-terminal disease vs a quickly moving terminal disease vs a short-term illness such as a viral infection). The current tendency to focus on a single type of disease or cohort of patients, all with similar temporal progressions may lead to systems that do not generalize well for conditions with varying temporal properties. In addition, the extraction of temporal entities from clinical texts will generally be static across an EHR; however, the definition of a medical event needs to be standardized and applicable to all types of use cases, and robust methods to resolve cross-document event co-reference is needed. Finally, the timeline extraction systems discussed only utilize information available in the unstructured texts being processed, but the EHR has a wealth of structured information as well. Some timeline systems (not included in this review due to the absence of NLP methods) utilize this structured information to identify events and times for timeline construction (i.e. they ignore the textual data). Thus, future work is needed in integrating the structured timelines and timelines obtained from clinical texts to augment each other for a more complete and accurate representation of a patient's medical journey.

2.6.2 End-to-End Performance Evaluation

It is critically important that timeline extraction methods work well with high accuracy before it is implemented in the clinic and could influence patient care. Because end-to-end systems are built upon many other sub tasks, all sub tasks must

Table 4. End-to-End Clinical Timeline Extraction Systems

Tool:	**MedLEE [106]**	**Deep NLU [107]**	**Dehghan [49]**	**Najafabadipour [48]**
Year	2005	2011	2014	2020
TIMEX	Rule-based: Temporal Constraint Structure (TCS), explicit focus	Rule-based: Logic Form Parsing, explicit focus	Hybrid: CRF + Rules	Rule-based: Temporal Tagger
EVENT	Rule-based	Rule-based: LF Parsing	Hybrid: CRF + Rules	Rule-based: C-liKES and TNM Annotator
TLINK	Rule-based	Rule-based: LF Parsing	Rule-based	Hybrid: UDPipe + Rules
Event Normalization	Rule-based: absolute reference chosen over implicit	X	Rule-based	Rule-based: Semantic Similarity + Temporal Rules
Event Ordering	Temporal discourse, Simple Temporal Problems (STP) Graph	Assigns intervals to events	PathCluster	ule-based: Selects earliest occurrence of unique events, sorts temporally
Visualization	X	Simile Timeline Widget	PathVisualization	X
Evaluation	X	X	Precision and Recall	P,R,F1 for components, manual for timeline system
Scope	Single Document	Single Document	Multiple Documents	Multiple Documents

perform equally well before the end-to-end system will perform well. For example, Najafabadipour, et al. [48] noted that many of the errors incurred in timeline construction were the result of incorrect TLINKs generated in the prior step. Thus, evaluation of the individual parts of a timeline extraction system is vital; however, an overall evaluation of the completed timeline is also needed. Unfortunately, there are a limited number of strategies for evaluating end-to-end timeline extraction systems, and within those that do there is a lack of consensus on evaluation methods.

Of the four end-to-end timeline extraction systems discussed in Section 2.6.1 only two have any type of formal evaluation for the resulting timeline [49, 48]. In order to evaluate the final timeline, both of these researchers had to obtain manually, clinician-annotated timelines, which is highly time-consuming. Dehghan obtained three clinician-annotated patient timelines that started at the patient's diagnosis and continued for the first 42 months. These were used as the gold standard for comparison with the automatically generated timelines on two levels, first at a high level abstracted to the clinical process such as "Oncology Treatment", and then at a more detailed level that included information like the type of chemotherapy. Additionally, Dehghan binned events (both at the high and low levels of annotation) into 6-month periods where relative order within a window was ignored. Precision and Recall were calculated based on this information and errors were thrown when there were any information mismatches for a given time bin. Najafabadipour et al. was able to obtain over 800 clinician-annotated timelines for use as a gold standard, which also started at the date of patient diagnosis. Events are at a low level as compared to Dehghan, and errors were generated if 1) the system identified a different total number of events for a patient, or 2) the same event instance had different date expressions from those identified by the experts. Note that in both of these evaluation strategies, temporal order is either abstracted to bins or ignored all together. While not a complete end-to-end clinical timeline extraction system as defined in this work, Raghavan [116] took a different direction and implemented an edit distance metric based on the popular "word error rate" metric used in automated speech recognition to evaluate how far away an automatically generated timeline is from the gold standard. This method focuses on the sequence of events, and includes the adding, substituting, and deleting of ordered medical events; however, it does not utilize event properties or values such as tumor stage, and only looks at relative order while ignoring durations. An evaluation of a patient timeline should consider both content (to a level of detail useful to clinicians), event order, and event duration. Note that event order does not necessarily mean exact dates, but rather relative ordering. We exclude getting the dates exactly correct as this is a challenging task even for humans; however, the system should be able to infer relative order of unique events and assign a reasonable date of occurrence if one is not explicitly defined in the EHR.

2.7 Representational Learning

Representational Learning in NLP aims to learn informative numerical representations of words that are referred to as *word embeddings.* These learned representations can take the place of, or augment, manually defined features that are fed into machine learning models for prediction tasks [66]. The following subsections describe the difference between distributed versus contextualized embeddings, provide an overview of pre-trained language models, and dive into the details of how contextualized embeddings are generated using self-attention.

2.7.1 Distributed vs Contextualized Word Embedding Models

Classic distributed word embedding models, such as Word2Vec [117] and GLoVE [118], use the co-occurrence of words in an input corpus to form their representations. The resulting word embeddings depend on the usage of a given word in the corpus used for training. If a single word is used in multiple ways (i.e. an animal bat vs a baseball bat) in the training corpus, then those contexts will be incorporated into a single embedding for that word. Once the embedding is learned it is static and cannot be changed without re-training the entire model on a new corpus. Thus, distributed word embedding models generate a single embedding for each word that is static and could include information from multiple semantic spaces.

In contrast, a contextualized word embedding is one in which the context of a token is incorporated into the embedding vector; thus, a word can have slightly different embeddings depending on how it is used in a sentence. For example, the word “bank” can have multiple meanings depending on how it is used, such as in “The river bank was perfect for fishing.” or “I visited the bank to withdraw some money.” In these examples, the word “bank” would have slightly different embeddings that are reflective of the context.

2.7.2 Pre-Trained Language Models

Learning good and informative representations is a computationally expensive task and requires massive amounts of data, thus, there has been much attention paid to pre-trained models in recent years. Pre-training allows users to take advantage of the information in large corpora in the form of a pre-trained deep neural network with which they only have to fine-tune the top layer to obtain word embeddings specific for their task. It is like using a pre-made cake and just adding the icing versus having to bake the cake from scratch.

In 2018, researchers from Google AI introduced Bidirectional Encoding Representations from Transformers (BERT) [56] as a new pre-trained language representation model that generates contextualized word embeddings. BERT has taken the NLP community by storm, producing new state-of-the-art performances on many NLP tasks [56]. Other pre-trained language models, such as ELMo [119] and OpenAI GPT [120], also generate contextualized embeddings using a unidirectional model; however, BERT implements bidirectional representations where it is able to use the context both before and after a token and, thus, incorporates more information into its embedding. A major advantage of BERT is that it implements a "Masked Language Model" [56], which allows it to learn from unlabeled data. This means BERT models can learn from massive data sets without the need to create a manually generated gold standard.

There are four main advantages of utilizing fine-tuning on pre-trained models like BERT [121]. First, we can take advantage of quicker model development because the model weights already encode a lot of information, thus fine-tuning only needs to be run for 2-4 epochs for a specific task. Second, we can utilize much less training data compared to if we were building a model from scratch as the bulk of the heavy lifting has already been done. Third it has been shown that simply fine-tuning BERT to a specific task results in better or comparable performance compared to specialty models [56]. Finally, BERT is an example of transfer learning. Transfer learning is utilizing the knowledge learned from one domain and transferring it to perform similar tasks in another using a different data set[4] [122]. Thus, it is flexible in that the primary underlying neural network can be used as a starting place for many different tasks versus other models built specifically for one task.

BERT models are utilized in two different ways in the literature. The first is fine-tuning BERT's neural network to perform a specific classification task directly, such as language translation or question-answering [56]. The second is to extract the contextualized embeddings from the BERT model to be used as features in downstream applications, such as classical machine learning classifiers. This work utilizes both of these methods.

2.7.3 Tokenization: Whitespace vs. Word-Piece

Prior to generating any type of word embedding, a system first has to determine how to break up a sentence into individual semantic units, such as words. This process is called "tokenization" and each semantic unit is called a "token". The simplest, and most natural, way to tokenize a sentence is "whitespace tokenization" where spaces, tabs, and other whitespace characters are used as token delimiters (Figure 5A). Tokens that are part of a temporal phrase are generally defined using whitespace tokenization. However, this can introduce punctuation into tokens, such as periods at the end of a sentence, that are not part of the semantics of a particular

[4] https://www.analyticsvidhya.com/blog/2020/07/transfer-learning-for-nlp-fine-tuning-bert-for-text-classification/

token, or the phrase as a whole. Additionally, language models can't process a token if it is not in that model's vocabulary. These are termed out-of-vocabulary (OOV) tokens, and are generally ignored by the language model during analysis. To avoid this issue, BERT utilizes a "word-piece tokenization" model.

In word-piece tokenization, if a token is not found in the model vocabulary the token will be broken into the largest pieces possible that are in the vocabulary with the worst case scenario of having a token broken into individual characters. These are termed "subwords" where all but the first subword includes a double hash "##" as a prefix. Note that the vocabulary distinguishes between tokens such as "##bed" and "bed", thus each gets it's own embedding. In addition, along with whitespace, BERT also uses punctuation as token delimiters. Thus, all punctuation forms its own token in the BERT model (Figure 5B) and has its own contextualized embedding representation. For example, the date "2010-06-30" in Figure 5 is kept as one token with whitespace tokenization, but is broken up into five tokens using word-piece tokenization.

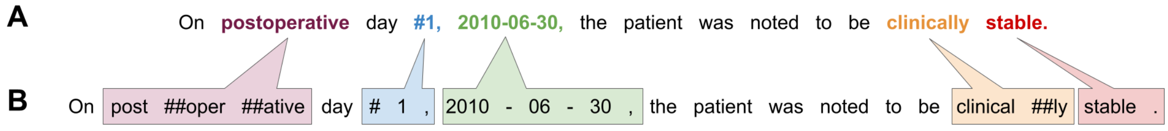

A
On postoperative day #1, 2010-06-30, the patient was noted to be clinically stable.
B
On post ##oper ##ative day # 1 , 2010 - 06 - 30 , the patient was noted to be clinical ##ly stable .

Fig. 5. Whitespace tokenization (A) vs. Word-piece tokenization (B).

2.7.4 Attending to Context

BERT creates contextualized embeddings through the implementation of a self-attention mechanism [123]. At a high level, self-attention integrates the embeddings of context words into a single embedding to represent a target word, thus incorporating context into the target word embedding to make it more representative of the semantics in the current sentence[5]. In BERT this works by utilizing a query Q and a set of key-value pairs (represented as K and V respectively) with length N, where N is the length of the input sentence. For example, say we want to create a contextualized embedding for the word "dog" in the sentence "The dog sat by the river bank."? The query Q is the word "dog", which is represented by an initial and static word embedding in the BERT model. The set of $K - V$ pairs are the initial static embeddings for all the words in the sentence, including the word "dog" itself. In this example, K equals V, but it doesn't necessarily have to. The scalar dot product of the current query Q is taken with all keys K to obtain another vector of length N (Figure 6A). By using the scalar dot product, words that are more similar to each other receive a higher value. This is repeated for all query words (i.e. all words in the sentence), which results in a matrix of scalar dot products (Figure 6B). This matrix is then scaled and run through a softmax function to obtain the final attention weight matrix where each row adds up to 1. This matrix is then used to weight each of the Value, V, vectors and linearly combine them to create the final contextualized word embedding for the query "dog" (Figure 6C).

The example in Figure 6 utilizes the full length of the original word embeddings to create the query and key-value pairs; however, in BERT, projections are used to convert the query and keys into vectors with length d_k ($d_k = 64$ in the "bert-base-uncased" model). These projected Q and K vectors are what is used for generating the attention weight matrix, which is then applied to the Values. Thus, unlike in Figure 6, the scalar dot product of the query vector for "dog" and the key vector for "dog" will not be equal to one because the word is being represented in different Linear Algebraic spaces. Equation 2.1 represents this process mathematically, where

[5] This video explains the process: https://peltarion.com/blog/data-science/self-attention-video

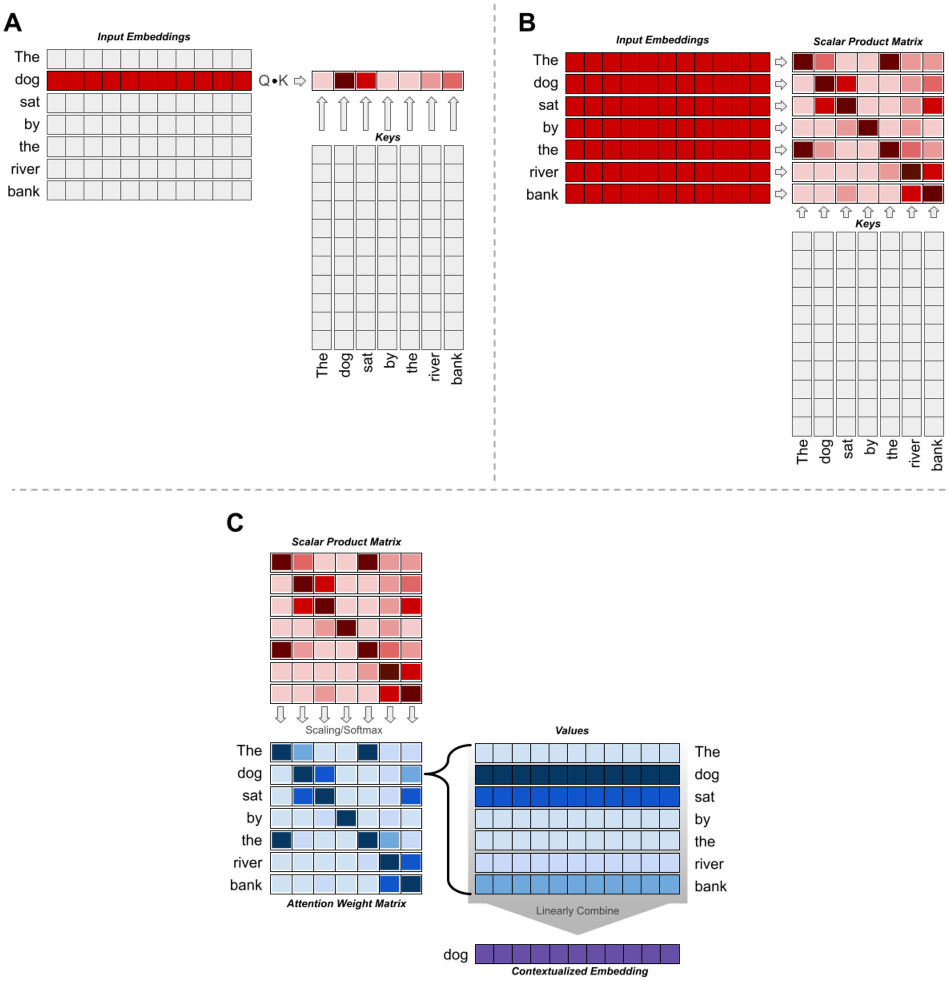


Fig. 6. Overview of Attention Basics

$\frac{1}{\sqrt{d_k}}$ is a scaling factor used to improve performance when d_k is large [123].

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{2.1}$$

Finally, BERT uses *multi-head attention* where each head, h, uses a different projection method allowing the Query and Key vectors to focus on different aspects of the sentence. Specifically, the “bert-base-uncased” model used in this work uses $h = 12$ attention heads; thus, Equation 2.1 is repeated 12 times, each using different Q and K projections (Figure 7A). The resulting contextualized embeddings from each attention head are simply concatenated to form the complete contextualized embedding (Figure 7B). In addition to multi-headed attention, BERT has 12 layers where each layer is composed of 12 attention heads. These layers incorporate additional linear projections, normalization, feed-forward layers, and positional information [56]. Thus, each word in the example sentence has 12 contextualized embeddings that are usually summarized, concatenated, or sub-sampled for use in downstream NLP pipelines (Figure 7C).

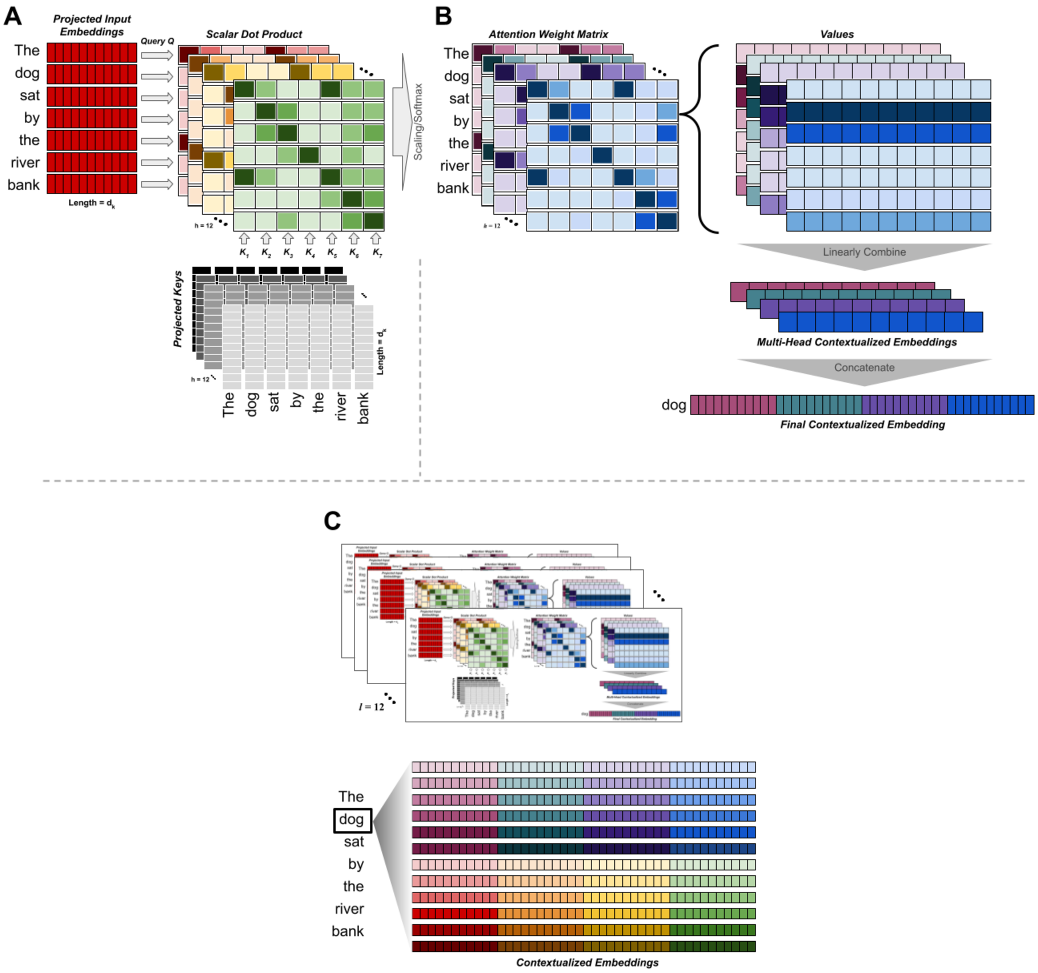


Fig. 7. Overview of Multi-Headed Attention with Layers

2.8 Evaluation Metrics

To assess performance of model predictions, this work reports the Precision, Recall, F1 Score, Accuracy, and Specificity (Equations 2.2-2.6) using the TIMEX type annotations. The Precision, Recall, and F1 are calculated in two ways in this work: 1) span-based and 2) class-based. Span-based is used when determining if the TERN system identified the correct span of text. This work uses the lenient definition where any overlap in span is considered correct. Results from Chapters 3-5 and the End-to-End results for the Phase 3 evaluation in Chapter 6 utilize the lenient span-based version of Precision, Recall, and F1 Score. Except for the Phase 3 evaluation, all of the metrics in Chapter 6 utilize the class-based calculations that are based off of identifying the correct temporal type for a given phrase. In most instances, the individual scores for each temporal type are summarized as a weighted average. Equation 2.7 shows the weighted average calculated across the DATE and DURATION temporal types, utilized in Chapter 6, where s is the metric score being averaged and w is the weight (i.e. number of instances for that temporal type).

$$Precision = \frac{TP}{TP + FP} \tag{2.2}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.3}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{2.4}$$

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{2.5}$$

$$Specificity = \frac{TN}{TN + FP} \tag{2.6}$$

$$WeightedAverage = \frac{(s_{DATE} * w_{DATE}) + (s_{DURATION} * w_{DURATION})}{w_{DATE} + w_{DURATION}} \tag{2.7}$$

2.9 Focus and Related Work

One of the first and most important steps in timeline extraction is the recognition of temporal expressions and their normalization to a computationally accessible form with the ultimate goal of identifying when events of interest happen and to place them sequentially on a timeline in order to perform temporal reasoning tasks. This work focuses on developing a TERN system tailored to the clinical domain that recognizes and normalizes TIMEXs to a form that is amiable to timeline generation. In the following chapters the construction, adaptation, and evaluation of Chrono, a hybrid rule-based and machine learning system developed to recognize and normalize temporal expressions into the Semantically Compositional Annotations for Temporal Expressions (SCATE) schema ([22]) is detailed. The SCATE scheme aims to improve upon the current TIMEX3/TimeML [17] standard by representing a wide variety of temporal expressions, allowing for events to act as anchors, and using mathematical operations over a timeline to define the semantics of each annotation. At this point, only one other system is known to parse temporal expressions into the SCATE schema, which is a character-based recurrent neural network implemented by Laparra et al. [124].

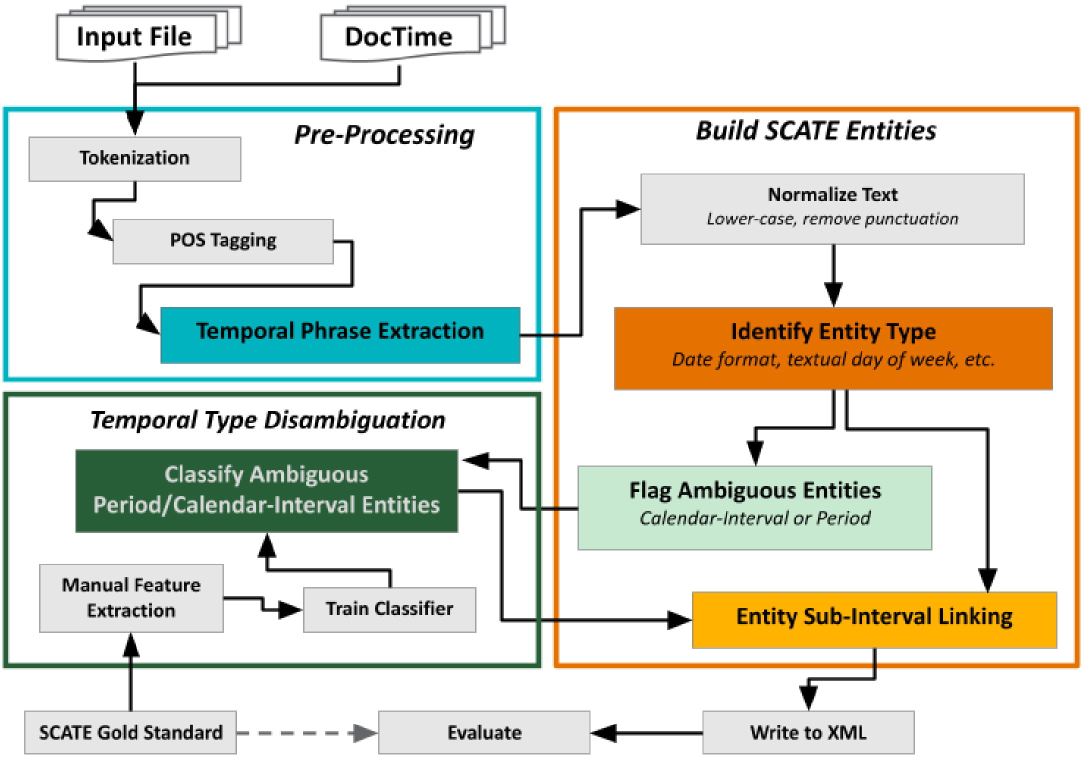
This work also narrows in on the overlooked task of determining the temporal type of RelIV-TIMEXs in order to ensure proper normalization. It is important to be able to accurately identify and normalize relative temporal expressions because they are ubiquitously used in clinical texts and important to the task of event ordering. There have been two recent works focused on the normalization of relative and incomplete temporal expressions [15, 71]. In 2015, Sun et al. [15] built SVM classifiers using contextual features to classify the type of anchor time (admission, discharge, previous TIMEX, previous absolute TIMEX) and anchor relation (before, after, equal/during). They utilized an adapted version of the rule-based general domain tagger Heidle time for TIMEX recognition and absolute TIMEX normalization, and achieved a statistically significant improvement over previous state-of-the-art methods. Additionally, in 2020 Dupuis et al. [71] utilized a clinically fine-tuned BERT model to classify the anchor time relation type on a subset of the Sun et al. relative TIMEX corpus, but did not surpass their results, and did not proceed to the normalization step. Both of these works are, to my knowledge, the only works focused on relative temporal expressions, and they both focused on the identification of anchor times and types. Additionally, all the given examples are of relative DATE phrases, not DURATIONs. Identifying the difference between DATE and DURATION temporal types is an important first step before the normalization. Thus, this work focuses on classifying RelIV-TIMEXs into either DATE or DURATION types, which is referred to as *Temporal Disambiguation* and detailed in the first part of Chapter 6. Finally, the rest of Chapter 6 describes the strategy and methods to create and utilize temporally fine-tuned contextualized word embeddings to perform the Temporal Disambiguation task and reports the performance results after integrating this module into Chrono.

CHAPTER 3

CHRONO: A TEMPORAL RECOGNITION AND NORMALIZATION TOOL

One of the first and most important steps in timeline extraction is the identification of temporal expressions and their normalization to a computationally accessible form. This chapter describes Chrono, a hybrid rule-based and machine learning system developed from scratch that identifies temporal expressions in text and normalizes them into the Semantically Compositional Annotations for Temporal Expressions (SCATE) schema developed by [22]. This scheme aims to improve upon the current TIMEX3/TimeML [17] standard by representing a wide variety of temporal expressions, allowing for events to act as anchors, and using mathematical operations over a timeline to define the semantics of each annotation, which makes it more applicable to generating timelines than other annotation schema.

Chrono was originally developed to participate in the SemEval 2018 Task 6: Parsing Time Normalizations challenge [41] on general domain texts (this chapter), however, it has been updated to also process clinical temporal data (Chapter 4), and to parse expressions into the main-stream TimeML schema (Chapter 5). Chrono has emerged as the top performing system for SemEval 2018 Task 6 for both general and clinical domain texts, and is shown to perform on par with the systems that participated in the 2012 i2b2 Temporal Challenge with minimal updates.



Input File
DocTime
Pre-Processing
Tokenization
POS Tagging
Temporal Phrase Extraction
Build SCATE Entities
Normalize Text
Lower-case, remove punctuation
Identify Entity Type
Date format, textual day of week, etc.
Temporal Type Disambiguation
Classify Ambiguous Period/Calendar-Interval Entities
Flag Ambiguous Entities
Calendar-Interval or Period
Manual Feature Extraction
Train Classifier
Entity Sub-Interval Linking
SCATE Gold Standard
Evaluate
Write to XML

Fig. 8. Overview of Chrono Workflow

3.1 Chrono System Description

Our approach to building this hybrid system includes four processing phases (Figure 8): 1) text pre-processing, 2) flagging numeric and temporal tokens, 3) temporal expression identification, and 4) SCATE normalization.

1) Text Pre-processing: Python's Natural Language Toolkit (NLTK) WhitespaceTokenizer and part-of-speech (POS) tagger [125] process raw text files to identify individual tokens, token spans, and POS tags. Punctuation is not handled at this phase as it is important for identifying correct spans.

2) Flagging Numeric and Temporal Tokens: All numeric tokens are flagged regardless of context. Subsequent phases utilize contextual information to determine if a numeric token is part of a temporal expression. Depending on the task, a rule may remove all or some punctuation, and/or convert tokens to lowercase prior to parsing. In the following, RP and LC denote **R**emoving all **P**unctuation and converting to **L**ower**C**ase, respectively.

Numeric Flagging: Tokens are flagged as numeric if either 1) the token has a POS tag of "CD" (Cardinal Number), or 2) the text can be converted to a numeric expression. Textual representations of numeric expressions are converted to numerics with the Word2Number[1] Python module. A custom method recognizes ordinals from "first" to "thirty-first" and converts them into the associated numerics 1 to 31, respectively. LC normalization is done prior to parsing textual numerics.

Temporal Flagging: Temporal tokens are flagged through rule-based parsing using lists of key words and regular expressions. This phase is more liberal in its identification of a temporal token than the SCATE normalization phase, so it identifies a broader range of potential temporal tokens that are refined in future steps. Tokens may be numeric and temporal simultaneously. Numeric tokens with the characters '$', '#', or '%' are NOT marked as temporal. The following types of tokens are flagged as temporal:

- Formatted date patterns using '/' or '-': mm/dd/yyyy, mm/dd/yy, yyyy/mm/dd, or yy/mm/dd
- Formatted time patterns matching hh:mm:ss
- Sequence of 4 to 8 consecutive digits matching range criteria for 24-hour times or for a year, month, and/or day (e.g. 1998 or 08241998).
- Spelled out month or abbreviation, e.g. "Mar." or "March", are flagged after

[1] https://github.com/akshaynagpal/w2n

RP except periods as they are required to retrieve correct spans.

- Days of the week, e.g. "Sat." or "Saturday", are parsed similar to months.
- Temporal words indicating periods of time, e.g. "yesterday" or "decade", are flagged after RP and LC.
- Mentions of AM and PM in any format are flagged after RP except periods.
- The parts of a week, e.g. "weekend" and "weekends", are flagged after RP and LC.
- Seasons of the year are flagged after RP and LC.
- Various parts of a day, e.g. "noon" or "morning", are flagged after RP and LC.
- Time zones are flagged after RP.
- Other temporal words, e.g. "this", "now", "nearly", and others, are flagged after RP and LC.

3) Temporal Expression Identification: A temporal expression is represented by a *temporal phrase*, which we define as two or more consecutive temporal/numeric tokens on the same line, or an isolated temporal token, with some exceptions. Figure 9 displays this process as a flow chart. Briefly, if a numeric token contains a '$', '#', or '%', or the text 'million', 'billion', or 'trillion' it is not included in a temporal phrase as these generally refer to non-temporal values. Additionally, isolated numeric tokens are not considered a temporal phrase.

4) SCATE Normalization: Chrono parses each temporal phrase into zero or more SCATE entities, links sub-intervals, and disambiguates the SCATE entities "Period" and "Calendar-Interval" via a machine learning module. Chrono implements

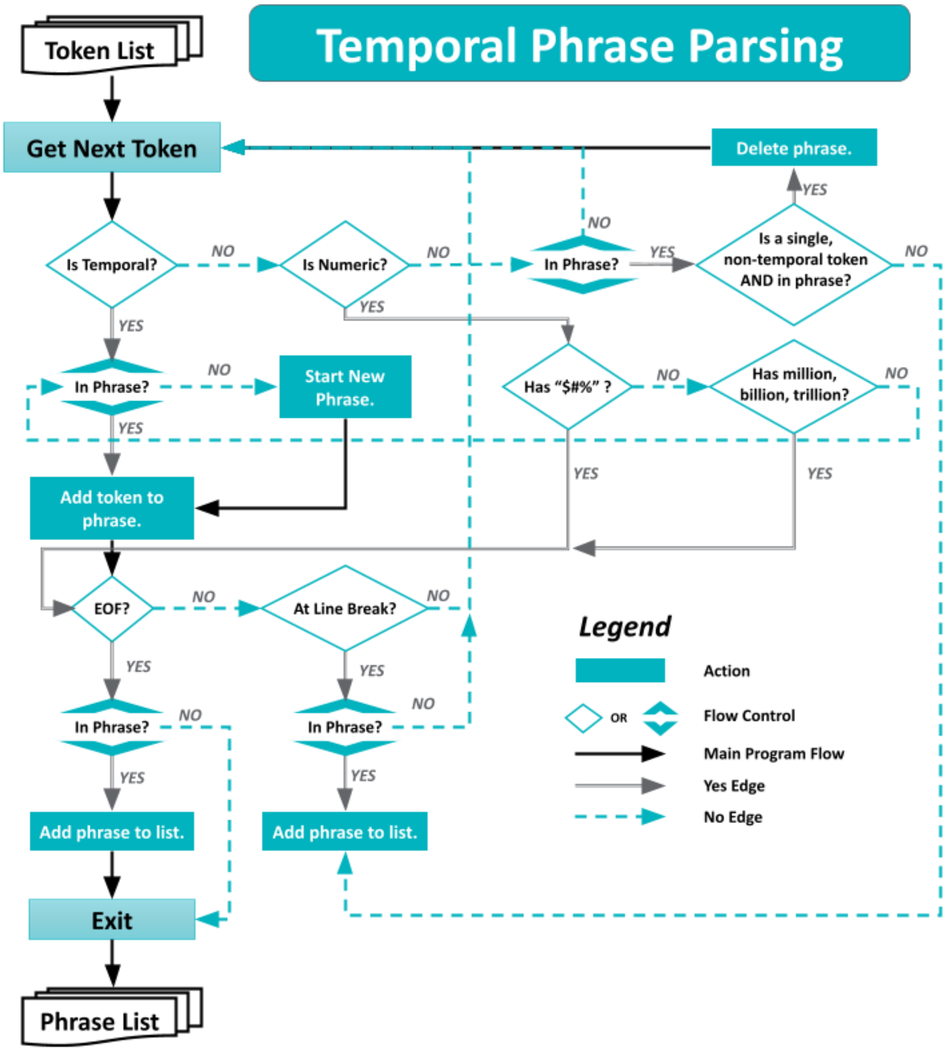


Fig. 9. Flow chart of Chrono's temporal phrase recognition algorithm.

32 types of entities with five parent types that have been described by [22]. In Chrono, entity types are parsed hierarchically, with certain types taking priority over another. For example a numeric date takes priority over a 24-hour time such that the phrase "1930" will be interpreted as a 4-digit year instead of the 24-hour time of 19:30 (i.e. 7:30pm). Figure 10 contains the priority of entity types implemented in Chrono. Parsing strategies also differ depending on the composition of a temporal phrase being

parsed. Each temporal phrase is interrogated sequentially by the following parsing strategies to identify the various elements in the phrase. Chrono assumes there is only one element of each type in a single phrase, and each token is assigned only one entity type plus a possible modifier type such as "Next" or "Last".

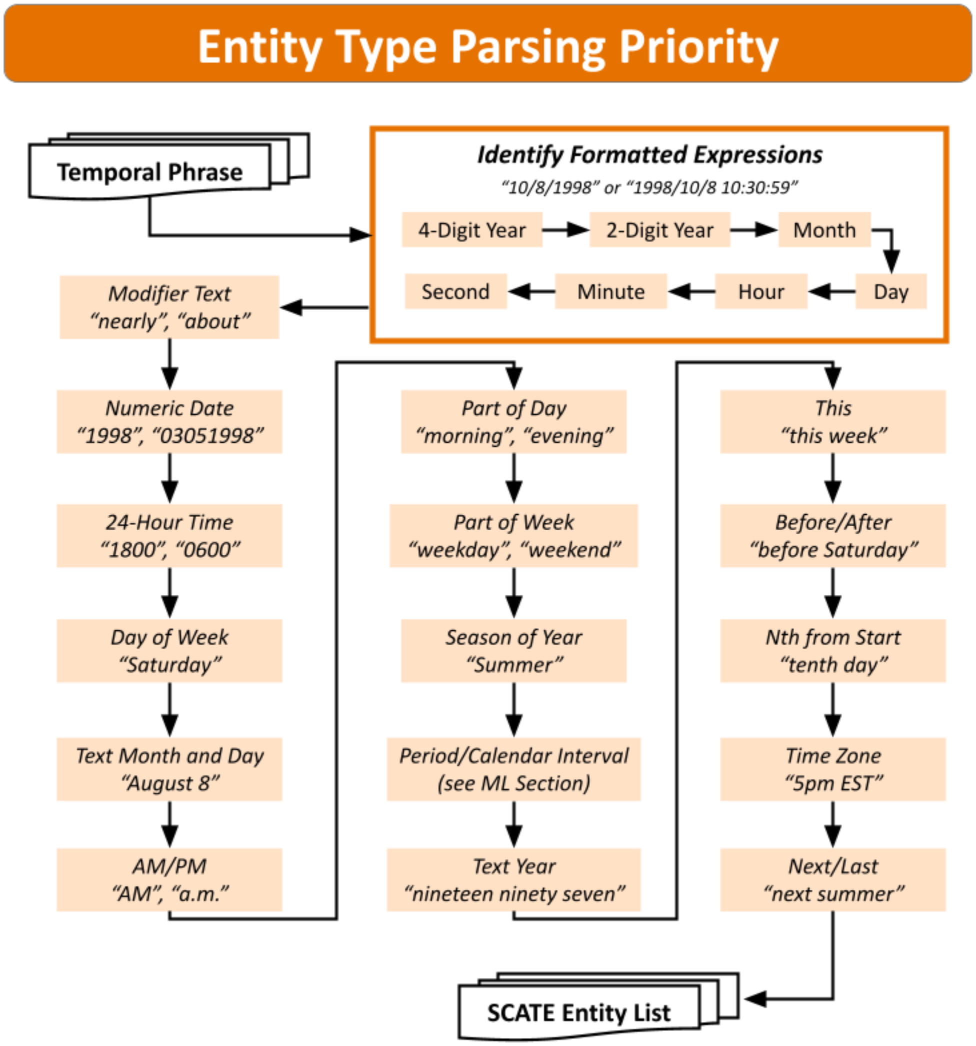


Fig. 10. Flow chart of the priority each entity type receives in Chrono.

Formatted Dates and Times: Formatted dates/times are parsed using regular expressions. To identify which format the date/time is in, Chrono looks for a 2-digit or 4-digit year first, then uses that position for orientation to identify the remaining elements. If a formatted date/time is identified, then the appropriate sub-intervals are linked during element parsing. 4-digit years take precedence over 2-digit years.

Numeric Dates and Times: Header and meta-data for Newswire articles frequently have numeric dates listed with no punctuation (e.g. “19980218” codes for “Feb, 18 1998”), and isolated 4-digit year mentions are frequent. After formatted dates and times are parsed, any phrase containing a numeric token is interrogated for a potential date or year mention. If a numeric token is 4-digits it is tested for a year between 1500 and 2050, 6-digit tokens are parsed for 2-digit year/month/day, and 8-digit strings are parsed for a 4-digit year and 2-digit month/day. All elements must be in the proper range, otherwise the token is skipped. Appropriate sub-intervals are linked during element parsing.

24-hour Time: 24-hour times are identified by either the format *hhmmzzz*, where *zzz* is the time zone, or a 4-digit number that has not been classified as a year. Hour digits must be less than 24 and minutes less than 60. Sub-intervals are linked at this time if existing. Time zones are handled separately and are linked back to the hour entity during the final sub-interval linking step.

Temporal Token Search: The majority of textual temporal entities are identified by looking for specific tokens. Token categories include days of the week, months of the year, parts of a day/week, time zones, and other temporal operators such as “early”, “this”, “before”, etc. Prior to looking for these tokens, text is normalized by RP and LC. Exceptions to RP include searching for day/month abbreviations, such as “Sat.” or “Aug.”. In these cases periods are not removed because they are part of the SCATE span. Another exception to RP and LC is identifying mentions of AM or PM where periods are kept and text is not converted to lowercase in order to capture variations like “PM” or “p.m.”. Non-temporal mentions of the months or seasons of the year “may”, “march”, “spring”, and “fall” are disambiguated using POS tags, where tokens that refer to a temporal entity generally have a POS tag of “NN” or “NP”. Sub-intervals are not linked during token searches.

Text Year: Another special case of parsing temporal tokens are textual representations of years such as “nineteen ninety-seven”. The Word2Number Python module was modified to recognize these phrases. Previously, it would add 19 and 97 together instead of returning 1997.

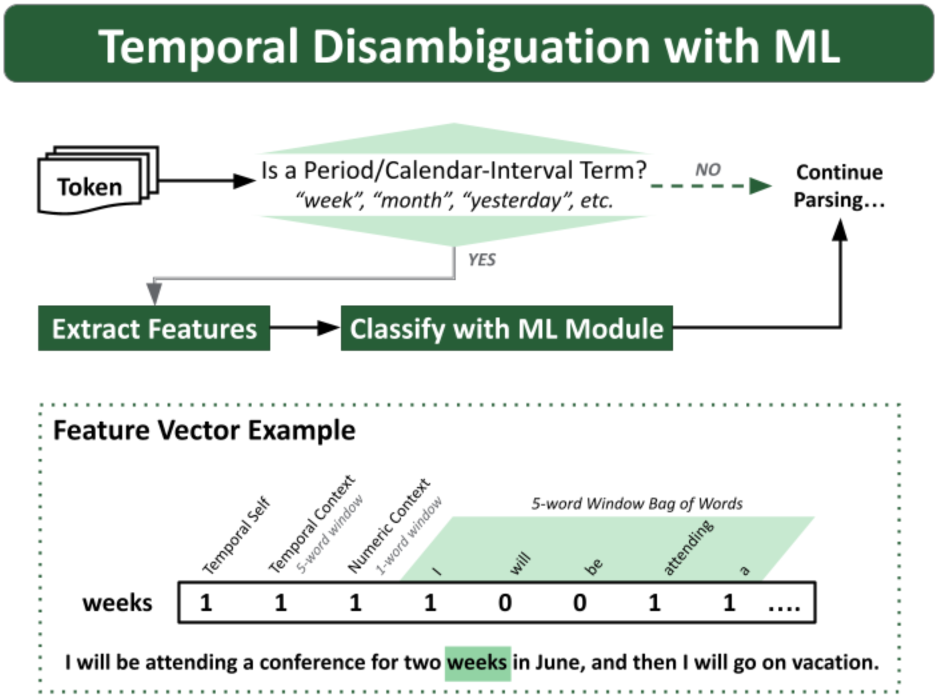
Periods and Calendar-Intervals: The same temporal token can refer to either a SCATE “Period” or “Calendar-Interval”. For example, in the phrases “in a week” vs “next week” the token “week” is classified differently. Due to language intricacies it is difficult to define a rule-base system to disambiguate these entities as the classification is contingent on the topic being discussed where phrasing around the entity can be different for each instance. Thus, Period/Calendar-Interval tokens are initially identified by a token search using a defined list of terms, then the identified term and its span are passed to a ML algorithm for classification.

Machine Learning Classification: Four ML algorithms are available in Chrono to differentiate between “Period” and “Calendar-Interval” entities using contextual information. Chrono implements Naive Bayes (NB), Neural Network (NN), Decision Tree (DT), and Support Vector Machine (SVM). Binary feature vectors (Figure 11) for all implementations have the following features:

- temporal_self: If the target is flagged as temporal, this feature is set to “1”.
- temporal_context: If there is at least one temporal word within a 5-word window up- or down-stream of the target this feature is set to “1”.
- numeric: If there is a numeric expression either directly before or after (a 1-word window) the target, this feature is set to “1”.

- context: All words within a 5-word window are identified as features and set to “1” if that word is present. Prior to identifying these features all words are normalized with RP and LC. The 5-word window includes crossing sentence boundaries before and after the target word.

We use NLTK with default parameters to implement NB and DT, NN is a simple feed-forward network with three hidden layers implemented using Python’s Keras package [2] with epochs set to 5 and batch set to 10, and SVM is implemented using SciKitLearn [126] with C set to 0.05 and max iterations set to 3.


Temporal Disambiguation with ML
Token
Is a Period/Calendar-Interval Term?
“week”, “month”, “yesterday”, etc.
NO
Continue Parsing...
YES
Extract Features
Classify with ML Module
Feature Vector Example
5-word Window Bag of Words

	Temporal Self	Temporal Context 5-word window	Numeric Context 1-word window	I	will	be	attending	a	
weeks	1	1	1	1	0	0	1	1	....

I will be attending a conference for two weeks in June, and then I will go on vacation.

Fig. 11. Example of the feature construction for Chrono’s temporal disambiguation module for Period and Calendar-Interval types.

Ordinals: Ordinals such as “first” or “3rd” are classified as an “NthFromStart”

[2] https://github.com/keras-team/keras

entity in the SCATE schema. These mentions are identified by normalizing with RP and LC before searching for the ordinal tokens representing the numbers 1-31.

Next/Last Parsing: Determining whether an entity is referring to a date in the future, "Next", or past, "Last", depends on context and the document time (doc-time). Next/Last parsing is done after all other parsing (first part of Figure 12), and checks two cases: 1) if a temporal phrase contains a year, no additional annotation is made, and 2) if specific modifier words are present (e.g. "next" or "last") immediately preceding a temporal expression, the modifier is annotated with a sub-interval referencing the following temporal entity. If neither of these cases hold, the year is set as the doc-time year, and the month and day are compared to the full doc-time to determine if it occurs before or after. Note the year assumption is not always valid and more complex, content-based parsing may be required to achieve higher Precision. Finally, if a day of the week (e.g. "Saturday") is mentioned, Chrono finds the first preceding verb in the sentence, and if it is past tense the temporal entity is annotated as "Last", otherwise it is annotated as "Next".

Sub-Interval Linking: After all SCATE entities are identified, all temporal phrases are re-parsed to identify sub-intervals within each phrase. For example, entities in the phrase "August 1998" are parsed by two different methods leaving the sub-interval link vacant. During sub-interval linking, the year "1998" has the "August" entity added as a sub-interval. Sub-interval linking reviews entities from the smallest to the largest, adding missing sub-intervals as needed. This method assumes each temporal phrase contains zero or one of each type of SCATE entity and is visualized as a flow chart in Figure 12.

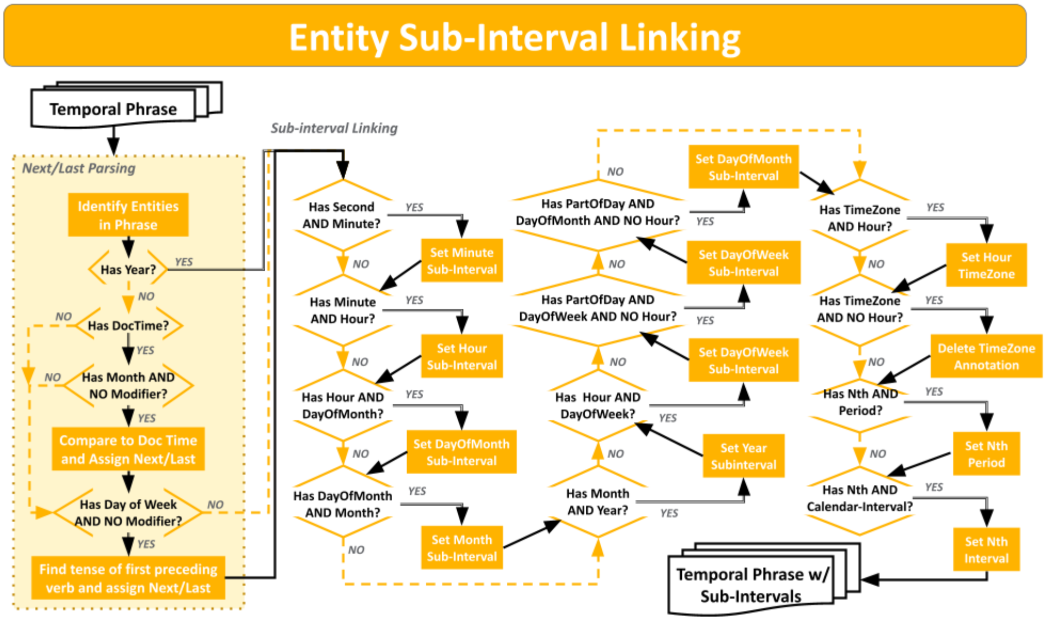

Entity Sub-Interval Linking
Temporal Phrase
Next/Last Parsing
Identify Entities in Phrase
Has Year?
YES
NO
Has DocTime?
NO
YES
Has Month AND NO Modifier?
NO
YES
Compare to Doc Time and Assign Next/Last
Has Day of Week AND NO Modifier?
NO
YES
Find tense of first preceding verb and assign Next/Last
Sub-interval Linking
Has Second AND Minute?
YES
Set Minute Sub-Interval
NO
Has Minute AND Hour?
YES
Set Hour Sub-Interval
NO
Has Hour AND DayOfMonth?
YES
Set DayOfMonth Sub-Interval
NO
Has DayOfMonth AND Month?
YES
Set Month Sub-Interval
NO
Has PartOfDay AND DayOfMonth AND NO Hour?
YES
Set DayOfMonth Sub-Interval
NO
Has PartOfDay AND DayOfWeek AND NO Hour?
YES
Set DayOfWeek Sub-Interval
NO
Has Hour AND DayOfWeek?
YES
Set DayOfWeek Sub-Interval
NO
Has Month AND Year?
YES
Set Year Subinterval
Has TimeZone AND Hour?
YES
Set Hour TimeZone
NO
Has TimeZone AND NO Hour?
YES
Delete TimeZone Annotation
NO
Has Nth AND Period?
YES
Set Nth Period
NO
Has Nth AND Calendar-Interval?
YES
Set Nth Interval
Temporal Phrase w/ Sub-Intervals

Fig. 12. Chrono's sub-interval linking algorithm, including Next/Last parsing.

3.2 Evaluation

Evaluation of Chrono's performance on the Training Corpora utilized python scripts provided by AnaforaTools ‡ that compare Anafora XML [127] annotation files. All metrics reported exclude the "Event" entity because event identification is currently not implemented by Chrono, and was not included in the SemEval Task. Chrono's annotation of the Evaluation corpora was uploaded to the Post-Evaluation submission system for SemEval 2018 Task 6, and overall Precision, Recall, and F1 measures are reported in Tables 6 and 8.

‡ https://github.com/bethard/anaforatools

3.3 Newswire Results

The AQUAINT corpus of Newswire texts [128] consisted of 81 documents, provided by the task organizers. Average Precision, Recall, and F1-measure of 5-fold cross validation for Track 1 (parsing) are reported in Table 5 (annotations for "Event" and "Modifier" are ignored). Scores for "100% Correct Entity" consider the entity location and all properties (like sub-intervals), and scores for "Correct Span" only consider the entity location.

On average, all ML algorithms perform similarly for the "100% Correct Entity". All versions also obtain a higher F1 score when only considering correct spans versus getting all entity properties correct. This indicates that Chrono correctly identifies the majority of temporal entities, but has trouble parsing some of the properties.

ChronoNN processed the final evaluation data set, which consisted of 20 previously unseen Newswire articles, and received a F1 of .44. The evaluation data set contained five articles from BBC that were not represented in the training data set. Chrono's low performance indicates that it may be over-fit to the the training data set. This is one downfall of rule-based systems, where new rules need to be developed for each new type of temporal representation. Upon further review we found the submitted version of Chrono had three minor parsing flaws that resulted in unintentional false positives.

1) Formatted dates such as "2013-02-22" were being parsed twice. The first parse specifically looked for a 4-digit year and identified all correct entities, then the second parse looked for a formatted date with a 2-digit year, but didn't check to see if a year had already been found, so returned a 2-digit year with the value "22". This was easily fixed by having the 2-digit year parser check for a 4-digit year flag before proceeding (month and day flags were already implemented).

2) 24-hour time priority was incorrectly placed above 4-digit year. This resulted in any isolated 4-digit year being parsed as a 24-hour time expression rather than a year as originally intended. A simple flip of parsing order resolved this issue.

3) Numeric temporal expressions, such as an isolated 4-digit year, were being parsed as a whole phrase rather than breaking out each token within the phrase. For example, the year in the phrase "Last 1953" was not being identified because it was not in a phrase all by itself. To fix this the parsing function was edited to loop through each token in a phrase (a method that was already implemented in most other parsers and was just overlooked here).

ChronoNN received a Post-Evaluation F1 of .55 for Track 1 after implementing these fixes, which sets ChronoNN as the top performing system for SemEval 2018 Task 6, Track 1.

3.4 Conclusions and Contributions

In conclusion, there are many TERN tools that normalize temporal expressions into the popular ISO-TimeML standard, but this annotation scheme has some limitations in the types of expressions it can represent. The SCATE scheme was developed to represent a wider variety of temporal expressions, allow for events to act as anchors, and use mathematical operations over a timeline to define the semantics of each annotation; however, no tools existed that could normalize temporal expressions into its extremely fine-grained structure.

This chapter described the first hybrid framework for normalizing fine-grained temporal information into the SCATE scheme, which is implemented in the tool Chrono and available on GitHub [§] A version of this chapter was published as a full

[§] https://github.com/AmyOlex/Chrono.

100% Correct Entity			
	P	**R**	**F1**
Chrono NB	.686	.630	.657
Chrono NN	.684	.629	.656
Chrono DT	.687	.632	.658
Chrono SVM	.689	.630	.660
Correct Span			
Chrono NB	.823	.752	.786
Chrono NN	.820	.749	.783
Chrono DT	.822	.751	.785
Chrono SVM	.827	.755	.789
Evaluation Results			
Chrono NN	.46	.42	.44
Post-Evaluation Results			
Chrono NN	.61	.50	.55

Table 5. Chrono results on Newswire corpus for Track 1. All standard errors are <= 0.03, and no method performed statistically significantly better than another.

paper in the Proceedings of The 12th International Workshop on Semantic Evaluation [42].

CHAPTER 4

CONVERSION OF CHRONO TO THE CLINICAL DOMAIN

Chrono was originally built for parsing temporal expressions in the general domain due to a lag in getting access to the clinical corpus. After participation in the TempEval challenge, access to the clinical THYME corpus was granted. The following subsections describe the SCATE annotated portion of the THYME corpus made available to TempEval challenge participants, Chrono's out-of-the-box performance in the clinical domain, modifications made to Chrono after a detailed error analysis, and the improved results.

4.1 THYME Corpus with SCATE Annotations

The THYME corpus consists of de-identified clinical notes and pathology reports for colon and brain cancer patients [129]. For this work, we utilized the subset of the THYME colon cancer documents that have associated SCATE annotations in the Anafora XML format from SemEval 2018 Task 6 [130]. The Training Corpus includes 22 clinical notes and 13 pathology reports along with their gold standard Anafora XML annotations. The Evaluation Corpus includes 92 clinical notes and 49 pathology reports with the annotations withheld.

4.2 Out-of-the-Box Performance

Chrono's performance decreased significantly on the THYME Evaluation Corpus out-of-the-box with an F1 of 0.35, Precision of 0.49, and Recall of 0.27 (Table 6). This is due to Chrono having only been trained on Newswire text, thus, it saw a limited

data set	System	Precision	Recall	F1
THYME Eval	Chrono	0.49	0.27	0.35
THYME Eval	Laparra et al.	0.52	0.63	0.57
Newswire Eval	Chrono	0.61	0.50	0.55
Newswire Eval	Laparra et al.	0.58	0.46	0.51
THYME Train	Chrono 100%	0.439	0.244	0.314
THYME Train	Chrono Span Only	0.696	0.352	0.468

Table 6. Baseline performance, excluding "Event", on THYME Training and Evaluation corpora using SVM.

number of temporal expression examples.

Chrono's performance on the THYME Training Corpus resulted in an F1 of 0.314 when considering all entity properties (100% Correct Entity), and an F1 of 0.468 when only considering correct token span (Span Only). The higher Span Only result indicates that Chrono is identifying more correct entities than the 100% Correct Entity score indicates, but it is not assigning all the properties correctly. With the AnaforaTools evaluation script we are able to look at the performance on each SCATE entity individually to identify specific entities that significantly impact performance.

4.3 System Modifications

Addressing cross-domain parsing issues felt synonymous to playing the arcade game of Whack-A-Mole, where as one issue was fixed another popped up. Several code improvements resulted in a cascading series of other code bugs and/or logical issues that needed resolution prior to realizing a performance improvement. This section describes these adventures in code improvement, which identify six primary challenges encountered in cross-domain application of temporal expression extraction. The following examples relay how complex and interconnected temporal expression extraction can be, and demonstrate the need to go beyond basic pattern identification and dictionary look-up strategies to including contextual and semantic information in order to capture all types of temporal expressions.

4.3.1 Lexical Diversity

Different domains are expected to differ in their lexicon. For example, the clinical domain contains many specialized medical terms and clinical jargon that is not encountered in general domain texts [131]. This is also true for a temporal lexicon. Originally trained on the Newswire corpus, Chrono's lexicon was limited to examples found in this domain; however, by expanding Chrono's temporal lexicon the performance on several SCATE entities increased.

Performance on the SCATE entity "Modifier" improved after refining the lexicon to include missed terms such as "nearly", "almost", "mid", "over", "early", and "beginning", and removing terms that should be annotated with other entities such as "this", "next", and "last". These descriptive temporal tokens are commonly used in clinical texts to describe various events in the patient narrative such as when symptoms occur or patient histories. The PartOfDay entity was also augmented with the terms "bedtime", "eve", and "midnight" as these, and similar terms, are frequently utilized in clinical notes for medication instructions, such as "take one pill at bedtime". Significant improvement in performance was observed after these additions, with an F1 increase of 0.117 for PartOfDay, and an F1 increase of 0.241 for Modifier.

Patient records revolve around temporal information, such as conveying medication instructions, describing symptom time lines, and outlining patients' histories. We found that temporal phrases associated with these events, like "at that time", "take one-time daily", "in four weeks time", "since that time", etc., were ubiquitous. All of these expressions include the token “time”, which is annotated as a Period entity in the SCATE Schema. This token, along with others found frequently in clinical text such as “/min” and “/week” that are most commonly used as short-hand for conveying medication frequency, were not included in Chrono’s temporal lexicon. This resulted in poor performance for the Calendar-Interval and Period SCATE entities. The addition of 15 terms that were not present in the Newswire corpus significantly improved performance for these phrases. This result indicates that commonly used tokens have domain-specific frequencies. For example, the token “time” was used on average 0.32 times per document in the Newswire corpus and just over 4 times per document in the THYME corpus (Table 7).

4.3.2 Frequent Frequency

The frequency for some lexical terms, like “time”, in clinical texts is understandable as certain concepts that convey a patient’s narrative may be utilized over and over again. However, it is interesting that this observation also applies at the temporal entity level. For example, the initial build of Chrono excluded the SCATE entity Frequency because it is highly complex to parse and did not appear regularly in the Newswire corpus (0.12 times per document on average, Table 7). However, in the THYME corpus, the Frequency entity appeared on average 8.9 times per document–a 72-fold increase–which had a major impact on Chrono’s performance. In clinical texts, phrases specifying frequency such as “2 time per day” or “once a day” are abundant as they are routinely used for specifying medication or symptom frequency. This increase in clinical usage extends to all but two temporal entities, with Frequency having the second highest fold change next to Event (Table 7).

	Chrono	Newswire	Clinical
Entity	Implements	Avg Freq	Avg Freq
AMPM-Of-Day	Y	0.06	1.26
After	Y	0.25	2.29
Before	Y	0.44	0.91
Between	N	0.28	1.11
Calendar-Interval	Y	1.83	6.80
Day-Of-Month	Y	2.84	8.66
Day-Of-Week	Y	1.33	1.29
Event	N	0.91	151.97
Every-Nth	N	0	0.09
Frequency	N	0.12	8.91
Hour-Of-Day	Y	1.15	1.46
Intersection	Y	0.11	1.60
Last	Y	2.80	3.86
Minute-Of-Hour	Y	1.12	1.31
Modifier	Y	0.42	1.31
Month-Of-Year	Y	3.31	9.77
Next	Y	0.72	0.80
NotNormalizable	N	0.06	0.06
NthFromStart	Y	0.30	0
Number	Y	1.17	13.66
Part-Of-Day	Y	0.19	0.91
Part-Of-Week	Y	0.04	0
Period	Y	1.64	4.97
Season-Of-Year	Y	0.07	0.03
Second-Of-Minute	Y	0.67	0.17
Sum	N	0.01	0.03
This	Y	1.43	2.60
Time-Zone	Y	0.44	0
Two-Digit-Year	Y	0.98	0.23
Union	N	0.02	0.03
Year	Y	1.67	9.91

Table 7. The average frequency per document of each SCATE Entity for the Newswire (81 documents) and THYME (35 documents) training corpora. The “Chrono Implements” column indicates whether or not Chrono identifies a given entity (Y=yes, N=no).

4.3.3 Disambiguating Dosage

Clinical text commonly contains non-temporal numerical information representing lab test results or medication dosage along with their frequency. The majority of these instances in the THYME corpus were not identified as temporal because their values and formats were distinct. However, Chrono confused a few occurrences of medication dosage with a 24-hour time instance. For example, in the phrase "Vitamin D-3 1000 unit tablet" the "1000" was incorrectly assigned the 24-hour time value of 10am. In the current implementation of Chrono, if a 4-digit dose falls within the correct year range (1500 to 2050) or 24-hour time it will be annotated as such. A fix for this issue has yet to be implemented in Chrono, as it has a low rate of occurrence, but may include rules to identify dosage amounts such as "mg" and machine learning methods to disambiguate 4-digit numbers.

Another example of the need to disambiguate numerical values is found in the clinical phrase "Carotid pulses are 4/4". Without context, the "4/4" could be interpreted as the date "April 4th". This instance did not cause an issue with Chrono because a 2- or 4-digit year is required for a phrase to be identified as a formatted date. While this strategy worked for this example, it could become a problem when parsing files that contain year-less formatted dates. Thus, future improvements will include a numerical disambiguation module to aid in determining if a numerical phrase is temporal.

4.3.4 Cross-Domain Supervised Learning Training Data

Supervised machine learning (ML) methods require the use of annotated training data in order to generate a predictive model. Naturally, training data is chosen from the domain of the task as it is the most relevant. Chrono utilizes ML to disambiguate the SCATE entities Period and Calendar-Interval. First, rule-based logic identifies if an entity is a possible Period or Calendar-Interval, but it is hard to tell which one without considering context. Then the ML module decides which class the entity should be labeled. The training data for this task was initially from the Newswire corpus, but this performed poorly on clinical texts with an overall F1 of 0.544. To incorporate domain-specific contextual elements, Chrono was re-trained using just the THYME corpus, which improved performance to an F1 of 0.577. We then generated a model that utilized both the Newswire and THYME data, which performed slightly better, giving an F1 of 0.578. As temporal expressions can be domain-agnostic, it makes sense that training on cross-domain data would generate a more robust and generalizable model; therefore, we chose to use the cross-domain model.

4.3.5 Lexical Variation

An advantage of processing clinical texts is that you are introduced to a variety of writing styles and preferences from different departments and medical personnel, where each may represent the same temporal concept differently. This results in lexical variations of concepts, for example, the concept of "Monday" can be represented as "M", "Mon.", or, "monday", and a temporal reasoning system must be able to identify that these all refer to the same day. The following sub-sections discuss issues associated with variation in formatted dates, times, and long temporal phrases.

Variation in Formatted Dates/Times: There are a number of standard formats to convey dates and times, of which only a few were identified in the Newswire corpus and implemented in Chrono. Clinical texts introduced additional variability in date and time formats that Chrono was unable to handle correctly. For example, the date format "21-SEP-2009" contains a mixture of letters and numbers needing to be interpreted. Chrono uses regular expressions to identify formatted dates and times; however, the expression restricted all components to be digits, so dates with alphanumeric characters were not captured. Editing the regular expression to allow for alphanumeric characters fixed the capturing issue, but resulted in an error downstream where other methods expected a numeric month to be returned. Ultimately, a custom function was written to convert months represented as text to integers as existing conversion packages were not versatile enough to accommodate all lexical variations of these entities.

Similarly, hour and minute formats such as “5:45 PM” were not being recognized correctly because Chrono’s regular expression looked specifically for the format found in the Newswire corpus that contained seconds (hh:mm:ss). Debugging formatted time expressions proved to be a challenge because Chrono utilizes three different modules to parse out this data. First, a module to identify the hours, minutes, and seconds, followed by a module to identify AMPM entities, and finally, a module to link sub-intervals where both MinuteOfHour and AMPM entities are sub-intervals of HourOfDay. Interestingly, the performance of HourOfDay for the Span Only evaluation had an F1 score of 0.941 both before and after improvements, indicating that Chrono was actually identifying most of the hours correctly, but was missing specific SCATE properties.

Punctuation - To Include or Not to Include? Part of the HourOfDay parsing issue stemmed from temporal phrases at the end of a sentence, such as “2:04 AM.”, where the period ended up being part of the “AM” string. Initially, Chrono looked for AMPM entities without considering punctuation unlike the MonthOfYear parsing, which specifically accounts for punctuation such as “Dec.”. Thus, the “AM.” in the example was never identified, so the HourOfDay entity “2” would be lacking the subinterval link to the AMPM entity. To resolve this, Chrono was modified to utilize regular expressions in parsing out AMPM entities with and without surrounding punctuation.

One dilemma arose when considering the variants of an AMPM entity. For example, valid AMPM entity strings include “AM”, “am”, “A.M.”, and “a.m.”; however, “AM.” may not be considered a valid representation of an AMPM entity. Thus, Chrono specifically includes the period in the span only if there is a period after each letter in strings (e.g. “A.M.”), otherwise, the period is not included in the span. Implementing this fix resulted in a significant performance improvement for the AMPM entity and, oddly, a decrease in HourOfDay performance.

Where have the Minutes Gone? While the HourOfDay entity was performing well in the Span Only evaluation, the MinuteOfHour entity performed poorly in both Span Only and 100% Correct Entity evaluations. This was a result of Chrono looking for an HourOfDay in two different methods–one that identified formatted times and another that first looked for an AMPM entity and, if found, searched for an upstream HourOfDay. The majority of time expressions in THYME were formatted as “hh:mm” followed by an “AM” or “PM” which resulted in HourOfDay being identified by AMPM parsing and not the formatted time method. The AMPM method was designed to identify the pattern found frequently in Newswire texts (e.g. “5 PM”), which doesn’t include second or minute parsing. To fix this issue the formatted time method was adjusted to allow for the “hh:mm” format, so now the HourOfDay and MinuteOfHour entities are being identified and appropriate sub-intervals are annotated. However, this code improvement resulted in another decrease in performance of the HourOfDay entity.

Too Many Hours of the Day! The expected result of fixing the AMPM entity and formatted time parsing was increased performance on AMPM, MinuteOfHour, and HourOfDay entity parsing because the AMPM and MinuteOfHour sub-interval links were now identified correctly. However, HourOfDay performance actually became worse due to predicting too many HourOfDay entities. Further investigation revealed that every temporal phrase that included an AMPM entity had duplicate HourOfDay entities annotated (the same hour was annotated twice), one with the correct AMPM and MinuteOfHour sub-interval links and the other with no sub-interval links. This issue stemmed from a combination of the hierarchical parsing of formatted dates/times and inadvertently excluding a check to see if an HourOfDay entity already existed when parsing AMPM entities.

In Chrono, all temporal phrases are interrogated by all modules. To ensure only one entity of each type is identified in each temporal phrase Chrono implements a flag system. For example, in the phrase “Monday at 3:05 PM.” there is one DayOfWeek, one HourOfDay, one MinuteOfHour, and one AMPM entity. This phrase is first parsed by the formatted date/time module to identify the HourOfDay “3” and the MinuteOfHour “05” entity. Following is the identification of the “PM” AMPM entity; however, if this module finds an AMPM entity it then proceeds to look for an HourOfDay entity preceeding the AMPM substring. However, an HourOfDay had already been identified, and the AMPM module neglected checking this. Fixing this double parsing issue was straightforward as the AMPM module just needed to check if the HourOfDay flag had been set for the given temporal phrase. This error resulted in some initially puzzling results where the HourOfDay performance kept decreasing with every “improvement”, and ended up identifying twice as many HourOfDay entities as it should have. Different modules may be required for parsing different date/time formats, so it is important to ensure that all modules are consistently coded. It is also important to keep in mind that some formats are more frequent in one domain than another. This issue had not appeared when using the Newswire corpus because the majority of the AMPM entities were accompanied by the shorter format of “5 PM”, or contained the full “hh:mm:ss” format, whereas in the clinical domain the specification of hour and minutes, such as “3:05 PM”, was ubiquitous throughout the corpus.

Stop words splitting temporal phrases: Chrono was initially unable to handle stop words that connected temporal entities into a single phrase, which limited its performance on the THYME corpus due to the use of long temporal expressions in clinical texts. Chrono identified temporal phrases by looking for consecutive temporal and/or numeric tokens. If a stop word was identified (e.g. “is”, “of”, “at”, etc), the temporal phrase would be terminated–in some cases prematurely. For example, the phrase “beginning of this month on September 1” was originally separated into 3 temporal phrases: “beginning”, “this month”, and “September 1”. Other examples of temporal phrases that were incorrectly split include “2005 in April” and “October 14, 2010 at 02:07 PM”, which were both separated into two phrases. While individual temporal entities were identified correctly, the correct sub-intervals for each entity were unable to be assigned because Chrono only links sub-intervals within a single phrase. To fix this, code was added to tag “linking” words in the temporal phrase extraction module. Now, if a linking token is identified while constructing a temporal phrase it is ignored and the phrase is extended. This allows Chrono to correctly identify longer temporal phrases and results in correct assignment of sub-intervals, which brought the 100% Entity performance closer to Span Only.

Unexpected Effects of Longer Temporal Phrases: The inclusion of stop words in temporal phrases was a major upgrade to Chrono resulting in sub-intervals of longer phrases being correctly assigned. However, this had an unintended result that initially lowered the overall F1 scores for Calendar-Interval and Period entities. Investigating changes in performance revealed Calendar-Interval and Period entities that were correct were now incorrectly annotated with a link to a Number entity. This happened for phrases like “four times a day” or “one time a day”, which are highly frequent expressions in clinical notes as they are part of instructions for taking medications. This behavior resulted from Chrono’s parsing strategy for identifying associated numbers with SCATE entities where Chrono naively looked for a number token in the sub-string of characters preceding an annotated entity. This parsing strategy worked well for Newswire text as the majority of associated numbers appeared in formats similar to “2 weeks ago”, or “5 days”. Previously, Chrono assigned expressions like “four times a day” to two temporal phrases: “four times” and “day”. Thus, the Calendar-Interval “day” was correctly identified with no Number link. After including the stop words in the temporal phrases the first number in the phrase (e.g. “four”) was incorrectly associated with the Period or Calendar-Interval entity. Chrono’s number parsing strategy also became an issue with other frequent clinical phrases such as “one-time daily” where the number “one” was incorrectly associated with the Calendar-Interval “daily”. To fix this issue, Chrono’s definition of where a number had to be located in order to be linked to a SCATE entity was restricted to the immediately preceding token instead of the full preceding sub-string. This restriction works well for the THYME and Newswire corpora; however, may not work well with expressions such as “2 full weeks from now” where the Period “weeks” should be annotated with the Number “2”.

4.3.6 Document Design

Sentence Boundaries: An interesting temporal parsing issue appears in clinical texts regarding sentence tokenization due to item lists in the clinical record. Initially, Chrono did not tokenize on sentences as temporal phrases spanning sentence boundaries were not an issue in the Newswire corpus. However, clinical records in the THYME corpus contained entries like the following:

> “...my notes from December.
>
> 2. Ulcerative colitis...”

Where the top sentence ends with the temporal entity “December” followed by a numbered list item. Since Chrono did not consider sentence boundaries, this line break was removed in the preprocessing phase and the “2” that numbers the list item was parsed as a DayOfMonth associated with “December”. To resolve this issue, Chrono was updated to identify sentence boundaries. In Temporal Phrase Extraction, Chrono no longer allows a single temporal phrase to span sentence boundaries; however, the Temporal Disambiguation module still ignores these boundaries.

Metadata: Domain agnostic rules and procedures can be developed to identify many temporal expressions in written text, but metadata presents additional challenges in that it is inherently domain-specific, and can even be document type specific within the same domain. For example, pathology reports and clinical encounters with a physician can have their metadata formatted in different ways. In dealing with metadata the first question is if one wants to parse the metadata at all. A good reason to do so would be to gather contextual information that is not explicitly written in the text, like identifying the document creation date to disambiguate references to days of the week, etc. The gold standard SCATE annotations do contain dates from the metadata sections, so it is necessary for Chrono to identify these entities. Two issues arose when working on this problem: 1) How to identify a temporal token using whitespace tokenization when the metadata line contains little whitespace, and 2) whether or not to include the word “date” as a temporal token.

In the THYME corpus, metadata is formatted as:

[start_date=12/02/2010, rev_date=12/02/2010]

Using whitespace tokenization this line is split into two tokens–both marked as temporal as they contain formatted date strings. However, in the Temporal Phrase Extraction module this line is considered a single phrase because it is composed of two consecutive temporal tokens. This causes an issue as Chrono assumes there is only one of each SCATE entity type in a phrase; thus, initially Chrono only annotated one

data set	System	Precision	Recall	F1
THYME Eval	Chrono	0.76	0.51	0.61
THYME Eval	Laparra et al.	0.52	0.63	0.57
Newswire Eval	Chrono	0.57	0.54	0.55
Newswire Eval	Laparra et al.	0.58	0.46	0.51
THYME Train	Chrono 100%	0.729	0.478	0.578
THYME Train	Chrono Span Only	0.881	0.575	0.696

Table 8. Improved performance on THYME Corpora using SVM, excluding “Event”.

of the two dates in the metadata line. To resolve this, Chrono now converts all equal signs to spaces prior to whitespace tokenization, thereby separating the metadata text to four tokens. While this fix resolved the issue of parsing metadata dates, an equal sign could be useful information, so a more sophisticated approach will be required in the future.

The second issue with parsing metadata information arose when updating the lexicon of known temporal tokens. The word “date” is temporal, but had not been included in the initial lexicon of Chrono. Including “date” as a temporal token resulted in identifying the metadata line as a single temporal phrase again as it was now a consecutive sequence of four temporal tokens: “start_date”, “12/02/2010”, “rev_date”, and “12/02/2010”. As “start_date” and “rev_date” are just labels they should not be considered temporal entities. Some mentions of “date” were valid temporal expressions, but there were few of them. Thus, we decided to continue to exclude this token. To be applicable to different domains, more sophisticated methods to parse metadata will need to be implemented to resolve issues with temporal labels and other special characters seen in metadata text.

4.4 Improved Performance

Improvements made to Chrono using the THYME Training Corpus lead to a 0.27 and 0.24 increase in Precision and Recall, respectively, with a 0.26 increase in F1 measure for the Evaluation Corpus (Table 8). This resulted in Chrono being the top performing system for SCATE Normalization. Chrono's performance on the Training Corpus improved similarly with a Precision of 0.881 in the Span Only evaluation and 0.729 for the 100% Correct Entity. This indicates that Chrono is identifying the correct location of many entities, but it is having trouble setting all the properties correctly.

When designing a rule-base system it is possible to develop rules that overfit or are tailored to the training corpus (i.e. Newswire texts). Overfitting rules results in good performance on the training domain and poor performance on the testing domain, similar to Chrono's performance on the THYME corpus. However, when rules are adjusted to incorporate another domain it is expected that the performance in the training domain go down, indicating that it was overfitting the training domain. To see if this happened with Chrono, we re-evaluated our final model on the Newswire corpus. The results showed an insignificant 0.01 drop in F1 due to a 0.05 drop in Precision and a 0.04 increase in Recall, which indicates that Chrono is now more compatible with cross-domain application. Since we do not see a major drop in performance on the Newswire corpus we can conclude the original rules did not overfit the Newswire domain, but rather they were incomplete and required expansion to improve performance in the clinical domain.

4.5 Conclusions and Contributions

In conclusion, while the concept of time is the same regardless of the domain, its representation can vary. *This chapter demonstrated that clinical domain texts pose additional challenges to TERN systems, and identified 6 aspects of temporal parsing one should consider when migrating a system from the general to clinical domain.* These include:

1. Vocabulary differences (i.e. clinical terms and abbreviations).
2. The frequency of temporal entity usage (i.e. more mentions of frequency types).
3. Disambiguate numerical phrases as temporal or dosage/lab result.
4. Utilize appropriate machine learning data.
5. Lexical variation
6. Differences in document structure.

As Chrono was initially trained on Newswire texts, it's out-of-the-box performance on the THYME corpus was poor; however, through a detailed error analysis and algorithm improvements Chrono emerged as the top performing system for SCATE Normalization of clinical texts without compromising its ability to parse Newswire texts.

CHAPTER 5

THE I2B2 BENCHMARK

The 2012 i2b2 Temporal Challenge [132] provided the clinical NLP community with the first temporally annotated and de-identified clinical corpus for temporal reasoning. This corpus has become a benchmark in the field of clinical temporal reasoning for defining state-of-the-art performance for tasks such as TERN. The top systems participating in the 2012 i2b2 Temporal Challenge achieved span-based F-measure scores around 0.90, indicating good performance in identifying temporal expression spans, but saw reduced performance in normalizing the expressions to their correct temporal value.

There are only two known systems that parse temporal expressions into the SCATE schema, and they have access to a very limited set of gold standard data that is annotated with SCATE. Thus, it is not possible to assess Chrono's performance to the rest of the state-of-the-art TERN systems, which parse into the TimeML schema, because these annotation schema are not directly comparable. Additionally, through an investigation of the THYME Gold Standard annotations it was discovered that around 46% of errors were from incorrect Gold Standard annotations [67]. It is difficult to evaluate the performance of a method when close to half the errors are due to gold standard issues. Therefore, in order to compare the performance of Chrono to other state-of-the-art clinical temporal information extraction algorithms, it needed to also import and export annotations in the commonly used TimeML format. In addition to utilizing a common schema, it is important to evaluate Chrono on a gold standard with fewer errors. Thus, we upgraded Chrono to export in the TimeML format to get a more accurate performance evaluation using the 2012 i2b2 Temporal Challenge benchmark data set and to compare its performance to state-of-the-art methods from this challenge.

In the following sections the compatibility of the SCATE and TimeML annotations are discussed, modifications made to Chrono are described, results from running Chrono on the i2b2 data set are provided, and a detailed error analysis of Chrono and the top rule-base and hybrid systems from the 2012 i2b2 Temporal Challenge is provided.

5.1 Compatibility of SCATE and TimeML Annotation Schemes

To evaluate Chrono on the i2b2 data using the i2b2 scripts, the SCATE annotations needed to be converted to TimeML. While SCATE contains enough information to be converted to TimeML, the TimeML annotations do not contain enough information to be effectively converted to SCATE entities. This is due to the saved annotations in TimeML being normalized into the ISO standard. For example, the phrase “Thursday, June 3, 2000 at 12pm” would be saved in ISO as “06-03-2000T12:00:00”. This representation does not annotate the day-of-week mention “Thursday”, but SCATE does (even though it is redundant), and it is not clear from the ISO format if the text contains an AMPM entity, and second-of-minute, or a minute-of-hour entity, all of which must be annotated by SCATE if present to be counted as correct. Additionally, SCATE differentiates ”Periods” and ”Calendar Intervals” whereas TimeML treats them both as a DURATION or DATE, and it would not be straightforward to differentiate between them when converting to SCATE. For example, TimeML would annotate the token “week” as a DURATION in the following two phrases: “I have had pain for the past week.”, “I had pain all last week”. However, SCATE would annotated the first as a Period and the second as a Calendar

Interval. The main different between these scheme's is that SCATE is focused on the intervals of time while TimeML is focused on if the event associated with the interval happened continuously throughout said interval or only occurred at a specific point in time at the beginning or end of the interval mentioned in the text. This makes it difficult to convert based solely on either the TimeML or SCATE annotations, so additional measures need to be taken to include phrase context when converting these entities from SCATE to TimeML.

Converting SCATE to TimeML is possible as the SCATE data can be distilled down into the ISO format for DATE types and many DURATIONS as well. However, there are still challenges in retrieving a good conversion, and any conversion script would still need access to the full text document. Thus, it was decided to integrate the needed TimeML information into the existing SCATE objects within Chrono and provide an additional input/output mode for TimeML annotations instead of building a stand-alone conversion script.

5.2 System Modifications and Performance

Two phases of system modifications were implemented: 1) modifications to convert the existing SCATE annotations to TimeML, and 2) algorithm improvements to capture temporal elements not seen in the Newswire or THYME data sets.

5.2.1 Conversion Changes

ISO Formatting: The first change to Chrono was to convert explicit date/time strings to ISO format and store the normalized value for each temporal expression. This was done using an existing 3rd-party ISO conversion module in python named "dateutil". Initially, the raw temporal phrase identified by Chrono was input into this tool; however, some raw phrases were not able to be parsed by "dateutil", such as phrases that are part of the document metadata or header lines. Thus, for ISO conversion, a string was re-generated from the SCATE entities associated with a temporal phrase to be passed into the ISO conversion module. An example of a raw phrase that “dateutil” can not parse, but Chrono can is show in Quotation 4.

911203 Tuesday December 4A 1991 WEST (4)

Other phrases the “dateutil” method cannot handle are fuzzy and referential phrases such as “yesterday” or “3 days ago”. In addition, phrases such as “last Saturday at 3pm” were also parsed incorrectly as the “last” was ignored and the normalized ISO string would reference the next Saturday. For this later issue, proper setting of the reference time is required prior to conversion (see Future Work). For the former issue, a more complex solution is needed for proper normalization.

Finally, for proper ISO conversion of 2-digit years, such as ’97’, that are not part of date strings, a proper reference time had to be set in the “dateutil” method. This is simply set as the document creation time.

Periods and Intervals: Periods and Calendar Intervals are converted to DURATION ISO notation as this is the most frequent classification of these entities. This format must include the designation for a period (P), the number associated with the durations, and the units of the duration (e.g. D=days, M=months, Y=years, W=weeks). If the units are in seconds, minutes, or hours, the period designation must be accompanied by a “T”. For example, “3 months” would be coded as “P3M”, and the phrase “3 minutes” would be “PT3M”. Durations representing the same length of time, such as P1D and PT24H, are considered equivalent.

SCATE entities are clearly labeled as being a period or interval, which both are primarily coded as a DURATION in TimeML. If a number is associated with a SCATE entity, this is easily retrievable. Thus, implementing this conversion was straightforward for the majority of SCATE Period and Interval entities; however, some SCATE Periods and Calendar Intervals are actually annotated as a DATE in TimeML. Developing rules for this differentiation is difficult, so we decided to set all to DURATION at this point in time. The impact of this decision on performance is discussed in the following error analysis section.

Approximate Phrases: Another conversion from SCATE to TimeML was the conversion of entities with approximate modifiers. These entities required the TimeML APPROX attribute to be set along with a DURATION that had a number associated with it. While SCATE annotates these modifiers, it does not associate a number with them. Approximate phrases included "several days", "several minutes", "many days", etc. These have an annotation in ISO such as "P3D". However, choosing a number for the terms "several", "many", "few", etc. is challenging as the exact duration may be interpreted differently depending on the reader, context, and the magnitude of the units involved. For example, "a few minutes" could mean around 5 to 10 minutes, while "a few months" is more likely to be around 2 to 3. An analysis of these types of phrases was performed to determine what the consensus was in the i2b2 Gold Standard data set to inform the development of rules to convert these expressions from SCATE to TimeML.

An analysis of temporal expressions having the "APPROX" attribute set in the i2b2 Gold Standard revealed inconsistencies as to the exact numerical value with which these phrases were annotated. This was especially true for the modifier words "several" and "few". Numbers associated with "several" include 2, 3, 4, and 5. Numbers associated with "few" include 2, 3, and 4. Numbers associated with "many" include 5, 10 and 30. Even the same temporal expression was annotated with different values. For example, "several days" is coded as "P3D" in one document, and "P4D" in another. Similarly, “many days” is coded as P10D in one document, and P30D in another. This inconsistency makes it very difficult to correctly annotated these phrases and may impact performance.

In Chrono, these approximate modifiers were set to be a consistent value based on the average gold standard annotations and the descriptions for these terms on the LSAT exam*. In Chrono, the terms “few” and “several” are set to “3”, and “many” is set to “10”. Plural time expression, like “weeks”, without a number or approximate modifier are set to “2” with a modifiers of “NA”, and any singular period or interval is defaulted to a value of “1” with a modifier of “NA”.

5.2.2 Conversion Performance

After updating Chrono to output SCATE annotations into TimeML format we assessed it’s “out-of-the-box” performance. The i2b2 evaluation script was run to generate the aggregate performance of Chrono annotations for only TIMEXs and using overlapping span. Overlapping spans was chosen as Chrono spans are not directly coded to i2b2 preferences (such as including punctuation); thus, it is enough to know our spans overlap, which means we annotated approximately the correct phrase. Table 9 shows the overall Precision to be 0.56, Recall 0.81, and F1 0.66, which (except for Precision) are better than even the improved performance of Chrono on the THYME corpus.

Performing an error analysis of Chrono’s performance on the training data set revealed that the low Precision is due to Chrono annotating a lot of relative temporal and age-related expressions that i2b2 does not. For example, the term “recent” in the phrase “...go but not on home 02 with recent FEV1 27% of predicted value”,

* https://www.powerscore.com/lsat/help/lsat-quantity-terminology.cfm

and the term “now” in the phrase “...kidney transplantation and now has good graft function” provide ordering information in the clinical note for events, but are not annotated by i2b2 because they cannot be directly linked to a frequency, duration, date, or time. Age-related phrases include “28-year-old” and “72 years”, which are both specifically annotated in SCATE but not in i2b2. Additionally, Chrono was missing many temporal clinical abbreviations, such as “bid”, and was unable to parse 2-place dates formatted as “MM/YY” or “DD/MM”. Additionally, Chrono missed phrases like “postoperative day 2”. As Chrono is primarily rule-based, these stylistic writing differences between the THYME data set and the i2b2 clinical notes were not coded. Thus, further improvements to Chrono were made to account for these additional elements.

Additional sources of error include differentiating DURATION and DATE types. Chrono is coded to convert all SCATE Periods and Calendar Intervals to DURATION types in TimeML. However, some of these mentions are actually annotated as DATE types in the gold standard. For example, in the phrase “One week prior to presentation , he had chest pain..” the temporal expression “One week prior” is coded by Chrono as a DURATION, but in the gold annotations it is a DATE that is set to the day one week prior to the admission date. Similarly, phrases such as “On post-operative day 4” are annotated as a DURATION of 1 day by Chrono, but are given a specific date in the gold annotations. Both of these phrases require an anchor time and interval delta from the anchor time in order to calculate the date accurately. At the time, these errors were few compared to the lexical issues mentioned previously. Thus, these errors were not addressed in the next round of Chrono modifications, which was focused on improving Recall performance.

5.3 Improvements

The next round of improvements made to Chrono were focused on improving the Recall. The Precision numbers will naturally be lower because Chrono was built to identify a wider range of temporal expressions than what was annotated in i2b2. Thus, we focused on improving Recall first, followed by the Value Accuracy.

5.3.1 Clinical Abbreviations

The i2b2 data set contained a number of clinical abbreviations that actually represented a frequency. For example, "bid" represents "twice a day". To account for this, a new dictionary was created that contained a large list of temporal abbreviations used in clinical settings[†]. All of these abbreviations represented frequencies, thus, a new Frequency method was created to parse these phrases. Currently, only the abbreviations are parsed as frequencies, and none of the properties are being set. Future work will require setting the properties correctly and identifying frequencies that don't include abbreviations.

5.3.2 2-Place Dates

Two-place dates are tricky. They can either be of the format MM/YY, M/YY, MM/YYYY, MM/DD, or M/DD. If a 4-digit year is found, then it is unambiguous as to which place is a day, month, and year. However, if a 2-digit year is present, or the format is NN/NN or N/NN, it is unclear as to which place refers to day, month, and year.

Initially, Chrono was not recognizing 2-place dates at all as it looked for the standard 3-place format. Upon editing the code to identify 2-place dates as well,

[†] https://en.wikipedia.org/wiki/List_of_medical_abbreviations:_B

the issue became differentiating dates from test results and the formats MM/YY, M/YY, MM/DD, and M/DD. Chrono deals with this issue by constraining 2-place dates to have specific ranges of values. In a string with the format XX/NN or X/NN, the X or XX must be a numerical value between 01 and 12. If it is not the string is considered to not be a date. If the first place is determined to fit in the range for a month, the then second place must be between 1 and 31 to be classified as a day. If the second place is greater than 31 then it is classified as a 2-digit year. Now, this will of course run into situations where these rules will prohibit the date value from being interpreted correctly. For example, the string "01/10" could mean January 2010 or January 10th. Chrono would assign the later value to this string. While this may seem like a large issue, usually, when dates between 1/1/2000 and beyond are now relayed, the full 4-digit year is written for clarity, so we expect to not have too many issues with these rules. However, future work could include a machine learning algorithm to use the context of the passage to determine if the last 2 digits are representing a year or a day.

5.3.3 Improved Performance

Upon implementation of identifying clinical abbreviations and 2-place dates, Chrono's performance on the i2b2 training data set increased to a Precision of 0.64, Recall of 0.91, and F1 value of 0.75 9. Property attributes for Type, Value, and Modifier also increased, but are still below those of the state-of-the-art systems submitted to the i2b2 Temporal Challenge in 2012. Running Chrono for the first time on the unseen Evaluation data set from the i2b2 challenge resulted in similar performance to the improved training run with just a 0.01 drop in Precision, Recall, and F1.

Even in the Evaluation corpus we see that Precision is low and pull the F1 value down due to Chrono annotating additional types of temporal tokens not annotated

Run Type	Precision	Recall	F1	Type Accuracy	Value Accuracy	Modifier Accuracy
i2b2 Training						
Out-of-Box	0.56	0.81	0.66	0.49	0.45	0.66
Improved	0.66	0.92	0.77	0.60	0.54	0.79
Improved w/o relative	0.78	0.92	0.84	0.60	0.55	0.79
i2b2 Evaluation						
Improved	0.65	0.90	0.75	0.60	0.54	0.80
Improved w/o relative	0.78	0.90	0.84	0.60	0.54	0.80

Table 9. Chrono's performance on the i2b2 Training and Evaluation data sets.

by i2b2. In order to assess how much these tokens are affecting the Precision, we implemented a toggle in Chrono to turn off the annotation of relative and vague temporal tokens such as "briefly" and "recently". Table 9 shows the changed results with this toggle turned on. As can be seen, removing these relative terms increased the Precision and F1 measure without affecting the Recall or other properties, which affirms these extra terms were the issue.

5.4 Error Analysis on Evaluation Corpus and Comparison to Top Systems

An advantage of using the i2b2 data set is that they provide the output of the top 10 systems from the 2012 i2b2 Temporal Challenge, which can be analyzed and compared with new systems. Prior to performing an error analysis of Chrono's performance on the evaluation data set, a detailed error analysis of a few top performers from the i2b2 challenge was done to gain insight into the types of problems these systems had with this data and then compare that to the types of problems Chrono is having with this data.

5.4.1 Chosen Top i2b2 Systems

Using the performance results from the TIMEX section of Table 2 in Sun, et al. [21], we chose to analyze the TIMEX output of the following 3 systems:

System	System Type	Precision	Recall	F1	Type Accuracy	Value Accuracy	Modifier Accuracy
Mayo	Rule-based	0.88	0.92	0.90	0.86	0.73	0.86
MSRA	Hybrid	0.88	0.95	0.91	0.89	0.72	0.89
Vanderbilt	Rule-based	0.83	0.91	0.87	0.85	0.70	0.85
Chrono	Hybrid	0.78	0.90	0.84	0.60	0.54	0.80

Table 10. Performance of top systems from the 2012 i2b2 Temporal Challenge on the full evaluation data set along with Chrono's performance.

1. Mayo Clinic: The top performing rule-based system primarily using regular expressions.
2. Vanderbilt: A mid-range performing, rule-based system that was built on top of HeidleTime, a top performing general domain temporal tagger.
3. Microsoft Research Asia (MSRA): The top performing hybrid system utilizing rules, conditional random fields, and support vector machines.

Overall performance on the evaluation data set was re-calculated for these systems utilizing the provided system outputs and the evaluation scripts from the i2b2 data. These results are provided in Table 10 and match the results reported in Table 2 of Sun, et al. [21].

5.4.2 Error Analysis Strategy

Providing a detailed error analysis of all files in the evaluation data set would be time consuming, thus, a subset of files were chosen for analysis. To obtain the most informative files the i2b2 file-level evaluation results from the Mayo system were used to identify files with any one of Precision, Recall, or Value Accuracy that was close to or less than 0.75. These specific metrics were chosen as they are directly responsible for the ranking of systems in the i2b2 challenge, and they assess distinctly different aspects of each system's performance. The Mayo system what chosen to obtain these files initially because it was the top performing rule-based system and this

System	System Type	Precision	Recall	F1	Type Accuracy	Value Accuracy	Modifier Accuracy
Mayo	Rule-based	0.79	0.87	0.83	0.73	0.48	0.79
MSRA	Hybrid	0.81	0.95	0.87	0.88	0.59	0.86
Vanderbilt	Rule-based	0.74	0.89	0.81	0.80	0.52	0.82
Chrono	Hybrid	0.71	0.87	0.78	0.50	0.44	0.76

Table 11. Performance of top systems from the 2012 i2b2 Temporal Challenge and Chrono on the poor performing files from the evaluation data set.

analysis was meant to identify what types of phrases rule-based systems have trouble annotating and normalizing. Running this same analysis for the top hybrid system, MSRA, revealed no additional file that added to the list of difficult types of temporal phrases. Thus, the resulting list contained 18 files from the i2b2 evaluation data set that seem to be the most difficult files for rule-based and hybrid systems to parse. Table 11 shows the performance of the selected metrics of each top i2b2 system and Chrono for each of the 17 difficult files.

5.4.3 Top System Error Analysis Results

Analysis of the top i2b2 rule-based (Mayo and Vanderbilt) and hybrid (MSRA) results on the selected 18 low-performing files revealed several types of errors that each of the systems consistently made on the same types of temporal expressions:

- **Gold Standard:** Two of the poorest performing files were due to errors in the gold standard annotation.
- **Lexical:** Certain types of tokens were not recognized as temporal, or longer phrases were broken up so much the correct value could not be determined.
- **Frequency:** Some frequencies were either missed completely or phrases were incorrectly annotated as a frequency.
- **DURATION vs DATE:** Systems had a hard time determining if certain

vague or relative temporal phrases should be annotated as a DATE type or DURATION type.

- **Anchor Time:** Systems had trouble choosing the correct anchor time to calculate dates that were referred to by relative temporal expressions.
- **Delta Values:** Errors in identifying how much time to add or subtract from an anchor time to resolve a relative temporal expression.

In the following paragraphs, each of these error types is discussed with specific examples provided from each of the three top performing i2b2 systems. Following this detailed assessment of the top systems is a comparison to how Chrono performed on these same files.

Gold Standard errors include issues either with missing annotations in the gold standard file, or other problems related to the text or gold standard annotations that could be corrected to improve performance. Two gold standard annotation files were found to contain two different types of errors that lead to poor performance by all systems. One file had very poor Precision (around 0.3) with high Recall and value, which was an odd pattern compared to the rest of the chosen 17 files. This was cause by each of the systems annotating several 2-place dates that were not included in the gold standard. Looking into this it was found that these actually were dates and should have been annotated by the gold standard, however, it looked like the gold file was only half completed. In a second file, both Precision and Recall were high in all three systems, but the value accuracy was very poor across the board (around 0.25). Further investigation revealed the admission and discharge dates being from the year 2014, but in the actual text dates are given the year 2015. Additionally, many "POD#X" and "HD#X" phrases are included referring to postoperative days and hospital days. The annotators marked some of these as in the year 2014 and some in the year 2015. Thus “HD#3” was annotated to be 2/23/2014, but “HD#5” has the value 2/25/2015. Thus, the poor performance on these two files is due to incomplete or inconsistent annotations. If these issue had been corrected in the gold standard files, then the systems would have performed well on both.

Lexical errors include missing tokens that are annotated as temporal in the gold standard, annotating tokens as temporal that are not in the gold standard, or splitting up temporal phrases to a degree that causes incorrect value normalization. Overall, all 3 systems had few lexical errors in the chosen set of files; however, the ones they did have were usually consistent across the systems. The errors that did occur included all three systems missing the phrases “three cycles” and the token “one” in “one dose”, both of which were annotated as a FREQUENCY in gold. The two rule-base systems only annotated the “day” token in the phrase “day +11” where the hybrid system captured the full phrase. Similarly, the rule-based systems missed annotating the phrases “3 / week”, “14d”, and “2 wk”, but the hybrid MSRA system did capture all these phrase; however, it did not assign the correct type to any of them. With respect to breaking up phrases, all three systems broke up the phrase “daily for four days” into “daily” and “four days”. Mayo and Vanderbilt only annotated “weeks” in the phrase “one and a half weeks”, and MSRA missed annotating the token “later” in the phrase “A few days later” where the two rule-based systems captured the full phrase. Finally, the hybrid method from MRSA was the only one of the three to consistently annotate the token “sat” in phrases like “and o2 sat stable” as a DATE when it was actually referring to oxygen saturation.

Frequency errors included lexical issues where frequency phrases were not annotated at all, or where phrases were incorrectly flagged as a frequency. The rule-based systems from Mayo and Vanderbilt seemed to bear the brunt of these errors as their coded rules were unable to take context into account for phrases like “5 mg x 10 d” where it marked “x 10” as a FREQUENCY, however, the gold annotations marked “10 d” instead as a DURATION. The hybrid system from MSRA was able to annotate these correctly. Additionally, it seems the rule-based systems prioritize a FREQUENCY annotation over DURATION both in the example above and with the phrase “times one month”. Both Mayo and Vanderbilt only annotated the tokens “times one” and missed the “month” leading to this phrase being incorrectly annotated as a FREQUENCY when it should have been a DURATION. The MSRA system correctly captured the entire phrase “times one month” and gave it the correct DURATION temporal type. Finally, all three systems had trouble with the phrases “with a Vision stent , 3 x 18”, “negative CK X4”, and “negative troponin X4” that were all from the same file. All systems annotated the tokens “x 18” and “X4” as FREQUENCY types when they were not included in the gold standard as a temporal phrase.

DURATION vs DATE errors are those where a temporal phrase is marked as a DURATION type but should have been a DATE, or vice versa. Many DATE types are easy and straightforward to identify, such as the phrase “January 3, 2021”; however, temporal phrases that are referential or relative to an event or another time are more difficult. For example, in the phrase “a followup appointment is recommended in two weeks” the temporal phrase “two weeks” is referring to a specific date at which the next appointment should occur with the referential or anchor date being the time of the current visit (generally accepted as the document creation time unless otherwise stated in the text), so this would be annotated as a DATE type and given a specific date as the value in the TimeML schema. Table 12 lists the 17 phrases that tripped up at least one of the top systems with the correct type classification of DATE or DURATION. Mayo correctly classified only 3 while Vanderbilt did a little better to get 7 correct, and the hybrid system from MSRA performed the best with 9 out of

17 correct. In the following discussion of these errors we are only interested in the correct Temporal Type classification and not the actual value. Value accuracy will be discussed for many of these same phrases in the Anchor Time/Interval Delta section below.

Three phrases (1, 2, and 3 in Table 12) were incorrectly classified by all systems. In the SCATE schema, each of these three phrases would be listed as a Period type, however, in the TimeML schema two are DATE types (phrases 1 and 2) and one is DURATION (phrase 3). The key difference is that the two DATE type phrases are referring to a discrete event that will happen in one year (a CT scan) or one month (an ultrasound), whereas the DURATION phrase is referring to an event that has continuously happened over the course of three days (black stools). Phrase 3 is probably among the most difficult for any system to parse because it requires prior knowledge that the event of "black stools" is not discrete and can occur over multiple days.

Several phrases were consistently classified incorrectly by the Mayo and Vanderbilt systems. These include phrases 4, 5, and 6 in Table 12, which all reference dates in the past as indicated by the word "prior". Each of the rule-based systems seemed to miss this key word and assign these phrases to the DURATION type when they should have been a DATE. The hybrid MSRA system classified these instances correctly as a DATE; however, another instance of the word "prior" appears in phrase 7 and was classified correctly by Mayo as a DURATION, but incorrectly by Vanderbilt and MSRA as a DATE. Interestingly, phrase 8 also contains the token "prior" and is consistently classified incorrectly by Mayo as a DURATION, however, unlike the other phrases that include the word "prior", Vanderbilt identified this one correctly as a DATE along with MSRA. This indicates that each system may have a rule dictating priority over how these types of phrases are classified that potentially include key context words. Vanderbilt may have included the key word “until” in its rule-base, which may have led to the correct classification for this phrase.

The system from Mayo had particular trouble annotating the phrase “the day” that appeared twice in 2 files (phrases 9 and 10 were from one note and phrases 11 and 12 were from another). In each instance the phrase “the day” was annotated as a DATE by the gold standard, however, Mayo marked these as DURATION types while Vanderbilt and MSRA correctly classified them as DATE. Note the context for each instance of the same phrase “the day” is different for each occurrence. This actually leads to the values being different for each, however that is related to anchor time issues and will be discussed subsequently. A third file also contained the phrase “the day” (phrase 13), but gold annotated “the day PTA”, which means “the day prior to admission”. Again, Mayo defaulted to classifying this as a DURATION while Vanderbilt and MSRA correctly identified it as a DATE type.

Out of all 17 phrases, Mayo only got 3 correct (phrase 7, 14, and 15). As discussed above, Mayo most likely has a rule that classifies any phrase such as “two weeks” as a DURATION as it did this consistently regardless of the context. Interestingly, phrases 14 and 15 were both classified correctly by Mayo and incorrectly by MSRA. Both of these phrases include the key context word “later”, which was probably the signal word for a DATE classification in Mayo’s system and was not annotated as part of the phrase by MSRA. Vanderbilt also classified phrase 14 correctly and annotated the word “later” as part of the phrase, but got phrase 15 wrong as it missed annotating the key context word “later” indicating this was a DATE and not a DURATION. This may have been the result of Vanderbilt’s system using different rule sets to annotate these two phrases that handled the token “later” differently.

For the last 2 phrases listed in Table 12, phrases 16 and 17, neither Mayo nor Vanderbilt recognized these as temporal phrases. They were identified by MSRA as temporal, but the temporal type classification was wrong on both accounts. To normalize both of these phrases correctly, knowledge of clinical shorthand is required (e.g. "d/c" means discharge) and an understanding of the context and type of event (continuous or discrete) is needed. Even if MSRA used a machine learning module to classify temporal phrases as DATE or DURATION (note, it is unknown if they did), these two phrases would probably still present a challenge.

Finally, lets briefly revisit phrases 7 and 8, which are both from the same file, and both contain the same temporal phrase "two weeks prior", but one is annotated as a DATE and the other as a DURATION. All three systems were consistent in annotating these phrases and thus, each got one right and one wrong. This indicates static rules may have been implemented that do not take all the context into account in order to classify these phrases correctly. The complex and sometimes subtle contextual clues that humans can pick up on easily are clearly demonstrated throughout all of the examples in Table 12 where even the same temporal phrase can have a different meaning depending on the context (and as we will see in the next section can have different values as well). Thus, developing an exhaustive set of rules to identify any DURATION or DATE in any context is infeasible due to the variety of potential lexical and semantic forms; however, a machine learning model may be able to pull this off with the right features. While it is unknown if the MSRA system actually used a machine learning model for this task, it is clear that this system did perform better than either of the two rule-based systems on these difficult phrases.

Anchor Time and **Delta Value** errors go hand in hand, so will be discussed jointly. Anchor times are calendar dates used as the starting point for calculating the actual dates of a relative temporal phrases. For example, in the phrase "two weeks prior to admission" the anchor time would be the date of admission. To calculate the calendar date being referred to in this phrase you would also need to identify the

ID	Phrase	Gold	Mayo	Vanderbilt	MSRA
1	“...and a repeat CT scan in *1 year*.”	**DATE**	DUR	DUR	DUR
2	“A repeat head ultrasound is recommended in *one month*...”	**DATE**	FREQ	DUR	DUR
3	“*Three days ago* began to develop black stools...”	**DUR**	DATE	DATE	DATE
4	“...laprascopic cholecystectomy *7 weeks* prior to admission...”	**DATE**	DUR	DUR	**DATE**
5	“...during his most recent admission *1 year* prior .”	**DATE**	DUR	DUR	**DATE**
6	“...HSV outbreak occurred on 2017-09-13 approximately *one week* prior to delivery .”	**DATE**	DUR	DUR	**DATE**
7	“Over *the two weeks* prior to admission...”	**DUR**	**DUR**	DATE	DATE
8	“...chronic mild dyspnea on exertion until *two weeks prior* to admission .”	**DATE**	DUR	**DATE**	**DATE**
9	“...pain was intermittent through *the day*...”	**DATE**	DUR	**DATE**	**DATE**
10	“...it had essentially started earlier in *the day*...”	**DATE**	DUR	**DATE**	**DATE**
11	“...required a dilt gtt on *the day* prior to call-out...”	**DATE**	DUR	**DATE**	**DATE**
12	“...she was transitioned to PO diltiazem on *the day* of call-out .”	**DATE**	DUR	**DATE**	**DATE**
13	“...daughter says that on *the day PTA*...”	**DATE**	DUR	**DATE**	**DATE**
14	“*A few days later* she complained of dizziness .”	**DATE**	**DATE**	**DATE**	DUR
15	“...watched after his initial diagnosis , but *six months later* he developed...”	**DATE**	**DATE**	DUR	DUR
16	“...treated with levaquin on the floor and will complete a *14d* course at rehab .”	**DUR**	-	-	DATE
17	“...will see them again *2 wk* after d/c...”	**DATE**	-	-	DUR

Table 12. Temporal phrases that were hard to correctly classify as a DURATION or DATE temporal type. Gold standard temporal phrases are italicized and type classifications are in **bold green** if they match gold and colored red otherwise.

value (i.e. how many days) to add or subtract from the anchor time, which we refer to as the Delta Value. Anchor Times and Delta Values are only valid for relative temporal phrases classified as a DATE type, and these errors were the most pervasive throughout all poor performing files and across all systems making Anchor Time and Delta Value errors the top cause of poor performance. Table 13 lists several example

ID	Phrase	Gold	Mayo	Vanderbilt	MSRA
1	*Yesterday morning* , he developed...	5/12/2006	11/16/2006	5/13/2006	2003
2	On physical exam *today*...	5/16/2006	11/16/2006	5/13/2006	5/13/2006
3	Prior to discharge *today*...	5/16/2006	6/18/2006	6/18/2006	**5/16/2006**
4	Cholangiogram on *postoperative day number two* showed...	8/26/2009	8/19/2009	**8/26/2009**	8/24/2009
5	Cholangiogram on postoperative day number two...*At the time*...	8/26/2009	-	**8/26/2009**	8/24/2009
6	On *postoperative day number 17*...	9/10/2009	9/3/2009	**9/10/2009**	**9/10/2009**
7	On postoperative day number 17...*At the time*...	9/10/2009	-	**9/10/2009**	**9/10/2009**
8	Mother presented on *day* of delivery with preterm labor...	2016-05-05	-	-	**2016-05-05**
9	...*day of life two*...	2016-05-07	2016-05-06	2016-05-06	2016-05-05
10	...*day of life 18*...	2016-05-23	2016-05-22	2016-05-22	2016-05-05
11	Antibiotics were discontinued on *day of life three*...	9/24/2017	**9/24/2017**	**9/24/2017**	9/22/2017
12	...required a dilt gtt on *the day* prior to call-out...	2/17/2013	-	2/21/2013	2/18/2013
13	...transitioned to PO diltiazem on *the day* of call-out .	2/18/2013	-	2/21/2013	2/21/2013
14	...was followed by urology during her stay and will see them again 2 wk after d/c...*At this time* , urology will coordinate removal of...	3/13/2013	2/21/2013	2/21/2013	2/21/2013
15	...underwent cardiac catheterization *today*...	6/10/2015	5/4/2015	5/4/2015	9/2/2015
16	“...until one and a half weeks prior to admission ... was prescribed cortisone drops . *A few days later* she complained of dizziness .”	12/21/2009	1/3/2010	12/30/2009	-
17	...with chronic mild dyspnea on exertion until *two weeks prior* to admission .	4/6/2012	-	**4/6/2012**	4/19/2012

Table 13. Temporal phrases for which it was hard to correctly identify the Anchor Time and/or Delta Value. Gold standard temporal phrases are italicized and assigned values are in **bold green** if they match gold and colored red otherwise.

phrases that had an Anchor Time or Delta Value error by at least one system (the full table can be found in Supplementary Table S1. In total there were 50 phrases that the systems had trouble on from the poor performing files. Mayo got 2 dates correct, MSRA got 8 correct, and Vanderbilt performed the best by identifying 11 dates correctly, primarily due to a single file.

Through the error analysis it became clear that these systems attempted to employee complex logic to ascertain the anchor time for some relative phrases. Most of the time, phrases like "at this time" were annotated correctly using either the date of admission or discharge as the date these phrases references. However, other phrases were more difficult, and one even required a multi-step calculation based on context. Some of the difficult phrases caused errors from:

- Context switching with notes written on multiple days.
- Referencing multiple days of care as "postoperative day" or "day of life".
- Knowing when the admission or discharge date is the anchor time.
- Using the last annotated date as the anchor time.
- Upstream annotation errors leading to a cascade of downstream errors.

Deciphering when the context switches from being written upon admission to being written on discharge was difficult for all systems. Phrases 1, 2, and 3 in Table 13 relay some phrases that include the temporal words "yesterday" and "today" in the same file, however, one has the admission time as an anchor while to other refers to the date of discharge. According to the gold annotations, the phrase "Yesterday morning" was referring to the day prior to admission, which was 5/12/2006, and the phrase "today" referred to the day of discharge. The phrase "Yesterday morning" was included in the “HISTORY AND REASON FOR HOSPITALIZATION” section of the note, while the phrase “today” was in the “HOSPITAL COURSE” section. All three systems calculated a different, and incorrect, date for the “yesterday” phrase. Vanderbilt seems to assume it was the day of admission so was off by 1 day. Mayo assigned a date of 11/16/2006, which seems to have come from a DATE annotation in the previous sentence with the phrase “...history of CAD status post non ST elevation MI in 11/17 who presents with chest pain...”. Mayo annotated the token “11/17” as a date when it was not annotated by gold. From this file, and many others, it seems the Mayo system uses the most recently annotated DATE as the anchor for many of these relative phrases. Similarly, the MSRA system may have similar logic as it annotated the phrase “Yesterday morning” as the year “2003”. Looking at the context, it seems to have gotten this from the prior phrase “In 2003 , he had..”. While Mayo and Vanderbilt annotated “2003” as a year, they did not consider it as an anchor date. For the “today” term, there are two phrases in this file (phrases 2 and 3 in Table 13), and gold gives the same value (the discharge date) to both of them. For phrase 2, Mayo is still using the 11/16/2006 date from the previous section as the anchor time, while Vanderbilt and MSRA assume “today” is refering to the admission date. Interestingly, for the second “today” phrase the context points directly to the day of discharge and MSRA was able to get the correct date; however, both Mayo and Vanderbilt have the date 6/18/2006. This seemingly came from a new DATE having been annotated in the context prior to this last phrase, “June 18 , 2006 , at 8:30 p.m.”, showing that both Mayo and Vanderbilt have rules for anchor times that depend on the last annotated DATE regardless of the rest of the context.

Another interesting file included several “postoperative day number X” phrases followed by “at this time” phrases. For this file, it was important to keep track of the most recently annotated date as the narrative was describing the events after a surgery event. The systems seemed to be able to do this, however, choosing the correct anchor time was difficult for Mayo and identifying the delta value was a challenge for MSRA. The Vanderbilt system was able to calculate the correct dates for all instances in this particular file. Phrases 4 through 7 in Table 13 show a few example phrases from this file that will now be discussed. For this particular file, the admission date was 8/17/2009. One may assume that a surgery would have been performed on the day of admission in most cases, and this is exactly what the Mayo system does. Mayo was able to correctly identify the delta values to calculate the remaining "postoperative day" phrases; however, because this system assumed the anchor time was the day of admission the values were consistently off by a few days. In actuality, the key phrase "the patient was taken to the Operating Room on 2009-08-24" should have set the anchor time for all the postop phrases. Vanderbilt was able to identify this correctly, and thus obtained all correct dates that matched the gold annotations. MSRA was also able to ascertain this anchor date; however, this system was unable to process the delta value correctly when they were spelled out, which resulted in most of the postop phrases being set to the day of the surgery. This conclusion was reached because MSRA was able to obtain the correct calculated "postoperative day number 17" (phrase 6) when a number was used instead of a word. In addition, all systems were able to assign the "correct" values to the various "at this time" phrases, as these phrase values match the postop day date assigned in the previous sentence (phrases 5 and 7). The performance of each system on this file indicates the importance of not just assuming an operation or other medical event happened on the day of admission and instead looking for contextual clues as to what the anchor time should be for each phrase.

Similar to the "postoperative day number X" phrases in the previous paragraph, another file described the care of a newborn throughout the first month of its life referring to days of birth as “day of life X”. Similar to the issues MSRA had above with not recognizing any delta values that were spelled out, it assigned all values to be the day of admission. Both Mayo and Vanderbilt did perform calculations, however, they were off from the gold standard consistently by 1 day. Further investigation revealed that these two systems were using the day of admission as the first day of life; however, gold says the first day of life was the day after admission. This is a bit difficult for even a person to decipher because of phrase 8 “Mother presented on *day* of delivery with preterm labor...”, which would indicate that the first day of life may be the admission date. However, further reading reveals the context of phrase 10 includes a specified date: “...was discontinued on 05-23 (day of life 18).” Using this information to back-calculate when day of life 1 was we end up with the anchor date of 5/6/2016 instead of the admission date 5/5/2016. Notably, identifying this particular anchor date is a very complex task and requires high-level reasoning. Thus, identifying a single algorithm or machine learning model to calculate this will be challenging if possible at all. The Mayo and Vanderbilt systems were only a day off and had all the delta values correct, so this doesn’t seem too bad; however, in the previous file discussing postoperative event the operation event was more than a day away from the admission date, so it is not always good to assume the admission date is the anchor date. Assuming the day of delivery is the admission date probably does catch many of these types of files, for example, phrase 11 in Table 13 is from another file and references “day of life three”. Both Mayo and Vanderbilt get it correct by assume the admission date was the day of delivery, and MSRA has the now familiar problem of assigning this phrase the anchor time (admission date) because it can’t parse out “three” as a delta value.

A fourth file provides even more challenges for these systems in identifying anchor times. This file references the day of admission, the day prior to admission and a day

2 weeks after discharge (phrases 12 through 14). Mayo actually fails to annotated 2 of these 3 phrases as a DATE to start with and has them listed as DURATION types. For phrase 12, MSRA is the closest, but misses the key word “prior” and ends up assigning this phrase as the day of admission when it should have been the day prior to admission. Vanderbilt seems to be using the last annotated date from several sentences prior in the phrase “...was weaned off her pressors on 02-21...” as the anchor date as it doesn’t recognize the term “call-out” to indicate the day of admission. Both Vanderbilt and MSRA also use this same date for the next 2 phrases (13 and 14) “the day” and “at this time”, both of which require knowledge of the context to calculate correctly. Phrase 13 should be more straightforward with the immediate context. Instead of “the day prior to call-out” from phrase 12, we have a shift in context for phrase 13 with “the day of call-out”. Phrase 14 requires context from further away and over multiple sentences. The full phrase is “...was followed by urology during her stay and will see them again 2 wk after d/c...At this time , urology will coordinate removal of...”. Note the phrase “will see them again 2 wk after d/c” that refers to a date 2 weeks after discharge. This requires the parsing of the token “d/c”, which none of the systems seem able to do, and the knowledge of they will be seeing “urology” again and “urology” will be the one to coordinate a procedure, so this time it would be correct to use the previous date obtained from “2 wk after d/c” for the phrase “At this time”. Since none of the systems classified “2 wk” as a DATE, they didn’t have that information to go off of. If they did then the rules shown previously about using the last annotated date would probably have led to obtaining the correct anchor time in this instance; however, that doesn’t always work. For example, in phrase 15 “...underwent cardiac catheterization today...”, the term “today” was annotated by all 3 systems, but the date was calculated incorrectly, because all 3 systems used some other previously annotated date as the anchor instead of setting “today” as the

date of admission, which is what was provided by the gold standard. Thus, having a blanket rule to classify these referential dates as the last annotated will certainly catch some, but will not be very precise.

Finally, two additional phrases demonstrate the difficulty of figuring out whether to use the admission date, discharge date, or some other date from the context. Phrase 16 provides an example similar to phrase 14 where the anchor date is another relative phrase in the context prior. The full phrase plus context is "...until one and a half weeks prior to admission ... was prescribed cortisone drops . A few days later she complained of dizziness ." Before being able to identify the date for "A few days later" you first have to identify the date for the phrase "one and a half weeks prior to admission". In this instance, the "prior to admission" should be a straightforward clue as to what the anchor date is for this phrase, but one would need to be able to link it to the following phrase "A few days later". The MSRA system annotated "few days" as a DURATION, so did not provide a DATE, however, the Mayo and Vanderbilt systems did provide a date albeit the wrong one. The key was not being able to annotated the "one and a half weeks" phrase correctly, so Mayo chose to use the discharge date as the anchor, and Vanderbilt chose to the use admission date. Any phrase with the term "admission" or "discharge" seems like it would be simple to parse. Some systems seemed to utilize these keywords while others did not. For example, in phrase 17 "...with chronic mild dyspnea on exertion until two weeks prior to admission ." Mayo annotated "2 weeks" as a DURATION, Vanderbilt correctly chose the admission date as the anchor, but MSRA chose the discharge date as the anchor, so calculated the wrong date.

There are 5 main errors associated with Anchor Times and Delta Values. The most complex is determining temporal context switches to figure out what date or event a relative temporal phrase is referring to, especially in documents that were written over multiple days and don't necessarily specify which day each section was written. Another challenge is figuring out if relative phrases are actually referencing the admission or discharge date instead of a date in the written context. As in the examples above, sometimes these relative phrases are referencing the admission or discharge date, but sometimes they are referring to a previously mentioned date in the text (which itself might also be relative). For example, the phrase "at this time" could refer to the admission or discharge date, but it could also refer to an event that happened on a postoperative day or something that has not yet happened in the future (example from phrase 14 in Table 13). It was shown several times the assuming the last annotated date is the context of a current relative temporal phrase does not work, thus, being able to accurately identify temporal contexts is important for deciphering anchor times. Finally, phrases 14 and 16 show great examples of how prior incorrect annotations can have downstream effects on whether or not certain referential phrases can be calculated correctly. These phrases show a chain a referring temporal phrases, thus, if the first phrase in the chain is calculated incorrectly, or not annotated at all, that affects the downstream interpretations as well, leading to cascading errors.

In summary, Lexical issues contributed the least to the poor performance of the top i2b2 systems, while more complex errors involving the properties and normalized value of temporal expressions contributed the most. Lexical, as well as gold standard, errors are relatively straightforward to fix; however, DURATION vs DATE, Anchor Time, and Delta Value errors are more complex as they require context to understand and may not be able to be resolved with rules and regular expressions. The biggest problem all three systems had was resolving relative temporal expressions, such as "over the past two weeks", "two weeks prior" or "a few days later". Determining whether these are DURATION or DATE types is the first challenge, then once a

DATE type is assigned the system has to figure out the Anchor Time and Delta Value needed to calculate the correct date for a given relative temporal phrase. Both of these tasks can be complex as they both require knowledge of the context and some reasoning ability in order to correctly assign a date value. From the results of the error analysis on the top three systems, the most challenging aspect of identifying the correct Anchor Time is figuring out when the temporal context switches. If a system can be designed to correctly assign temporal contexts to phrases, sentences, or paragraphs, then many of the events related to relative or vague temporal phrases will be able to be more accurately placed on a timeline.

5.4.4 Chrono Error Analysis

Chrono was run on the same set of poor performance files from the analysis of the top i2b2 systems with a similar error analysis performed for comparison. The last row of Table 10 shows Chrono's performance on the full evaluation corpus as compared to the top i2b2 systems. Chrono performs on par with the top systems as far as Recall is concerned, however, Precision is about 0.1 lower than the others, due to Chrono identifying additional terms annotated by SCATE but not TimeML. Notably, the type and value accuracy is notably lower than the other systems. These scores indicate Chrono is identifying many of the same temporal phrases as the top i2b2 systems, but work still needs to be done when assigning properties and normalized values. the following discusses the error types as seen in the top i2b2 system analysis except for the gold standard errors as Chrono performed similarly to the top systems for these two files.

DURATION vs DATE errors were the second most problematic for Chrono and were frequently tied to the lexical parsing issues just discussed as well as some hard-coded rules. Prior to parsing temporal phrases into the TimeML schema, Chrono first parses text into the SCATE schema. In this schema phrases such as “day of life X” only have the “day” token parsed. So Chrono is recognizing this phrase (hence the high Recall as evaluation is set to count partially overlapping spans as correct), but it is setting it to a “Period” type. Currently, any Period or Calendar-Interval types in the SCATE schema are automatically set to a DURATION type in TimeML, thus, regardless of the context, this phrase will be set incorrectly to a DURATION instead of a DATE. As shown in the error analysis for the top performing i2b2 systems, developing a solid rule set to take all possible contexts into account to determine if a specific Period or Calendar-Interval from teh SCATE schema should be a DURATION or DATE in the TimeML schema would be challenging; thus, in the future Chrono will need to implement some type of machine learning approach to decipher these types of phrases.

Anchor Time and **Delta Value** errors on relative phrases are not that relevant to Chrono as it is not as mature as the top i2b2 systems in determining which phrases are DATE and which are DURATION types. This is mostly due to the static rule that sets all of these phrases to DURATION types. There are, however, a few instances where Chrono uses an anchor time to assign a DATE value, which are 2-place dates such as “9/02”, times without a date context such as “10:45 am”, and the phrase “Yesterday morning”. In Chrono, the anchor date is currently always set to the admission date. This is admittedly a naive way to set an anchor time as the other top systems clearly had logic in place to pick an anchor time as the last annotated, or closest, date in the text. This naive rule did work well in most all cases of 2-place date and time phrases analyzed; however, for these examples it did not and needs to be improved. The instance of “9/02” was annotated by all systems as “9/2/2015”, which is interpreting the format as “month/day”, however, the format was actually “month/year”, so should have been “9/2002”. Clues to this can be obtained from the context of the sentence and the note. First the sentence was talking in past tense about a procedure that happened in the past. Second the admission date was set to before Sept 2015, so it does not make sense to set a date for a procedure that happened in the past to a date in the future. Thus, setting some type of anchor date or time frame for the sentence may have helped interpret this correctly. For the "10:45 am" phrase, the full phrase with context is "On 03-02 at 10 am...for intubation at 10:45 am...". The context necessary for assigning a correct value is the phrase "On 03-02", which is located a few sentences prior to the time being annotated. Due to the more complex anchor time logic of the other 3 systems, Chrono was the only system that missed getting the value correct for this phrase. Finally, the phrase "Yesterday morning" was assigned the admission date "5/13/2006" by Chrono when it should have been the day prior. This issue is easily solved by adding in a rule to subtract a day from the anchor date when "yesterday" is present in the phrase.

In summary, while Chrono performs on par with the top i2b2 systems with respect to Recall, it is clear through the error analysis that it is not as mature. Chrono is unable to correctly parse out parameters and determine normalized values of complex and relative temporal phrases. This is primarily due to a few naive coded rules and a dictionary that is missing some key clinical temporal phrases and terms.

5.5 Conclusions and Contributions

At present, it is recognized that the utility of TERN tools that parse into the SCATE Scheme are limited as few downstream tools utilize these annotations, so it would be beneficial to have a system that can parse into both schemes. Additionally, it is standard to compare new TERN tools using benchmark data sets to the current state-of-the-art systems. The current benchmark data set in the clinical temporal reasoning field is the 2012 i2b2 Temporal Challenge; however, this data set uses the ISO-

TimeML annotation scheme, which is not directly comparable to SCATE. *To address these issues, this chapter discussed the differences between SCATE and ISO-TimeML and implemented 3 strategies to convert SCATE annotations to ISO-TimeML that include ISO formatting for explicit dates and times, conversion of Period/Calendar-Interval types to DATE/DURATION, and setting numerical values for approximate phrases.* After additional improvements to Chrono's algorithm, dictionary, and rule-base to parse new types of phrases not previously encountered using the development corpus, improved performance was achieved, specifically in Recall, which is now on par with the top performing systems from the i2b2 challenge. *Chrono is now the first system to parse temporal phrases into both the SCATE and ISO-TimeML schemes.*

In addition to providing the first dual-parsing TERN system, *this chapter also identified 6 types of errors state-of-the-art systems make when processing the 2012 i2b2 data set, which sets the stage for future work in this area.* Specifically, determining if a relative phrase should be a DURATION or DATE type, and identifying the Anchor Time and Delta Value for a relative phrase with a DATE type were the most frequent and complex errors experienced by the top i2b2 systems. The next chapter, dives into the details of the relative DURATION/DATE disambiguation task that both Chrono and state-of-the-art systems found challenging. As this is one of the most common errors in all systems, the remainder of this work focuses on developing a feature extraction methods using temporally fine-tuned BERT models to perform this temporal disambiguation task.

CHAPTER 6

TEMPORAL DISAMBIGUATION OF RELATIVE TEMPORAL EXPRESSIONS

There are two types of Temporal Disambiguation tasks that have been historically referenced in the literature: *Temporal Sense Disambiguation* (TSD) and *Temporal Type Disambiguation* (TTD). Both are similar to the classic task of Word Sense Disambiguation (WSD) [133, 134, 135, 136]. In language, the same lexical form of a word can have multiple meanings depending on the surrounding context. For example, the word "bat" could refer to a fuzzy animal with leathery wings, or a wooden stick used to hit a ball. The WSD task is to figure out what concept the word "bat" is referring to by utilizing context clues. Similarly, the TSD task it to identify if a word, such as "spring", is referring to the temporal sense of the Spring season, or a non-temporal sense of the word (e.g. an action or a physical spring) [137, 138]. On the other hand, the TTD task aims to identify the temporal type of a temporal expression so that it can be normalized correctly. An example is the expression "a week ago". In all instances, the word "week" refers to the concept of 7 days, so it has the same semantic meaning regardless of temporal type. However, TTD determines if the expression "a week ago" refer to a single point in time that an event occurred (a DATE type), or a span of time for which an event took place (a DURATION type). TTD is vital for RelIV-TIMEXs as they have to be assigned the correct type in order to be correctly normalized and positioned on a timeline.

Just as in WSD, utilizing the context around a temporal phrase can aid in TTD. For example, in the sentence "I crashed my car a week ago", it is clear the expression “a week ago” is referring to a single point in time that can be normalized to a specific date. On the other hand, in the sentence “My headaches started a week ago” the temporal expression is referring to a span of time, or duration, over which an event (headaches) continued to occur. In both cases the temporal expression is exactly the same, but the context surrounding them is different, including event type and key words like “started”.

An analysis of the context words surrounding relative temporal expressions in the i2b2 Gold Standard data set set shows that DURATION context tends to include words like “of”, “for”, and “over”, and DATE context is more likely to include the words “on”, “was”, and “at” (Figure 13 A and B). However, these terms are not exclusively used in the context of one or the other temporal type (Figure 13, C). Reduction to those terms exclusive to the context of DATE or DURATION (Figure 13 D and E) reveals that words like “feeds” and “started” are exclusive to DATE types, and “received”, “complete”, “past”, and “ago” are exclusive to DURATION in the i2b2 corpus. However, looking at the frequency of these terms it is clear they cannot be used for building universal rules as they only appear in a few expressions.

While there is no set of keywords that can always be used to disambiguate one type from the other, we could incorporate other features in a rule set to do this task. In addition to the contextual lexicon, additional features might include the type of entities/events nearby, punctuation, capitalization, verb tense, part of speech, and others. However, parsing this information from clinical records is a difficult task due to a lack of standardization, inconsistent punctuation use and capitalization, incomplete sentences, typographic errors, and other data quality issues [139, 140]. Additionally, some of these require additional parsing, such as performing named entity recognition to identify events, or concept extraction. To avoid building complex rule sets to accomplish the task of temporal disambiguation, this work sets out to embed

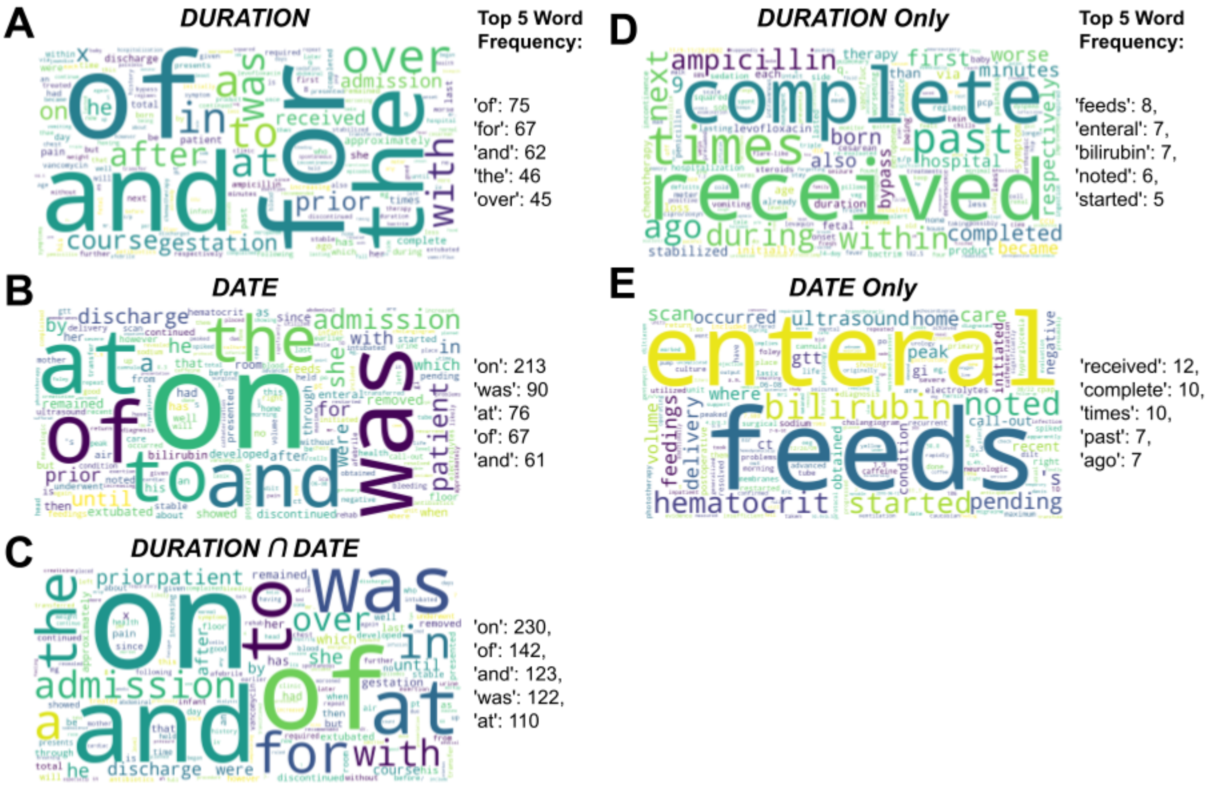


Fig. 13. Word clouds of context tokens surrounding relative DATE and DURATION temporal phrases.

temporal information into contextualized word embeddings and utilize those embeddings as the features in supervised learning models to disambiguate relative DATE and DURATION temporal phrases. Thus, the focus of this work is to improve the temporal type classification accuracy of Chrono for relative DATE and DURATION phrases.

This chapter is organized as follows: Section 6.1 provides details on how the 2012 i2b2 Gold Standard data set was filtered for RelIV-TIMEXs to create a RelIV-TIMEX Training and Evaluation data set. Next, two methods of infusing temporal information into BERT's contextualized embeddings are described in Section 6.2, including how the RelIV-TIMEX data set was reformatted for the sequence-to-sequence task, and the results of fine-tuning BERT to perform the temporal type multi-label classification task directly. Section 6.3 details the construction of features for two classic learning models, an SVM and CNN, using BERT's contextualized embeddings, and Section 6.4 reports on their architecture, training, and evaluation using the RelIV-TIMEX data set. Finally, Section 6.5 reports the performance of the new temporal disambiguation module and Chrono in 3 phases: 1) evaluating the temporal disambiguation modules on the RelIV-TIMEX data set using the gold standard phrases; 2) integration of the best modules into Chrono for evaluation on the RelIV-TIMEX data set using Chrono's phrase spans, in addition to comparison of 3 state-of-the-art systems from the 2012 i2b2 Challenge; and 3) results of Chrono when using a temporal disambiguation module for the full i2b2 data set using all temporal types in an end-to-end evaluation.

6.1 Creating the RelIV-TIMEX Gold Standard Data Set

For this work we utilize several variations of the 2012 i2b2 data sets, previously reviewed in Chapter 2, for training and evaluation. Briefly, the i2b2 data sets contain temporal phrases annotated and normalized to the ISO-TimeML standard. Temporal types include DATE, DURATION, TIME, and FREQUENCY. The training data set contains 190 documents with a total of 2,366 annotated temporal expressions, and the evaluation data set contains 120 documents with a total of 1,820 temporal expressions (Table 14). For End-2-End evaluations of Chrono and the other state-of-the-art systems, as well as the training of the multi-label classification sequence-to-sequence models in Section 6.2 below, the i2b2 data set is used as-is. When training the SVM and CNN classification models, the i2b2 data set is filtered to only DATE and DURATION types, which is referred to as the DD-TIMEX data set, and this is further filtered to only RelIV-TIMEXs for evaluation of the models. The DD-TIMEX

Temporal Type	i2b2 Train	i2b2 Evaluation	RelIV-TIMEX Evaluation
DATE	1641	1222	429
DURATION	407	341	307
TIME	69	60	-
FREQUENCY	249	197	-

Table 14. Number of annotated temporal expressions for the four temporal types in the full i2b2 data set and the filtered RelIV-TIMEX data set.

and RelIV-TIMEX data sets are described in more detail below.

6.1.1 Training Data Set: DATE/DURATION TIMEXs Only (DD-TIMEX)

As this work is focused on building a classifier for the DATE/DURATION TIMEX types, the i2b2 Training and Evaluation data sets were filtered to include temporal expressions that were annotated as a DATE or DURATION only (2,047 expressions, Table 14). All TIME and FREQUENCY annotated expressions were removed from the existing gold standards. These modified data sets are used in all model training, and are referred to as the "DD-TIMEX" Training and Evaluation Gold Standards. Note that these contain relative, incomplete, vague, and absolute/explicit temporal expressions.

6.1.2 Evaluation Data Set: RelIV-TIMEXs Only

To assess the performance of the temporal disambiguation module on the RelIV-TIMEXs, all absolute/explicit or incomplete temporal expressions were removed from the DD-TIMEX Evaluation data set. Any TIMEX meeting one of the following criteria was manually removed from the DD-TIMEX Evaluation data set:

- An explicit date or time, full or partial (e.g. 2/4/2013, 9am, 5/6, etc).
- A proper month or day of the week (e.g. February, Monday, etc).

- The name of a holiday (e.g. Halloween).

This primarily removed DATE types for a total of 429 RelIV DATEs and 307 RelIV DURATIONs (Table 14). This data set is referred to as the RelIV-TIMEX Evaluation data set and is only used for evaluation purposes.

Note that the DD-TIMEX data set is used to train all models described below, and the RelIV-TIMEX data set is used only for evaluation. This was done due to the limited number of relative examples so that the models would have more data to train from as the context surrounding explicit and incomplete temporal expressions can be similar to those of relative expressions. Additionally, the final model needs to be able to also classify some incomplete phrases for integration into the End-to-End pipeline.

6.2 Infusion of Temporal Information Into BERT Models Through Fine-Tuning

Recently, there has been an increase in attention to the infusion of temporal information into contextualized embeddings with the goal of improving prediction tasks. However, the focus has primarily been on temporal relation prediction ([141, 142]) with some recent work on temporal tagging in the general domain ([143]) and prediction of clinical outcomes ([144]). As of yet, there are no publications utilizing contextualized embeddings for the task of temporal disambiguation of RelIV-TIMEXs.

This work evaluates whether fine-tuning on simplistic and/or complex temporal classification tasks embeds temporal information into the extracted contextualized embeddings. Figure 14 summarizes the various combinations of fine-tuning, embedding extraction, and classification strategies explored in this dissertation. All strategies start with either the uncased BERT Base language model [56], referred to as "Bert-Base", or the clinical BioBert model fine-tuned on biomedical literature and clinical notes by Alsentzer et al. [72], referred to as “ClinBioBert”. The strategies using the unmodified BertBase and ClinBioBert contextualized embeddings are considered the baseline for this work (Figure 14A), and are referred to as the “baseline BERT models” when discussed together.

In the following sections, we first describe a high-level binary classification task used to fine-tune BertBase and ClinBioBert. This binary fine-tuned model is either used to obtain contextualized embeddings for input into classification models (Figure 14B), or as the initiating model for fine-tuning a sequence-to-sequence (Seq2Seq) classification model (Figure 14C and 14D). Next, two versions of a Seq2Seq fine-tuning method that utilizes the binary fine-tuned BERT models for initialization (Figure 14C and 14D) or the baseline BERT models (Figure 14E and 14F) are discussed, including an evaluation of their ability to directly classify temporal types (Figure 14D and 14F). Finally, the Seq2Seq fine-tuned BERT models are used to extract contextualized embeddings for down-stream Support Vector Machine (SVM) and Convolutional Neural Network (CNN) classifiers (Figure 14C and 14E).

Unless otherwise specified, all work, including fine-tuning BERT models and training of classifiers, was performed on the Compile.vcu.edu server with 128 AMD 32-Core Processors and an Nvidia Tesla T4 GPU using CUDA Version 11.6.

6.2.1 Binary Temporal Sentence Classification

Binary temporal fine-tuning is achieved by fine-tuning the existing BertBase and ClinBioBert models on a binary temporal task (Figure 14 B, C, and D). *For this work we chose to classify sentences as either containing or not containing temporal information.* The “BertForSequenceClassification” model from the HuggingFace Transformers Python library [145] was used, which is the default BertBase model configuration with a single linear layer added for classification. For the binary clas-

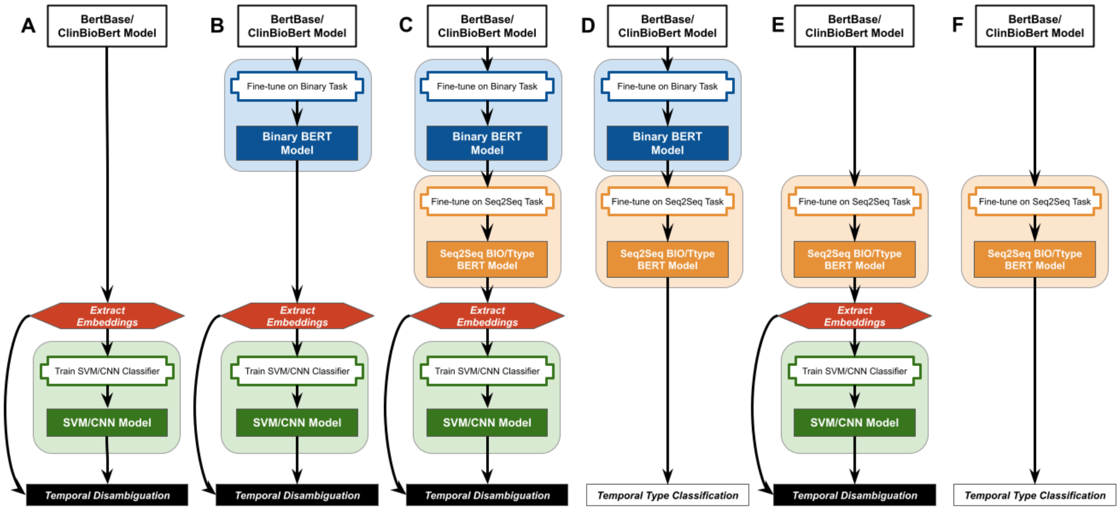


Fig. 14. Overview of the fine-tuning, embedding extraction, and classification strategies explored in this dissertation.

sification, a classification layer with 2 labels was specified for fine-tuning, and the embedding for the “[CLS]” token, which represents the full sentence, was used as the input (Figure 15).

6.2.1.1 Gold Standard Training and Evaluation Data Sets for the Binary Sentence Classification Task

The i2b2 annotated training data set was processed to mark all sentences across all documents as either containing a temporal annotation of any type or not. Specifically, the i2b2 annotated XML files were parsed with a modified python script obtained from Emily Alsentzer’s GitHub page*. This script was originally written to convert the i2b2 TimeML annotated XML files into a form that could be used for training a Seq2Seq classifier on the annotated EVENTs. The output was a text file with 2 columns: a word or token in the sentence and its associated beginning-inside-outside (BIO) label (e.g. B-event, I-event, O). This script was edited to extract the

*https://github.com/EmilyAlsentzer/clinicalBERT/tree/master/downstream_tasks/i2b2_preprocessing

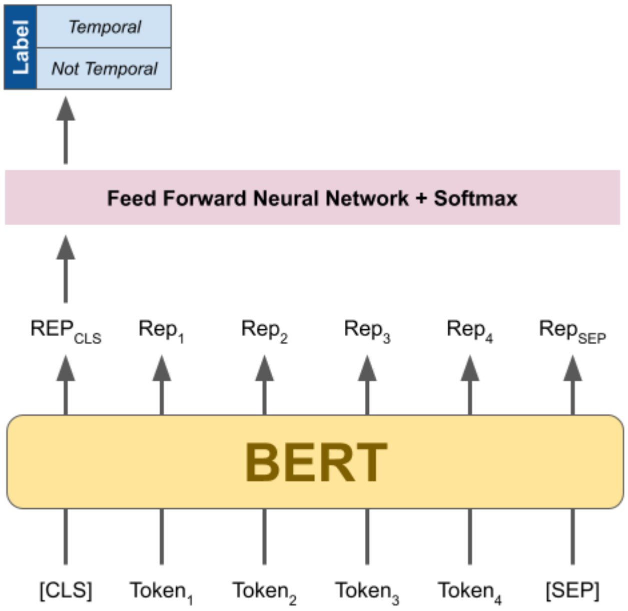


Fig. 15. Binary classification BERT model structure.

TIMEX tag data instead of the EVENT tag, and to output an additional file with binary labeling of sentences (i.e. 1 if there is at least one TIMEX tag associated with a sentence or 0 if not). This binary labeled file was used as input for the binary fine-tuning of BertBase and ClinBioBert models.

The i2b2 training data set contains a total of 7020 sentences with approximately 28% having at least one type of temporal annotation, and the evaluation data set contains a total of 5281 sentences with 27% having a temporal annotation (Table 15). While preserving the ratio of temporal to non-temporal sentences, the training data set was split into development (90%) and validation (10%) data sets to be used for identifying optimal hyper-parameters. The full training data set was then used to build the final model with the evaluation data set used for reporting performance.

Data Set	Temporal Sentence Count	Non-Temporal Sentence Count	Total
Training	1935	5085	7020
Evaluation	1432	3849	5281

Table 15. Summary of temporal and non-temporal sentences in the i2b2 Training and Evaluation data sets.

6.2.1.2 Binary Fine-Tuning

The BertBase and ClinBioBert models were put into fine-tuning mode and trained on the binary classification task of determining if a given sentence did or did not contain temporal information. Fine-tuning was done over 2, 4, 6, and 8 epochs, with learning rate = 2e-5 and epsilon = 1e-6. Hyper-parameters in BERT whose names include 'bias', 'gamma', or 'beta' have a weight decay rate of 0.0, and all others have a weight decay rate of 0.01[†]. A batch size of 16 and a max sentence length of 256 were utilized as additional GPU and BERT parameters, respectively. The binary fine-tuning was performed on the Pine.cs.vcu.edu server.

6.2.1.3 Binary Fine-Tuning Results

The BertBase and ClinBioBert binary classification models, henceforth referred to as Binary BertBase and Binary ClinBioBert, performed well after 4 epochs on the Training and Evaluation data sets (Table 16).

6.2.2 Fine-Tuning Sequence-to-Sequence BERT Models for Temporal Type Classification

While this work is focused on RelIV-TIMEX type classification, it would be beneficial to know if it is possible to achieve state-of-the-art results by directly fine-tuning a BERT model to identify all temporal types at the token level. This would

[†] https://mccormickml.com/2019/07/22/BERT-fine-tuning/

		Binary BertBase			**Binary ClinBioBert**		
	Measure	Temporal	Non-Temporal	Weighted Avg	Temporal	Non-Temporal	Weighted Avg
Training	P	0.98	1	0.99	0.99	1	0.99
	R	0.99	0.99	0.99	0.99	0.99	0.99
	F1	0.99	0.99	0.99	0.99	1	0.99
	Accuracy	-	-	0.99	-	-	0.99
Evaluation	P	0.93	0.98	0.96	0.93	0.97	0.96
	R	0.94	0.97	0.96	0.92	0.98	0.96
	F1	0.93	0.97	0.96	0.93	0.97	0.96
	Accuracy	-	-	0.96	-	-	0.96

Table 16. Results of fine-tuning BERT models on the binary temporal sentence classification task for 4 epochs.

result in limiting the amount of work needed to identify each temporal type using multiple learning models or rules. To achieve this, the baseline BERT models, as well as the binary fine-tuned versions, were used as the initial models for fine-tuning a sequence-to-sequence (Seq2Seq) classifier at the token level. Figure 16 visualizes this altered BERT architecture at a high level where the individual contextualized embeddings are passed to a dense linear layer for classification.

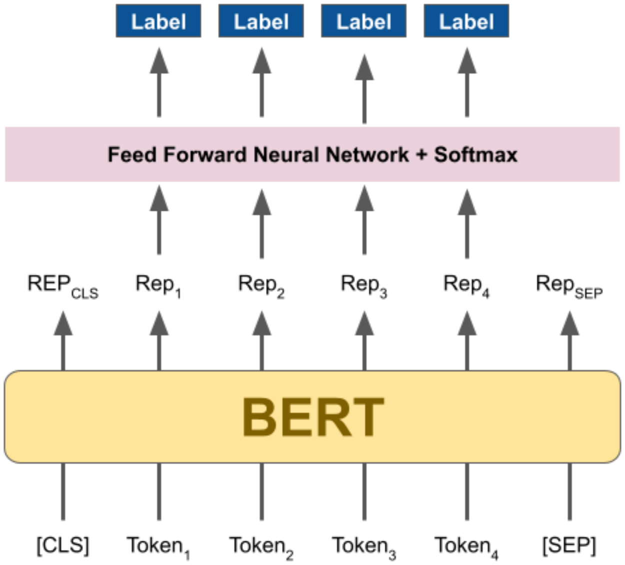


Fig. 16. Sequence-to-Sequence classification BERT model structure.

6.2.2.1 Gold Standard and Evaluation Data Sets

As described in Subsection 6.2.1.1, a python script obtained from Emily Alsentzer's GitHub page[‡] was modified to annotate tokens in the i2b2 Training and Evaluation data sets with TIMEX labels instead of EVENT labels. Labeling was done in two ways: 1) Tokens were labeled using the beginning-inside-outside model where the "beginning" is the first token of a TIMEX, the "inside" is all subsequent tokens in a TIMEX, and any token not part of a TIMEX is labeled as "outside". These models are referred to as Seq2Seq-BIO models. Thus, each of the four TimeML TIMEX types had two associated labels for a total of 9 labels (left side of Table 17). 2) Tokens were labeled with the temporal type only (Seq2Seq-Ttype models) without differentiating between the beginning and inside of a TIMEX (right side of Table 17) for a total of 5 labels. Due to the large number of "outside" label for both the BIO and Ttype labeling schemes, these values were excluded when calculating the evaluation metrics in order to focus on the temporal types specifically.

6.2.2.2 Sequence-to-Sequence Fine-Tuning

Seq2Seq temporal fine-tuning was achieved by fine-tuning the baseline BertBase and ClinBioBert models and the binary fine-tuned versions on the Seq2Seq multi-label classification task (Figure 14 D and F). The "BertForTokenClassification" model from the HuggingFace Transformers Python library [145] is used. This model adds a dense linear layer on top of the hidden-states output for individual token classification. When initializing from the binary classification pre-trained models, the linear classification layer with 2 labels is replaced by a multi-label classification layer using either 9 or 5 labels for the BIO or Ttype strategy, respectively.

[‡] https://github.com/EmilyAlsentzer/clinicalBERT/tree/master/downstream_tasks/i2b2_preprocessing

BIO Label	**BIO Count**	**Ttype Label**	**Ttype Count**
B-TIME	56	TIME	173
I-TIME	117		
B-DATE	1152	DATE	2247
I-DATE	1095		
B-DURATION	313	DURATION	739
I-DURATION	426		
B-FREQUENCY	185	FREQUENCY	339
I-FREQUENCY	154		
O (outside)	91682	O (outside)	91682
Total	**3498**	**Total**	**3498**

Table 17. Token counts for the BIO and Ttype multi-label Seq2Seq classification tasks.

The respective BERT models were put into fine-tuning mode and trained on the Seq2Seq-BIO or Seq2Seq-Ttype classification tasks. Fine-tuning was done over 2 and 4 epochs, with learning rate = 2e-5 and epsilon = 1e-6. Hyper-parameters in BERT whose names include ‘bias’, ‘gamma’, or ‘beta’ have a weight decay rate of 0.0, and all others have a weight decay rate of 0.01[§]. A batch size of 32 and a max sentence length of 256 were utilized as additional GPU and BERT parameters, respectively.

6.2.2.3 Seq2Seq Fine-Tuning Results

Seq2Seq fine-tuning was evaluated for both 2 and 4 epochs using the BIO and Ttype classification strategies. Models run for 4 epochs returned better results overall and were chosen to move forward in the pipeline. Table 18 displays the weighted average of the BIO and Ttype Seq2Seq models fine-tuned for 4 epochs. Full results, including confusion matrices can be found in the Appendix in Supplementary Tables S2-S3. The Seq2Seq-BIO BertBase fine-tuned model outperformed the Seq2Seq-BIO ClinBioBert models, and both binary adaptations. However, the best performing model is the Seq2Seq-Ttype ClinBioBert model. In all instances, first fine-tuning on the binary classification task resulted in poorer performance on the Seq2Seq classification, and none of the models outperformed the current state-of-the-art from the i2b2 tasks [10]. Thus, simply fine-tuning an “out-of-the-box” BERT model for temporal type classification is not a viable strategy. Recent work by Almasian et al. [143] also notes that Seq2Seq temporal type classification performance does not yet surpass rule-based approaches, and is working to develop a transformer architecture to improve performance. However, Xu et al. [146] found that contextualized character embeddings do improve performance of classifiers when normalizing TIMEXs from

[§] https://mccormickml.com/2019/07/22/BERT-fine-tuning/

general and clinical domain texts into the SCATE schema. While the neural architecture may not be optimal for direct classification of temporal types, the infusion of temporal information into the contextualized embeddings may improve performance of classical learning models for the temporal type disambiguation task, which is the topic of the next section.

		BertBase		ClinBioBert	
Labeling Strategy		Seq2Seq	Binary-to-Seq2Seq	Seq2Seq	Binary-to-Seq2Seq
BIO Model	P	**0.752**	0.561	0.746	0.529
	R	**0.79**	0.523	0.76	0.513
	F1	**0.769**	0.53	0.749	0.507
Ttype Model	P	0.805	0.632	**0.811**	0.637
	R	**0.845**	0.61	0.843	0.59
	F1	0.823	0.615	**0.824**	0.611

Table 18. Seq2Seq fine-tuning results using BIO and Ttype labeling strategies.

6.3 Feature Construction with Temporally-Infused Contextualized Embeddings

Fine-tuning BERT models on the temporal type classification task was not able to surpass state-of-the-art results; however, the modified contextualized embeddings may aid classical learning models. *This section aims to determine if these modified contextualized embeddings can be used as features to perform TTD for RelIV-TIMEXs using classic Support Vector Machine (SVM) and Convolutional Neural Network (CNN) models to obtain state-of-the-art or better results.*

Feature extraction aims to identify a single, or set of, contextualized embeddings from the BERT models to be used as input for learning models. This includes using contextualized embeddings for just the temporal phrase, as well as adding embeddings from the surrounding context, and embeddings from words to which the temporal phrase is paying the most attention. Figure 17 shows the 3 strategies to feature extraction from contextualized embeddings for the SVM and CNN architectures, which are explained in detail in the following sections. However, to obtain a single representative embedding for each token from a BERT model, summarization of the 12 contextualized embeddings returned by BERT must be performed first in order to use them as features for downstream classification. The following subsections first detail the methods used to summarize contextualized BERT embeddings at the token level, followed by the explanation of the algorithm developed to identify to which tokens a temporal phrase is paying the most attention by summarizing the attention weight matrices.

6.3.1 Contextualized Embedding Token-Level Summarization

6.3.1.1 Resolving Sub-word Embeddings for Out-of-Vocabulary Terms

Word-piece tokenization can result in a single token identified by whitespace tokenization having multiple tokens (termed subwords), each with its own embedding (Figure 5). In BERT models, subword tokens are identified with a prefix of two hashes "##". For this work the last subword embedding is chosen to represent the entire whitespace tokenized token[¶].

6.3.1.2 Summarizing Token Embeddings

BERT utilizes a multi-head self attention model with multiple hidden layers (see Section 2.7.4), which results in each token having multiple embeddings. Specifically, the BertBase and ClinBioBert models used in this work have 12 hidden layers each

[¶] https://mccormickml.com/2019/05/14/BERT-word-embeddings-tutorial/)

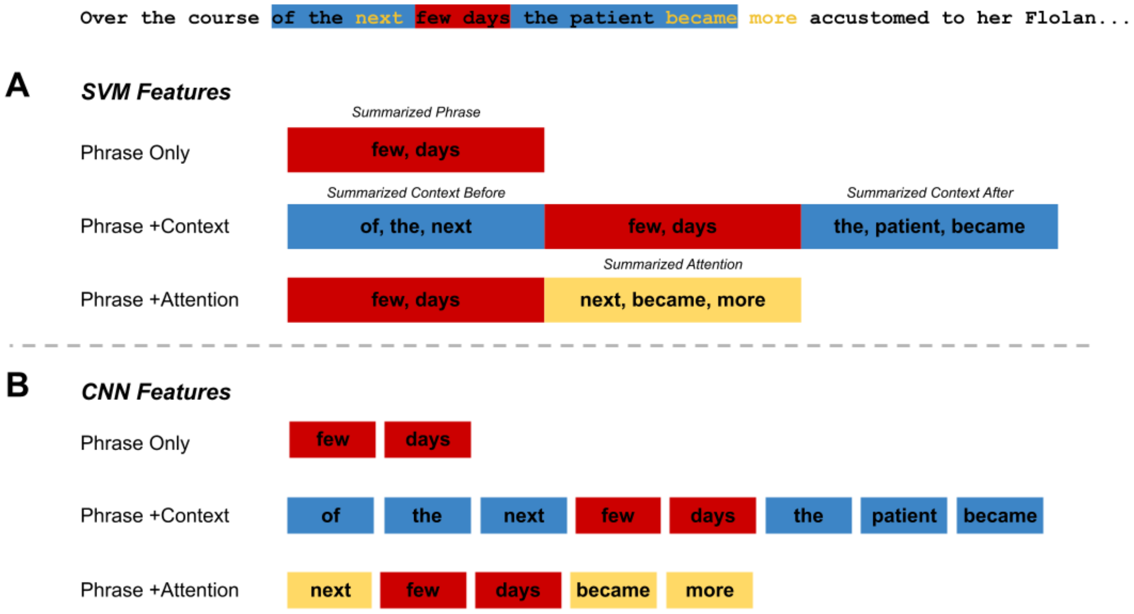


Fig. 17. Overview of contextualized embedding feature extraction strategies. The example sentence highlights the temporal phrase (red), context tokens with a window of 3 (blue), and the top 3 tokens most attended to by the temporal phrase (yellow). SVM feature components (phrase, context, or attention) are summarized into a single embedding then concatenated as shown. CNN features are sorted based on sentence order, then fed into the model for classification.

with 12 attention heads. As indicated earlier, the contextualized embeddings generated by the 12 attention heads are concatenated so that each layer outputs one embedding per token; however, the layers are not combined further. Thus, in these BERT models each token has 12 contextualized embeddings.

BERT hidden layers are updated in a sequential fashion where the output of one layer is the input to the next; thus, it is reasonable to assume that the last few layers will contain more contextual information about a single token than the first few layers. This is supported by the finding that concatenating the embeddings from the last 4 hidden layers to represent a token achieved the highest results out of other combinations for the CoNLL-2003 Named Entity Recognition task in the original BERT paper [56]. Thus, for this work, each BERT token is represented by a 3072 length vector from the concatenation of the last 4 hidden layers, each of length 768.

From this point forward, tokens are referred to as "summarized tokens" to indicate they have been preprocessed to resolve embeddings reported by multiple hidden layers and word-pieces. It is these summarized token embeddings that are fed into the feature extraction algorithm.

6.3.2 Identifying Temporal Phrase Attention

As discussed in Chapter 2, attention is the key to obtaining contextualized word embeddings because it dictates how much of every other token's embedding should be included in the current token's contextualized embedding. This work seeks to explicitly include the embeddings of the other tokens that are being paid the most attention to by a complete temporal phrase. *In this section the attention architecture is reviewed followed by an explanation of the algorithm that summarizes the attention matrices of all tokens in a temporal phrase to identify a set of tokens attended to by the temporal phrase.* To my knowledge, this is the first work to attempt the summarization of BERT attention matrices.

6.3.2.1 Review of Attention Architecture

At the core of the BERT attention structure returned by the model is an nxn matrix of attention weights, where n equals the length of the sentence (or padded sentence if using padding), including the [CLS] and [SEP] tokens (Figure 18A). Each row in the attention matrix sums to 1 and represents the weight or amount the current token is attending to all other tokens in the sentence. A higher attention value indicates higher importance or a stronger relationship compared to lower attention values. Since BERT has 12 attention heads we have one of these matrices per head. In addition, BERT has 12 hidden layers, each with a set of 12 matrices, for a total of 144 nxn attention weight matrices (Figure 18B).

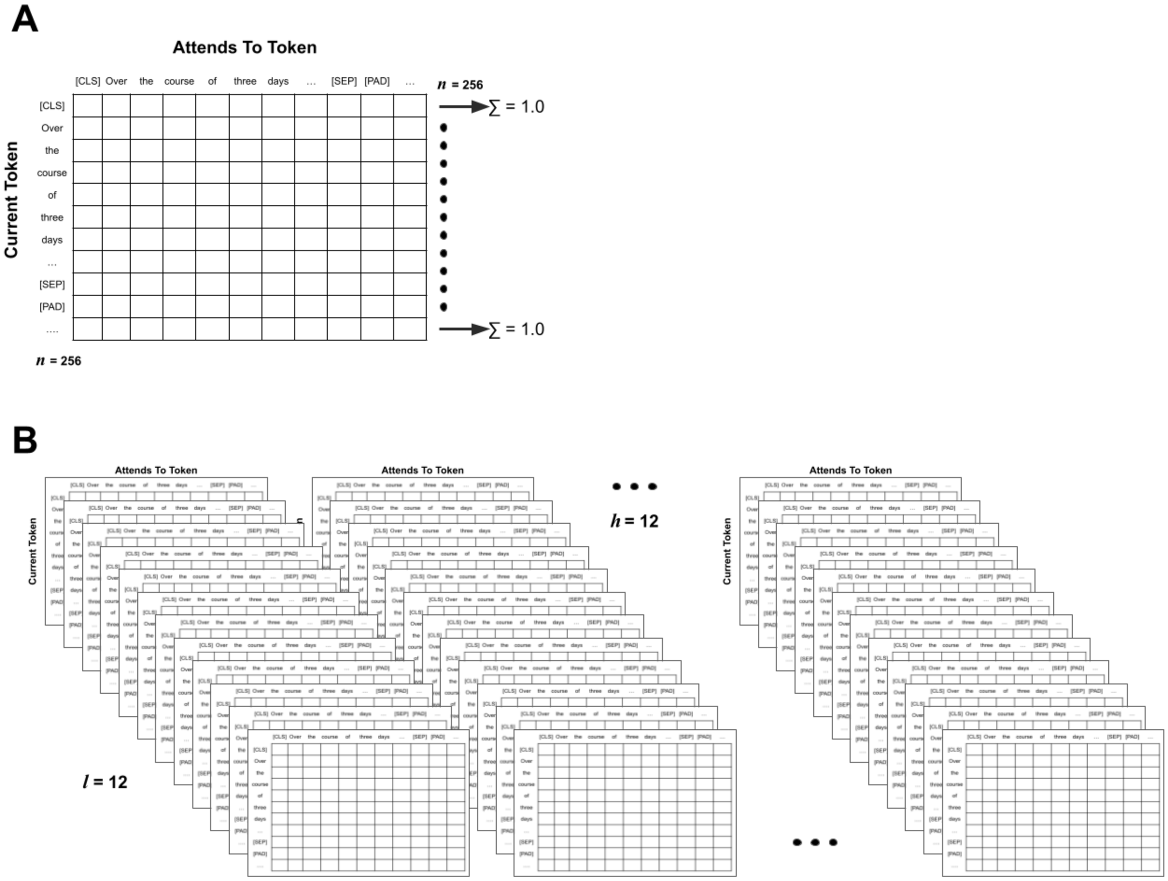


Fig. 18. Overview of attention architecture in BERT. A) Matrix of attention weights with sentences padded to $n = 256$ tokens. All columns for a given row sum to 1. B) Representation of the 12 attention heads, h, that each have 12 layers, l, of attention matrices.

6.3.2.2 Attention Summarization Algorithm

The attention matrices contain weights for how much a given token is attending to every other token in the sentence, including itself. Higher weights mean more attention is being paid to a specific token. Attention summarization returns an ordered list of which tokens an entire temporal phrase is attending to, sorted from highest to lowest. This requires summarization of each attention matrix for each head in each layer (i.e. 144 attention matrices). Figure 19 provides an overview of this method.

Briefly, the attention matrix for head h in layer l is subset to only those rows corresponding to the indices of the temporal phrase (Figure 19A, top). This matrix subset is then merged into a single vector by taking the maximum value at each position along the column axis (Figure 19A, bottom). This process is done for all 12 attention matrices in a given layer, which results in a matrix of summarized phrase attentions of dimensions 12 x n, where n is the number of tokens plus padding and special tokens in the sentence (set to 256 for this work, Figure 19B top). This matrix is then summed column-wise to obtain a single vector of length 1 x n (Figure 19B, bottom). Next, this summed vector has the phrase, [CLS], [SEP], period, and comma tokens set to a weight of zero (Figure 19C). This is done because a token generally has the strongest attention to itself, the [CLS] or [SEP] tokens, or separating punctuation like commas and periods. This is exacerbated after using the max and summation functions for summarization. As the goal is to identify what other tokens the phrase is paying attention to, we mask the phrase itself as well as other uninformative tokens from consideration by setting their weights to zero. Note, the [PAD] tokens are not masked. This is because some temporal phrases are the entire sentence; thus, allowing [PAD] tokens (which have the lowest weights anyway) ensures the algorithm does not return an empty list of attention tokens. After masking, the resulting 1 x n vector is normalized so that all values sum to 1 (Figure 19C, bottom). This process is then repeated for each layer to obtain another 12 x n matrix (Figure 19D, top). This matrix is also run through the summation and normalization process to obtain the final temporal phrase attention vector of size $1xn$ (Figure 19D, bottom). Token indices are sorted based on the summarized attention weights to obtain the top 3 tokens most attended to by the temporal phrase. The contextualized embeddings for these tokens are then utilized as features for the SVM and CNN classifiers.

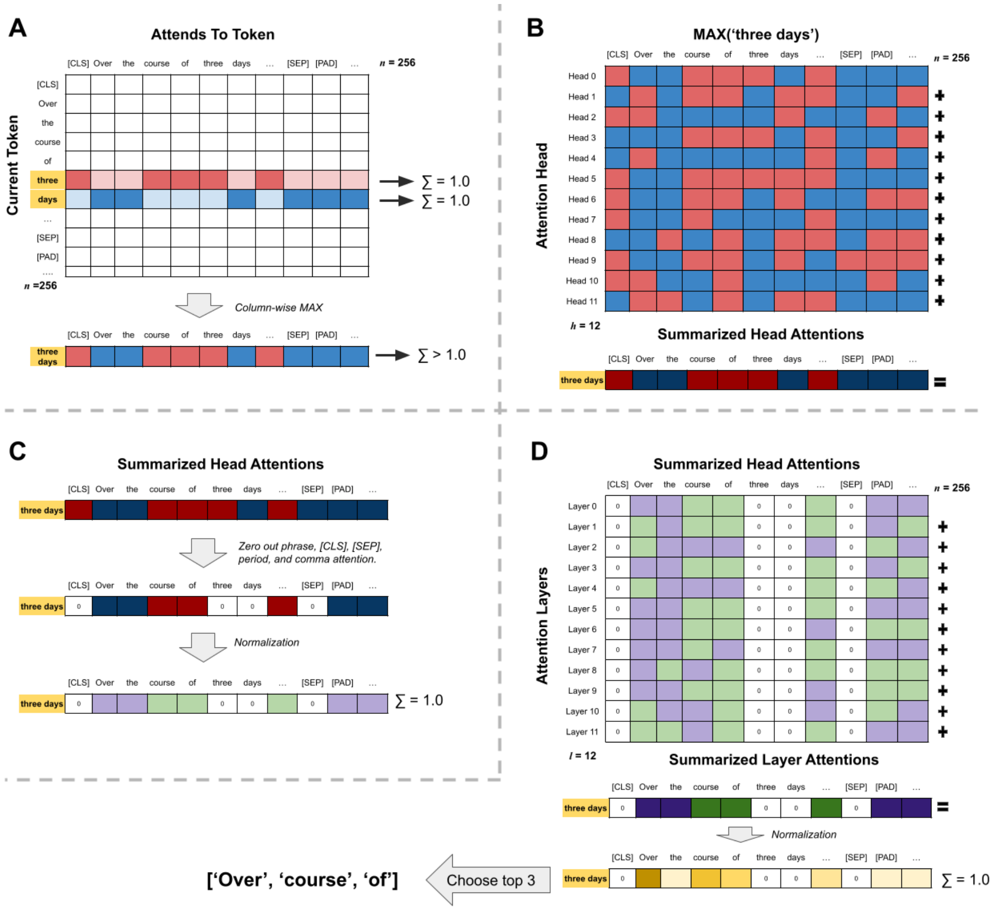


Fig. 19. Overview of attention summarization. A) Merge attention weight vectors for temporal phrase. B) Summarize head attentions. C) Filter and normalize summarized head attentions. D) Summarize layer attentions to get a single summarised attention vector for the temporal phrase.

6.4 Classifier Model Architecture and Training

In the following subsections the architecture and training for the learning classification models are described. These models are binary classification models that identify whether a temporal phrase is a DATE or DURATION temporal type using contextualized embeddings as features. SVM and CNN architectures are utilized, and each requires different formatting and summarization strategies for the contextualized embeddings obtained from the fine-tuned BERT models. First, the SVM architecture and training is described along with some additional contextualized embedding summarization methods needed in order to obtain a single input feature vector with consistent dimensions. Second, the CNN architecture and training is described.

6.4.1 SVM Model Architecture and Training

The primary problem addressed by this section is the classification of a RelIV-TIMEX as either a DATE or DURATION temporal type. We assume that other rules and algorithms have already identified the temporal phrase under question and have determined it to be a RelIV-TIMEX. However, before normalization can take place we need to determine its temporal type. As this is a binary choice, we chose to first evaluate how a classic Support Vector Machine (SVM) model performs on this task.

The SVM architecture requires a single feature vector per observation (the temporal phrase) as input, and outputs a 1 or -1 as the classification. For this work, DATE is set to the positive class, and DURATION is the negative class. As we are classifying a full temporal phrase and not just an individual token, we frequently have more than one contextualized embedding. Thus, the embeddings for a phrase need to be further summarized. Figure 17A shows the phrase summarization strategies that include the phrase only, the phrase plus the context before and after, and the phrase plus the summarized embeddings of the top 3 attention tokens. In addition to the feature extraction strategies, the contextualized embeddings are sourced from the baseline BertBase and ClinBioBERT models as well as models that were fine-tuned either on the binary classification task (Binary-BertBase/ClinBioBert), one of the Seq2Seq multi-class classifications tasks (Seq2Seq-BIO/Ttype-BertBase/ClinBioBert), or both (Binary-Seq2Seq-BIO/Ttype-BertBase/ClinBioBert) as summarized in Figure 14.

The following subsections describe how the contextualized embeddings are summarized for SVM training and prediction, including context and attention tokens, and the model training parameters.

6.4.1.1 TIMEX Representation for SVM Architecture

Previously, we described the summarization of each token's contextualized embeddings both by merging the embeddings in different hidden layers and by summarizing out-of-vocabulary tokens by using the last subword embedding. For SVM input, a single vector must be calculated. This vector must be a consistent length, which rules out concatenating embeddings because each TIMEX could be a different length. Thus, this work averages the summarized embeddings for tokens that map to a temporal phrase.

The representative phrase embedding is calculated by averaging all summarized token embeddings that are part of the phrase. This results in each temporal phrase being represented by a single numerical vector of length 3072 for use as a feature in the downstream classification models (Figure 17A, Phrase Only).

6.4.1.2 Incorporating Context

Contextualized embeddings, by definition, already contain some contextual information; however, *this work sets out to determine if adding an explicit summarized embedding for the context before and after a temporal phrase to the summarized phrase embedding will improve performance in downstream classification models.* A context window of up to 3 tokens before and 3 tokens after the temporal phrase was used. For each window (before/after) the summarized token embeddings were averaged using the same algorithm that was used for averaging the temporal phrase embedding (Algorithm **??**). These context embeddings are concatenated to the temporal phrase embedding sequentially (context before + phrase + context after) to create a single feature embedding of length 9216 (Figure 17A, Phrase +Context). If in the instance the temporal phrase is the entire sentence, or it is located at the beginning or end of a sentence, then the temporal phrase embedding is duplicated and used as the context. Additionally, if there are less than 3, but greater than 0, tokens in either of the before/after windows, then only those tokens are utilized in the summarized context embedding, thus the window is a minimum of 1 and maximum of 3 tokens.

6.4.1.3 Incorporating Attention

As with context, contextualized embeddings by definition already have attention weights represented; however, *this work sought out to summarize the attention weight matrices to identify specific tokens that the phrase as a whole is attending to, and add those embeddings to the SVM feature explicitly.* The identification of tokens being attended to has already been described previously in Section 6.3.2.2. For this work we take the top 3 attention tokens and average their embeddings. This summarized attention embedding is then concatenated to the end of the summarized phrase embedding, which results in a vector with length 6144 for use as a feature in the downstream classification models (Figure 17A, Phrase +Attention).

6.4.1.4 SVM Model Training

A total of 30 SVM models were trained using the extracted embeddings from each of the BertBase and ClinBioBert models shown in Figure 14A-C and Figure 14E in combination with the two Seq2Seq models (BIO and Ttype) and the 3 feature extraction strategies (Figure 17A; Phrase, +Context, +Attention). The DD-TIMEX

Training data set was used for model training and validation (Table 14). Hyper-parameter optimization was done using a grid search over the values listed in Table 19 with 5-fold validation and no limit on epochs.

Parameter	**Values Searched**
Kernel	Linear, RBF, Poly
C	0.1, 1, 10, 100
Degree	2,3,4
Gamma	0.0001, 0.001, 0.01, 0.1, 1, 10

Table 19. SVM Hyper-parameter Optimization

6.4.2 CNN Model Architecture and Training

Convolutional Neural Networks (CNNs) are feed-forward deep neural networks historically used for learning on images (computer vision) [147]; however, CNNs have also been successful in performing NLP tasks [148, 149]. Instead of operating on a 2-dimensional matrix, CNNs for NLP operate on 1-dimensional vectors, i.e. word embeddings. An advantage of using a CNN for NLP tasks include its ability to incorporate local structure into the classification via convolution and sub-sampling layers, such as taking into account adjacent words. For this work, the summarized embeddings described in Section 6.3.1 are sequentially input into the CNN (Figure 17B) for the temporal phrase, the phrase +Context, or the phrase +Attention to determine if accounting for the position of words can improve the binary classification task of temporal disambiguation between DURATION and DATE types. For the CNN implementation, the summarized embeddings are not averaged, but rather used as-is and just ordered as they appear in the sentence.

6.4.2.1 CNN Model Training

The CNN model architecture was implemented using the “KerasClassifier” wrapper for SciKitLearn in Python’s Tensorflow package. Briefly, it is composed of a 1-dimensional convolutional layer followed by max pooling layer, then another 1-dimensional convolutional layer. Next is a dropout layer, a flattening layer, and then 2 dense layers, the first outputting to 10 nodes with the ReLU activation, and the last outputting one node with the sigmoid activation. Hyper-parameters searched over for each layer are listed in Table 20.

Parameter	**Values Searched**
Filters	32,64,128
First Kernel Size	3,5
Second Kernel Size	2,3
Pooling Size	2,3
Stride	1,2
Dropout	0.05, 0.10

Table 20. CNN Hyper-parameter Optimization

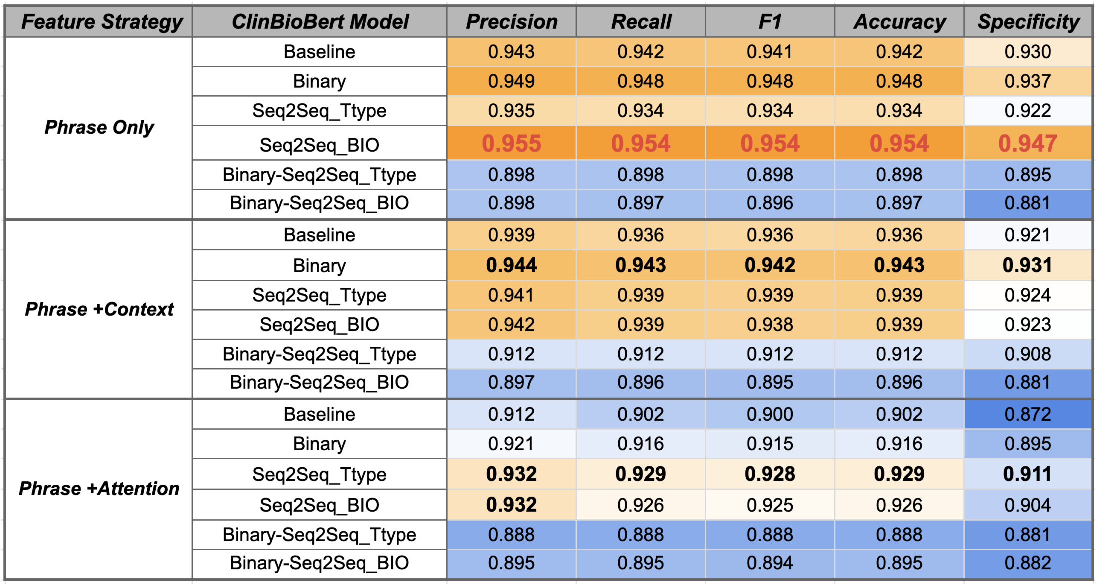
6.5 SVM and CNN Classifier Results

Performance of the SVM and CNN models was evaluated in several phases. First, the temporal phrases defined by the annotations in the RelIV-TIMEX Gold Standard Evaluation data set (Section 6.1.2) are used to build the features, i.e. no temporal phrase recognition is performed. Second, the models from the top performing strategy plus the baseline models are then integrated into Chrono and evaluated using Chrono’s temporal phrase recognition algorithm against the RelIV-TIMEX Gold Standard. Finally, End-2-End evaluation is performed using the best strategy and compared to state-of-the-art results on the complete i2b2 Evaluation data set. Due to the stochastic nature of CNN models, reported results are the average scores across 5 duplicate models using the same hyper-parameters (see legends of Supplementary Tables S4-S15). For all evaluations the metrics Precision, Recall, F1, Accuracy, and Specificity are calculated using the TIMEX type classification (i.e. DATE or DURATION) to evaluate performance on a specific data set (see Equations 2.2-2.6). The weighted average (Equation 2.7) uses the system-specific number of DATE or DURATION instances as the weights for each metric across the DATE and DURATION results, and is used for ranking.

In the following subsections, the results of each evaluation phase are provided with discussion. All performance scores and confusion matrices for each model can be found in the Appendix in Supplementary Tables S16-S17 for SVM and S4-S15 for CNN.

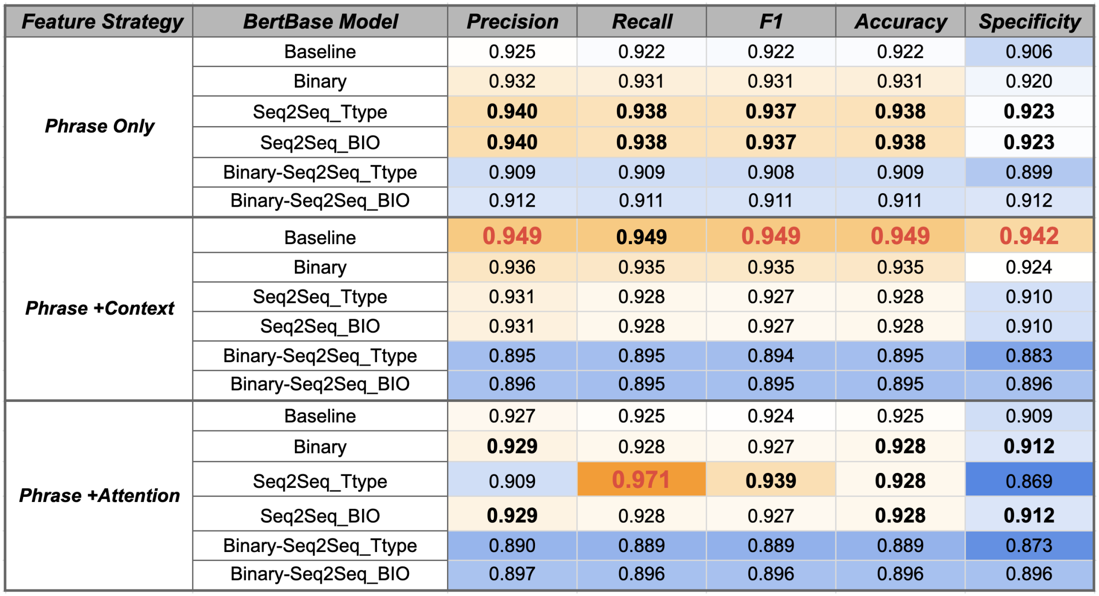
6.5.1 Evaluation Phase 1: Using RelIV-TIMEX Gold Standard Temporal Phrases

The SVM and CNN model variations were first evaluated using the temporal phrases from the RelIV-TIMEX Gold Standard evaluation data set as input, which contains 429 DATE types and 307 DURATION types (Section 6.1.2). The SVM results for ClinBioBert and BertBase model variants are shown in Tables 21 and 22, respectively. Likewise, CNN results are shown in Tables 23 and 24. Main findings and conclusions are discussed below.



Feature Strategy	ClinBioBert Model	Precision	Recall	F1	Accuracy	Specificity
Phrase Only	Baseline	0.943	0.942	0.941	0.942	0.930
	Binary	0.949	0.948	0.948	0.948	0.937
	Seq2Seq_Ttype	0.935	0.934	0.934	0.934	0.922
	Seq2Seq_BIO	0.955	0.954	0.954	0.954	0.947
	Binary-Seq2Seq_Ttype	0.898	0.898	0.898	0.898	0.895
	Binary-Seq2Seq_BIO	0.898	0.897	0.896	0.897	0.881
Phrase +Context	Baseline	0.939	0.936	0.936	0.936	0.921
	Binary	**0.944**	**0.943**	**0.942**	**0.943**	**0.931**
	Seq2Seq_Ttype	0.941	0.939	0.939	0.939	0.924
	Seq2Seq_BIO	0.942	0.939	0.938	0.939	0.923
	Binary-Seq2Seq_Ttype	0.912	0.912	0.912	0.912	0.908
	Binary-Seq2Seq_BIO	0.897	0.896	0.895	0.896	0.881
Phrase +Attention	Baseline	0.912	0.902	0.900	0.902	0.872
	Binary	0.921	0.916	0.915	0.916	0.895
	Seq2Seq_Ttype	**0.932**	**0.929**	**0.928**	**0.929**	**0.911**
	Seq2Seq_BIO	**0.932**	0.926	0.925	0.926	0.904
	Binary-Seq2Seq_Ttype	0.888	0.888	0.888	0.888	0.881
	Binary-Seq2Seq_BIO	0.895	0.895	0.894	0.895	0.882

Table 21. ClinBioBert SVM performance using the Gold Standard RelIV-TIMEX Evaluation data set. Scores are weighted averages across DATE and DURATION. Bold = best performance within Feature Strategy; Red = best performance across all SVM models; Orange = high, white = median, and blue = low scores relative to all scores in the table.



Feature Strategy	BertBase Model	Precision	Recall	F1	Accuracy	Specificity
Phrase Only	Baseline	0.925	0.922	0.922	0.922	0.906
	Binary	0.932	0.931	0.931	0.931	0.920
	Seq2Seq_Ttype	**0.940**	**0.938**	**0.937**	**0.938**	**0.923**
	Seq2Seq_BIO	**0.940**	**0.938**	**0.937**	**0.938**	**0.923**
	Binary-Seq2Seq_Ttype	0.909	0.909	0.908	0.909	0.899
	Binary-Seq2Seq_BIO	0.912	0.911	0.911	0.911	0.912
Phrase +Context	Baseline	0.949	**0.949**	0.949	0.949	0.942
	Binary	0.936	0.935	0.935	0.935	0.924
	Seq2Seq_Ttype	0.931	0.928	0.927	0.928	0.910
	Seq2Seq_BIO	0.931	0.928	0.927	0.928	0.910
	Binary-Seq2Seq_Ttype	0.895	0.895	0.894	0.895	0.883
	Binary-Seq2Seq_BIO	0.896	0.895	0.895	0.895	0.896
Phrase +Attention	Baseline	0.927	0.925	0.924	0.925	0.909
	Binary	**0.929**	0.928	0.927	**0.928**	**0.912**
	Seq2Seq_Ttype	0.909	0.971	**0.939**	**0.928**	0.869
	Seq2Seq_BIO	**0.929**	0.928	0.927	**0.928**	**0.912**
	Binary-Seq2Seq_Ttype	0.890	0.889	0.889	0.889	0.873
	Binary-Seq2Seq_BIO	0.897	0.896	0.896	0.896	0.896

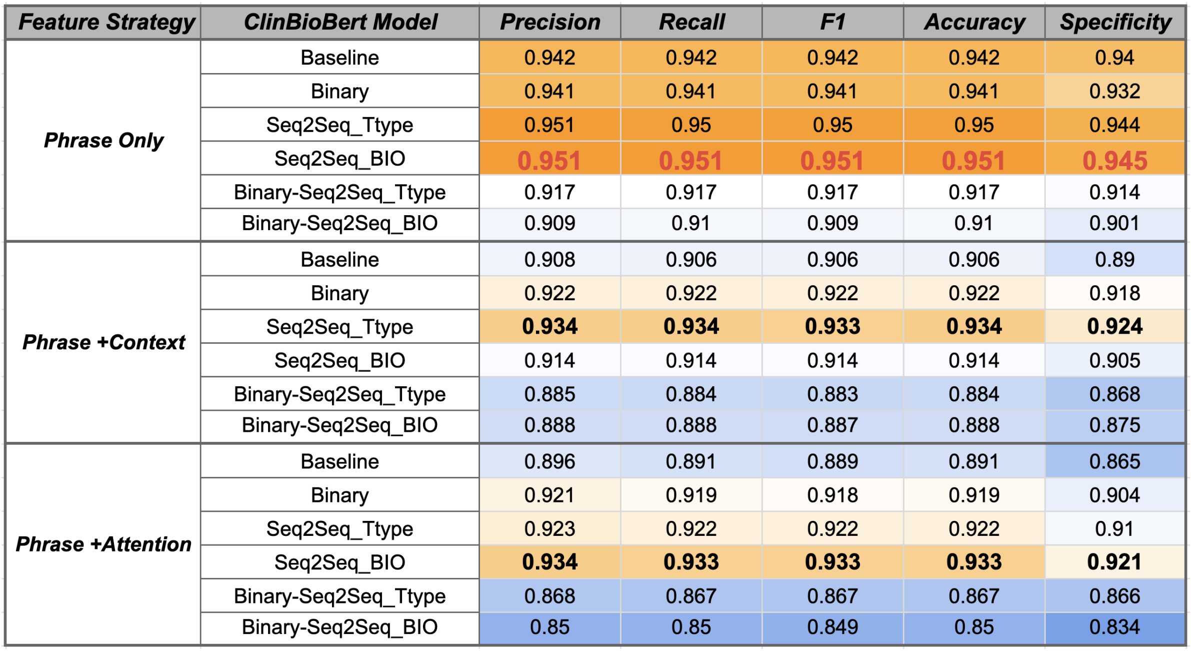
Table 22. BertBase SVM performance using the Gold Standard RelIV-TIMEX Evaluation data set. Refer to Table 21 for metric and color coding descriptions.

6.5.1.1 Temporal Fine-Tuning on a Single Temporal Task Improves Performance

Fine-tuning the ClinBioBert model on either the binary temporal sentence classification task or the multi-label Seq2Seq temporal type classification improves the SVM and CNN model classification performance from the respective baseline models (Tables 21 and 23, respectively). Specifically, the ClinBioBert-Seq2Seq-BIO fine tuning strategy (Figure 14F) achieves the highest F1 results for both the SVM and CNN models (SVM F1 = 0.954, CNN F1 = 0.951), with the SVM model coming out on top (Table 21).

In contrast, continually fine-tuning the baseline ClinBioBert model, or chaining fine-tuning tasks, first on the binary task followed by the Seq2Seq task (Figure 14C) actually results in a substantial degradation of performance with the majority of F1 scores across both SVM and CNN models being less than 0.900. The pattern of improved performance after fine-tuning on a single temporal task and degraded performance after chaining fine-tuning tasks holds true for all feature selection strategies (Phrase Only, Phrase+Context, Phrase+Attention).

For both ClinBioBert and BertBase baseline models, fine tuning on a more complex temporal task (Seq2Seq temporal type classification) versus a more simplistic task (binary temporal sentence classification) generally results in better performance for the Phrase Only and Phrase+Attention feature selection strategies in both the SVM and CNN models, except for the BertBase Phrase Only strategy where the inverse is true (Table 24). This also holds true when comparing the more complex BIO Seq2Seq task versus the more simplistic Ttype task where the BIO task returns embeddings that result in better performance. For the Phrase+Context models, the simpler binary fine tuning task results in features that outperform both Seq2Seq strategies in the SVM, and one of the 2 strategies in the CNN models. Overall, these results indicate that fine-tuning on a single, yet complex (Seq2Seq BIO), temporal task creates contextualized embeddings that are more relevant to the temporal disambiguation task.

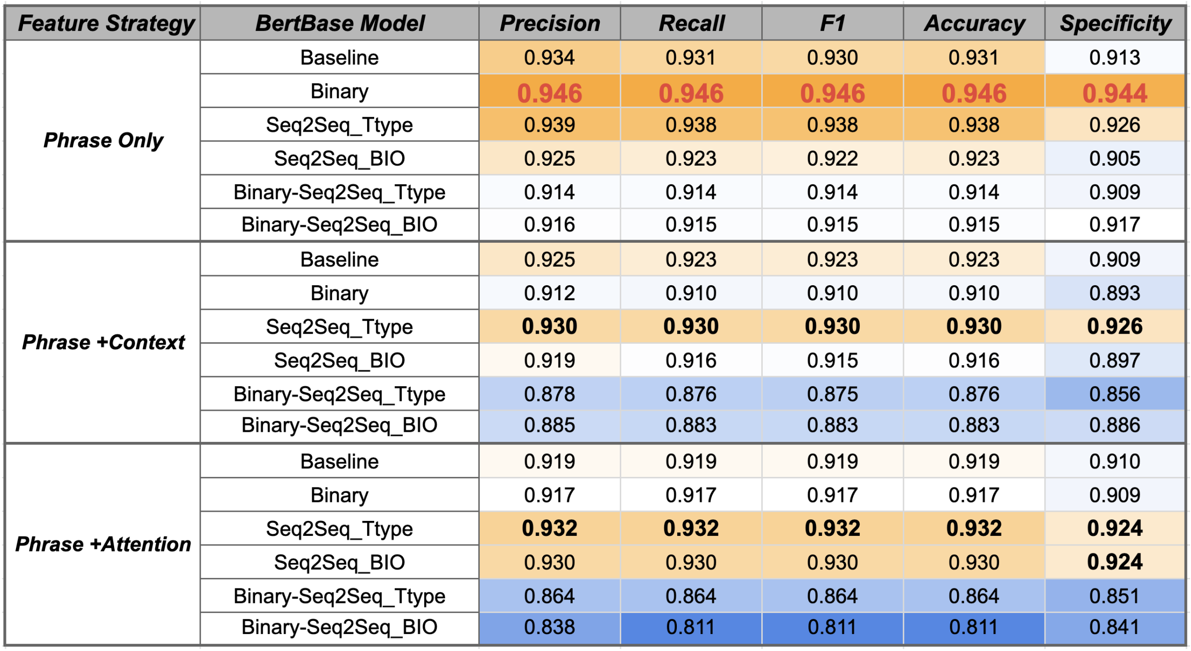


Feature Strategy	***ClinBioBert Model***	***Precision***	***Recall***	***F1***	***Accuracy***	***Specificity***
Phrase Only	Baseline	0.942	0.942	0.942	0.942	0.94
	Binary	0.941	0.941	0.941	0.941	0.932
	Seq2Seq_Ttype	0.951	0.95	0.95	0.95	0.944
	Seq2Seq_BIO	**0.951**	**0.951**	**0.951**	**0.951**	**0.945**
	Binary-Seq2Seq_Ttype	0.917	0.917	0.917	0.917	0.914
	Binary-Seq2Seq_BIO	0.909	0.91	0.909	0.91	0.901
Phrase +Context	Baseline	0.908	0.906	0.906	0.906	0.89
	Binary	0.922	0.922	0.922	0.922	0.918
	Seq2Seq_Ttype	**0.934**	**0.934**	**0.933**	**0.934**	**0.924**
	Seq2Seq_BIO	0.914	0.914	0.914	0.914	0.905
	Binary-Seq2Seq_Ttype	0.885	0.884	0.883	0.884	0.868
	Binary-Seq2Seq_BIO	0.888	0.888	0.887	0.888	0.875
Phrase +Attention	Baseline	0.896	0.891	0.889	0.891	0.865
	Binary	0.921	0.919	0.918	0.919	0.904
	Seq2Seq_Ttype	0.923	0.922	0.922	0.922	0.91
	Seq2Seq_BIO	**0.934**	**0.933**	**0.933**	**0.933**	**0.921**
	Binary-Seq2Seq_Ttype	0.868	0.867	0.867	0.867	0.866
	Binary-Seq2Seq_BIO	0.85	0.85	0.849	0.85	0.834

Table 23. ClinBioBert CNN performance using the Gold Standard RelIV-TIMEX Evaluation data set. Refer to Table 21 for metric and color coding descriptions.

6.5.1.2 Adding Context can Help BertBase Embeddings Compensate for Domain Shifts

As discussed, additional fine-tuning on a single temporal task improves performance for the ClinBioBert models and the BertBase Phrase Only and Phrase+Attention feature selection strategies. However, the inverse is true for the Phrase+Context BertBase models where any type of fine tuning degrades performance from the baseline model for both SVM and CNN classifiers (Table 22 and 24, respectively). Overall, the



Feature Strategy	***BertBase Model***	***Precision***	***Recall***	***F1***	***Accuracy***	***Specificity***
Phrase Only	Baseline	0.934	0.931	0.930	0.931	0.913
	Binary	**0.946**	**0.946**	**0.946**	**0.946**	**0.944**
	Seq2Seq_Ttype	0.939	0.938	0.938	0.938	0.926
	Seq2Seq_BIO	0.925	0.923	0.922	0.923	0.905
	Binary-Seq2Seq_Ttype	0.914	0.914	0.914	0.914	0.909
	Binary-Seq2Seq_BIO	0.916	0.915	0.915	0.915	0.917
Phrase +Context	Baseline	0.925	0.923	0.923	0.923	0.909
	Binary	0.912	0.910	0.910	0.910	0.893
	Seq2Seq_Ttype	**0.930**	**0.930**	**0.930**	**0.930**	**0.926**
	Seq2Seq_BIO	0.919	0.916	0.915	0.916	0.897
	Binary-Seq2Seq_Ttype	0.878	0.876	0.875	0.876	0.856
	Binary-Seq2Seq_BIO	0.885	0.883	0.883	0.883	0.886
Phrase +Attention	Baseline	0.919	0.919	0.919	0.919	0.910
	Binary	0.917	0.917	0.917	0.917	0.909
	Seq2Seq_Ttype	**0.932**	**0.932**	**0.932**	**0.932**	**0.924**
	Seq2Seq_BIO	0.930	0.930	0.930	0.930	**0.924**
	Binary-Seq2Seq_Ttype	0.864	0.864	0.864	0.864	0.851
	Binary-Seq2Seq_BIO	0.838	0.811	0.811	0.811	0.841

Table 24. BertBase CNN performance using the Gold Standard RelIV-TIMEX Evaluation data set. Refer to Table 21 for metric and color coding descriptions.

BertBase Phrase+Context SVM classifier is the highest F1 out of all combinations (Table 22) with an F1 score of 0.949, which is not far behind the best ClinBioBert SVM classifier with an F1 of 0.954. This could be the result of the ClinBioBert models already containing the needed context in the embeddings as this model was essentially created from chaining fine-tuning tasks on biomedical and clinical texts. Thus, incorporating context explicitly may be adding too much noise. However, the BertBase model has no clinical or biomedical information already embedded; thus, explicitly including context into the extracted features from the unmodified BertBase embeddings seems to help it compensate for a domain shift.

6.5.1.3 Context and Attention Degrade Performance

Adding in the context or attention progressively degrades the baseline ClinBioBert performance for SVM and CNN classifiers (Phrase Only F1= 0.941, Phrase

+Context F1=0.936, Phrase +Attention F1=0.900), and this pattern generally holds true for all fine-tuned variations of the ClinBioBert model for both the SVM and CNN classifiers (Table 21, 23).

This also holds for the SVM classifiers using the fine-tuned BertBase model variations in general, with some mixed results for the Seq2Seq models (e.g. BertBase Seq2Seq Ttype, Table 22). However, for the baseline BertBase SVM classifier, adding context or the attention tokens as part of the feature vector improve performance with the +Context model achieving the best BertBase results overall (Phrase Only F1=0.922, Phrase +Context F1 = 0.949, Phrase +Attention F1 = 0.924). Finally, for the CNN classifiers using the BertBase models, the baseline plus all fine tuning variations have decreased performance when adding in context and attention, except for the BertBase Seq2Seq BIO model, which sees improved performance (Table 24).

Interestingly, the CNN classifier using the ClinBioBert Phrase Only model outperforms the BertBase equivalent, but by a smaller margin compared to the SVM classifiers (BertBase F1 = 0.930, ClinBioBert F1 = 0.942, Table 23,24). Adding in context and attention degrades the performance of both, but with a greater effect on the ClinBioBert model. Specifically, the F1 delta between the Phrase Only model and Phrase+Attention model for ClinBioBert is 0.053 and the F1 delta for BertBase is 0.011.

A possible reason for the degradation of results when context or attention is added is that the model may be dealing with too much information so that it washes out the differences between DATE and DURATION phrases. Another possibility is that this is causing the algorithm to pay too much attention to the context or attention tokens as the contextualized embeddings already contain some of this information. This may also be the reason why we see mixed results for the SVM classifiers using the BertBase models as these models need the additional contextual information to perform well. Figure 20 shows the difference (+Attention - Phrase Only) in the top 5 most frequent attention tokens for the classifications returned by the best performing SVM model, ClinBioBert Seq2Seq BIO (Table 21, Figure 20A), and its CNN (Figure 20B) and BertBase equivalents (SVM, Figure 20C; CNN, Figure 20D). A negative value means the token appeared less frequently in the attention list for phrases classified as a DATE/DURATION, and a positive value indicates the opposite. For example, in Figure 20A, the most frequently attended to token by phrases classified as a DURATION is the token "for". This token has a delta value of -3 for DURATION indicating that when you add attention to the SVM feature 3 phrases that attend to the term "for" that were classified as a DURATION when using the Phrase Only strategy are now being classified as a DATE. Likewise, for phrases classified as a DATE the most frequently attended to token is "on" and its delta value is a +2, which means including attention into the feature vector results in 2 additional phrases that used to be classified as a DURATION now being classified as a DATE.

Interestingly, the "[subword]" token seems to be a major focus when using ClinBioBert as a starting model. For this analysis, all subwords that started with the double hash ("##") were replaced by the single token "[subword]". In the ClinBioBert vocabulary, subwords are usually associated with clinical entities such as procedures or symptoms. Thus, if a subword is being attended to by a temporal phrase, it will appear in the top 3 attention tokens, and may indicate that the procedure, symptom, or other clinical entity is associated with the temporal phrase. Figure 20A and B show that, when counted together, subwords are among the top 5 most frequent tokens in the phrases annotated as a DATE or DURATION. Interestingly, when adding in the attention vector as a feature for SVM and CNN classification phrases attending to these subwords that were classified as a DURATION are given a DATE classification. This may mean that adding in attention is causing the models to pay too much attention to these subwords with a preference to classify these phrases as a DATE. In general, for the ClinBioBert Seq2Seq BIO SVM and CNN models, adding in attention seems to bias the classification towards a DATE as the top most frequent DURATION tokens consistently loose phrases while the top DATE tokens gain phrases. Comparing this to using BertBase as the starting model, the subword token barely even makes the top most 5 frequent attention tokens (ranks fifth for the DATE class only), which may indicate BertBase is not putting much emphasis on these clinical entities as the ClinBioBert models do. Additionally, the opposite trend is seen for the BertBase Seq2Seq BIO CNN model (Figure 20D), where the DURATION delta is positive and DATE is mostly negative indicating adding in the attention vector as a feature is causing more phrases that attend to tokens like "at" to be classified as a DURATION where they were classified as a DATE when using the Phrase Only feature vector.

6.5.1.4 Bias Towards DATE Classifications

To identify which temporal type the models are performing better on we looked at the confusion matrices for the top performing SVM and CNN ClinBioBert Seq2Seq BIO models (full confusion matrices for all models are in Supplementary Tables S4-S17). The top performing SVM and CNN models tend to have a bias towards classifying a temporal phrase as a DATE type (Table 25) with the SVM misclassifying 25 DURATION types as a DATE versus misclassifying only 11 DATE types. Similarly, the CNN misclassified 23.6 DURATION types on average (across 5 replicate models) compared to 15.2 DATE types. Taking the average number of instances a DATE or DURATION is misclassified across all SVM and CNN models results in a similar trend (Table 25) where DURATION types are close to 2 times more likely to be misclassified as a DATE versus a DATE to be misclassified as a DURATION.

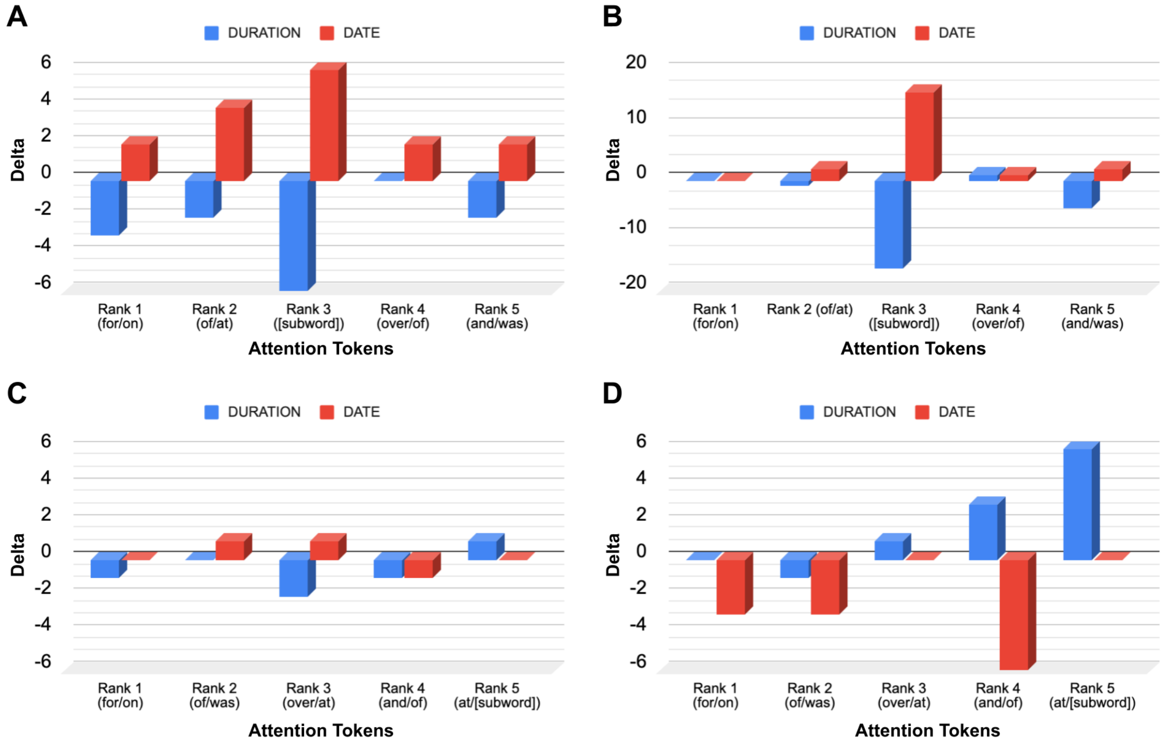


Fig. 20. Difference in frequencies of the top 5 most frequent attention tokens for temporal phrases classified as a DURATION (blue) or DATE (red). Top 5 most frequent attention tokens were identified using the gold standard DATE/DURATION classifications. The delta value (y-axis) represents the difference in the frequency of these tokens when using the +Attention versus Phrase Only feature strategies (+Attention - Phrase Only) to classify temporal phrases as a DATE or DURATION. The x-axis lists the rank and the top term for DURATION and DATE in that order, unless the same term was ranked the same across both DURATION and DATE classifications, in which case only one term is listed. A) ClinBioBert Seq2Seq BIO SVM classification, B) ClinBioBert Seq2Seq BIO CNN classification, C) BertBase Seq2Seq BIO SVM classification, D) BertBase Seq2Seq BIO CNN classification.

This indicates that even the top performing models have a bias towards classifying a temporal phrase as a DATE type. This is likely the result of the imbalance in the DD-TIMEX training data set as there are 4 times as many DATE types than

DURATIONs (Table 14).

	Misclassification	
Model	DURATION as a DATE	DATE as a DURATION
ClinBioBert Seq2Seq BIO SVM	25	11
ClinBioBert Seq2Seq BIO CNN	23.6	15.2
All SVM Model Average	43.1	19.2
All CNN Model Average	46.6	24.8

Table 25. Frequency of DATE and DURATION temporal type misclassification using the RelIV-TIMEX Gold Standard temporal phrases. Comparison of the ClinBioBert Seq2Seq BIO SVM and CNN model misclassifications to the average frequency across all SVM and CNN models.

6.5.1.5 Best Strategy and Model

Overall, the best performing classifier is the SVM using the ClinBioBert Seq2Seq BIO model with an F1 score of 0.954. The ClinBioBert and BertBase Seq2Seq Ttype and binary fine-tuned models also performed well when used in SVM or CNN classifiers; thus, these 6 models plus the respective baselines were moved forward to the next phase of evaluation that includes integration with the Chrono temporal phrase recognition algorithm and comparison to state-of-the-art systems that participated in the i2b2 challenge.

6.5.2 Evaluation Phase 2: Integration of the Temporal Disambiguation Module into Chrono.

The ClinBioBert-Seq2Seq-BIO SVM model was found to perform the best when using the gold standard temporal phrases; however, temporal phrase recognition algorithms do not always identify the exact phrase annotated in a gold standard. Thus, the next evaluation phase integrated the temporal disambiguation model into Chrono to utilize Chrono's temporal phrase recognition algorithm. Figure 21 is a re-production of Figure 8 with the temporal disambiguation module shown in the workflow. Specifically, Chrono identifies and classifies temporal phrases using the fine-grained SCATE Schema. It then converts these SCATE annotations into TimeML formatted annotations. If an entity is identified as a Period or Calendar-Interval during the conversion process, it is sent to the Temporal Disambiguation module where it is classified as a DATE or DURATION type. Depending on the temporal type identified, the phrase is then sent to the TimeML Normalization module before being output to an XML file.

In this phase the performance of Chrono is still being compared to the RelIV-TIMEX Gold Standard. Since Chrono identifies all temporal expression types, the results have to be filtered to only those that overlap the RelIV-TIMEX Gold Standard. The following sub sections discuss how results from Chrono and the 3 state-of-the-art i2b2 systems were filtered to obtain a fair comparison to the RelIV-TIMEX Gold Standard. Then the performance of Chrono using the temporal disambiguation module is reported along with a comparison to the state-of-the-art RelIV-TIMEX performance.

6.5.2.1 Creating a Fair Comparison to the RelIV-TIMEX Data Set

Previously, the temporal disambiguation module was evaluated only on RelIV-TIMEXs in the RelIV-TIMEX data set. In order to have a fair comparison, a Python script was written to filter Chrono and state-of-the-art system results to only those that overlapped with a temporal phrase in the RelIV-TIMEX evaluation data set. This resulted in a varying number of DATE and DURATION types for each system due to the systems breaking up the gold standard phrases into multiple phrases. For example, the gold standard phrases “the morning on the day” and “hospital day 2 through hospital day 3” were generally broken up into multiple phrases by one or more of the systems. The total resulting DATE and DURATION phrase numbers are

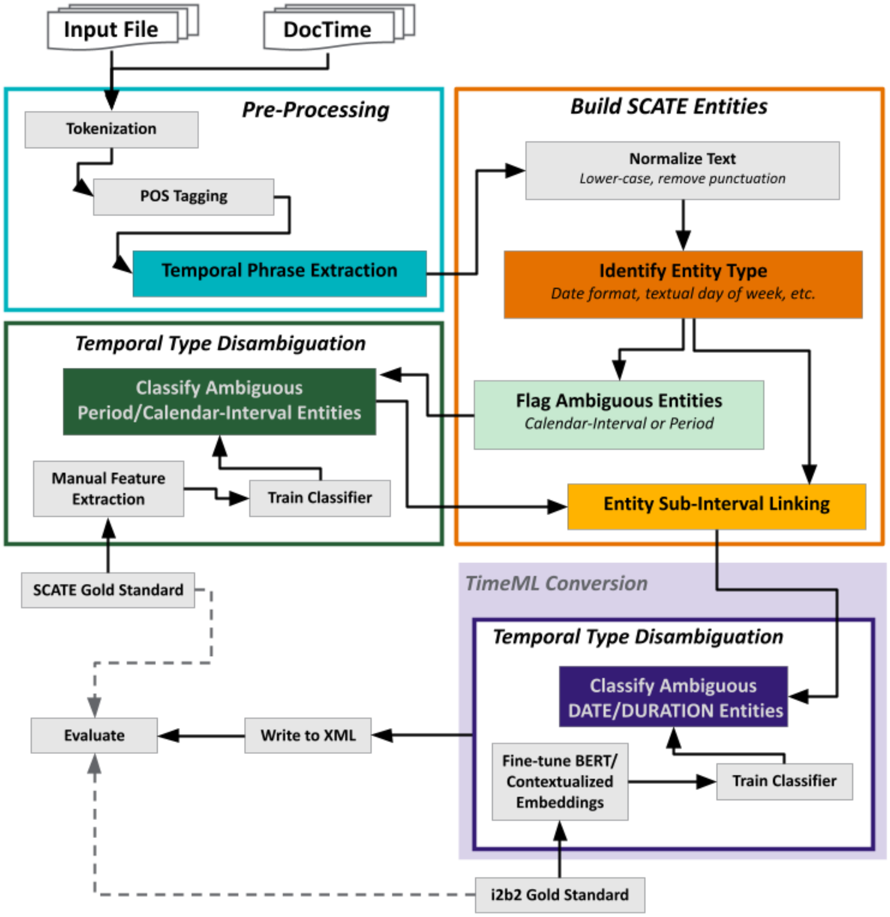

Input File
DocTime
Pre-Processing
Tokenization
POS Tagging
Temporal Phrase Extraction
Build SCATE Entities
Normalize Text
Lower-case, remove punctuation
Identify Entity Type
Date format, textual day of week, etc.
Flag Ambiguous Entities
Calendar-Interval or Period
Entity Sub-Interval Linking
Temporal Type Disambiguation
Classify Ambiguous Period/Calendar-Interval Entities
Manual Feature Extraction
Train Classifier
SCATE Gold Standard
TimeML Conversion
Temporal Type Disambiguation
Classify Ambiguous DATE/DURATION Entities
Fine-tune BERT/ Contextualized Embeddings
Train Classifier
i2b2 Gold Standard
Evaluate
Write to XML

Fig. 21. Chrono architecture with the DATE/DURATION Temporal Disambiguation Module.

listed in Table 26.

6.5.2.2 Improved Performance with Temporal Disambiguation Module

Integrating any of the temporal disambiguation models from the previous section into Chrono results in significant performance improvement (Table 27, top row vs "Chrono+TTD"). Previously, Chrono had a naive rule that assigned all SCATE Period and Calendar-Interval types to a TimeML DURATION type. This resulted in poor performance with a weighted F1 value of 0.361. As expected, all +TTD (Tem-

System	DATE	DURATION
Gold	429	307
Chrono	463	337
Mayo	455	335
Vanderbilt	458	337
MSRA	454	337

Table 26. Number of relative temporal phrases in state-of-the-art systems, Chrono, and the RelIV-TIMEX Gold Standard.

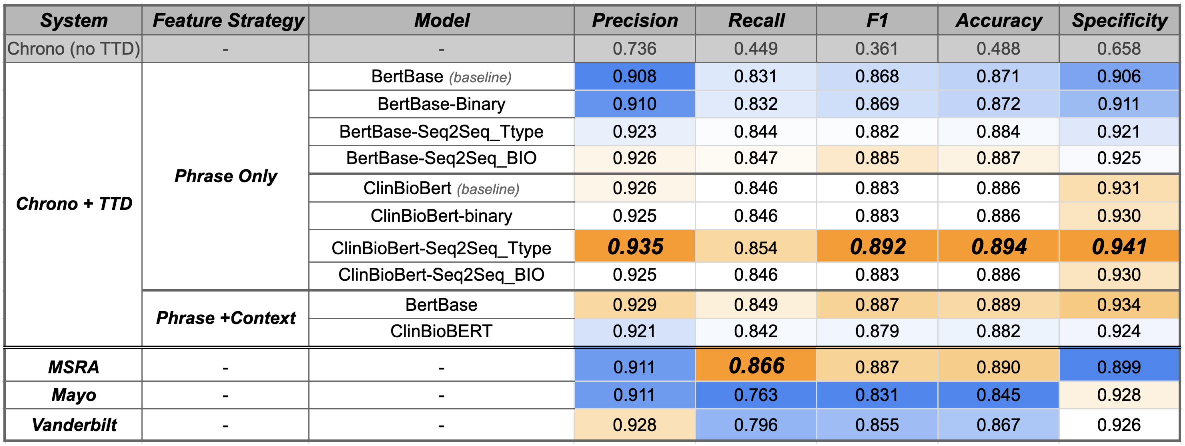
poral Type Disambiguation) variations improved on this baseline performance. The ClinBioBert models, overall, performed better than the BertBase models with the ClinBioBert Seq2Seq Ttype model achieving the best F1 score of 0.894 on the RelIV-TIMEX Evaluation data set. Interestingly, when using BertBase as the initiation model, fine-tuning on progressively more complex tasks (i.e. binary to Seq2Seq-Ttype to Seq2Seq-BIO) also continually improved performance over the baseline model. This same observation does not hold when using the ClinBioBert model as the initial model as the binary and Seq2Seq-BIO fine-tuning performed similarly to baseline while the Seq2Seq-Ttype fine-tuning resulted in the top performing model with a weighted F1 score of 0.892 (Table 27).

All of the models just discussed utilized the Phrase Only feature strategy because it was observed that adding in context or attention terms degraded performance. When adding context and attention, the same degradation of performance is observed as discussed in Section 6.5.1.3. This was the case for all except the BertBase model where adding context improved performance significantly. Thus, the baseline models plus context were run with Chrono to see if the improved BertBase performance held. Indeed, the BertBase +Context model, without any fine tuning, actually achieves the second highest performance with an F1 score of 0.887. Curiously, the same strategy of adding context to the ClinBioBert baseline model actually degrades the performance compared to the Phrase Only feature strategy with an F1 of 0.879 versus 0.883, respectively. The Seq2Seq BIO module increased the weighted F1 score to 0.883. Surprisingly, while the Seq2Seq-BIO model outperformed Seq2Seq Ttype on the RelIV-TIMEX Gold Standard phrases, the Seq2Seq-Ttype model performs the best when integrated into Chrono.

6.5.2.3 Chrono Outperforms State-of-the-Art Systems on Relative Temporal Expression Disambiguation

While it is good to know performance has improved with the new temporal disambiguation module, its performance needs to be compared with the other state-of-the-art systems on the same data set. For a fair comparison, the same filtering script was used on the state-of-the-art system results to obtain only those that overlap with the RelIV-TIMEX evaluation data set (Table 26). The bottom 3 rows of Table 27 contain the results of the RelIV-TIMEX evaluation on the state-of-the-art systems. Except for Recall, Chrono plus the ClinBioBert-Seq2Seq-Ttype module achieves the highest performance for all other metrics, including the best F1 score of 0.892 compared to the top state-of-the-art system, MSRA, with an F1 of 0.887. The MSRA system achieves the highest Recall of 0.866, however, this is offset by a lower Precision of 0.911 compared to Chrono's Precision of 0.935. Additionally, all of the ClinBioBert models exceed the Mayo and Vanderbilt performances for the majority of the metrics, with the BertBase modules seeing higher Precision and Specificity after fine tuning.

Comparing the confusion matrices of Chrono's best performing model and MSRA, which is also a hybrid system, reveals that Chrono is better at classifying DURATION type phrases than MSRA, which is contrary to it's performance when using the RelIV-TIMEX Gold Standard phrases (Section 6.5.1.4 and Supplementary Tables S4-S17).



System	Feature Strategy	Model	Precision	Recall	F1	Accuracy	Specificity
Chrono (no TTD)	-	-	0.736	0.449	0.361	0.488	0.658
Chrono + TTD	***Phrase Only***	BertBase *(baseline)*	0.908	0.831	0.868	0.871	0.906
		BertBase-Binary	0.910	0.832	0.869	0.872	0.911
		BertBase-Seq2Seq_Ttype	0.923	0.844	0.882	0.884	0.921
		BertBase-Seq2Seq_BIO	0.926	0.847	0.885	0.887	0.925
		ClinBioBert *(baseline)*	0.926	0.846	0.883	0.886	0.931
		ClinBioBert-binary	0.925	0.846	0.883	0.886	0.930
		ClinBioBert-Seq2Seq_Ttype	***0.935***	0.854	***0.892***	***0.894***	***0.941***
		ClinBioBert-Seq2Seq_BIO	0.925	0.846	0.883	0.886	0.930
	Phrase +Context	BertBase	0.929	0.849	0.887	0.889	0.934
		ClinBioBERT	0.921	0.842	0.879	0.882	0.924
MSRA	-	-	0.911	***0.866***	0.887	0.890	0.899
Mayo	-	-	0.911	0.763	0.831	0.845	0.928
Vanderbilt	-	-	0.928	0.796	0.855	0.867	0.926

Table 27. System performance on the RelIV-TIMEX evaluation data set of Chrono before and after the TTD model integration, and the three i2b2 state-of-the-art system. Values are the weighted average across individual DATE and DURATION performance. Cell colors are as described in Table 21 except the maximum and minimum are relative to each column instead of the entire table.

Chrono has a low misclassification of only 19 phrases (Table 28) compared to MSRAs 48 (Table 29). Additionally, the confusion matrices and the overall Recall score show that MSRA is identifying more relative temporal phrases overall with 38 "na" values versus Chrono's 68. This, however, is a function of Chrono's temporal phrase recognition algorithm, which isn't affected by the TTD module. Thus, improvement in Chrono's recognition algorithm should increase performance even further.

		Chrono+ClinBioBert-Seq2Seq-Ttype		
		DATE	**DURATION**	**na**
Gold	**DATE**	383	29	48
	DURATION	19	298	20

Table 28. Confusion matrix for Chrono+ClinBioBert-Seq2Seq-Ttype using the RelIV--TIMEX Evaluation data set.

		MSRA System		
		DATE	**DURATION**	**na**
Gold	**DATE**	413	20	21
	DURATION	48	272	17

Table 29. Confusion matrix for MSRA using the RelIV-TIMEX Evaluation data set.

6.5.3 Evaluation Phase 3: End-2-End Performance Evaluation

The final phase of evaluation is to incorporate the best performing temporal disambiguation module into Chrono and evaluate the performance on the full set of returned annotations, i.e. End-2-End evaluation. For the End-2-End evaluation, the i2b2 evaluation scripts were used unmodified. As the work described in this chapter focused on temporal type classification, Chrono's performance from Chapter 5 for the span-based Precision, Recall, and F1 scores does not change. Instead, the goal is to see an improvement in the "Type Accuracy". With this in mind, the Value and Modifier metrics will change, however, optimizing these is future work as no changes were made to the normalization module in Chrono. Table 30 shows the final End-2-End results using the best performing temporal disambiguation module from the previous section, ClinBioBert-Seq2Seq-Ttype. Including the temporal disambiguation module into Chrono increased the Type Accuracy from 0.65 to 0.82. This large increase puts Chrono on par with the other state-of-the-art systems, however, it does not exceed them with MSRA still holding the highest Type Accuracy of 0.89.

One limitation of Chrono is that the FREQUENCY type parsing has not been fully implemented, and is limited to identifying known abbreviations for frequency expressions. This could be a factor in the poor performance of Chrono, thus, all systems were re-evaluated after removing the FREQUENCY temporal phrases from

System	P	R	F1	Type	Value	Modifier
Mayo	0.88	0.92	0.9	0.86	0.73	0.86
Vanderbilt	0.83	0.91	0.87	0.85	0.7	0.85
MSRA	0.88	0.95	0.91	0.89	0.72	0.89
Chrono w/o ordering (Chapter 5)	0.78	0.9	0.84	0.65	0.56	0.8
Chrono w/o ordering (ClinBioBert-Seq2Seq-Ttype)	0.78	0.9	0.84	0.82	0.57	0.77

Table 30. End-2-End results for state-of-the-art systems, Chrono, and Chrono+ClinBioBert-Seq2Seq-Ttype.

the results and gold standard using the same filtering script as mentioned previously. Table 31 shows that Chrono's Type Accuracy does indeed increase from 0.82 to 0.89 such that it is greater than the Mayo and Vanderbilt systems, but it is still second to MSRA at 0.91. This indicates that FREQUENCY phrases are a contributing factor; however, they are not the only factor as Chrono's Precision is reduced while the Recall is improved resulting in an unchanged F1 score of 0.84 while the F1 scores of all other systems were improved. Thus, while Chrono is now on par with state-of-the-art systems, it still has room for improvement.

System	P	R	F1	Type	Value	Modifier
Mayo	0.91	0.91	0.91	0.86	0.72	0.84
Vanderbilt	0.84	0.92	0.88	0.87	0.71	0.85
MSRA	0.89	0.96	0.93	0.91	0.71	0.90
Chrono w/o ordering (Chapter 5)	0.76	0.94	0.84	0.69	0.60	0.83
Chrono w/o ordering (ClinBioBert-Seq2Seq-Ttype)	0.75	0.94	0.84	0.89	0.62	0.80

Table 31. End-2-End results for state-of-the-art systems, Chrono, and Chrono+ClinBioBert-Seq2Seq-Ttype with FREQUENCY temporal phrases removed.

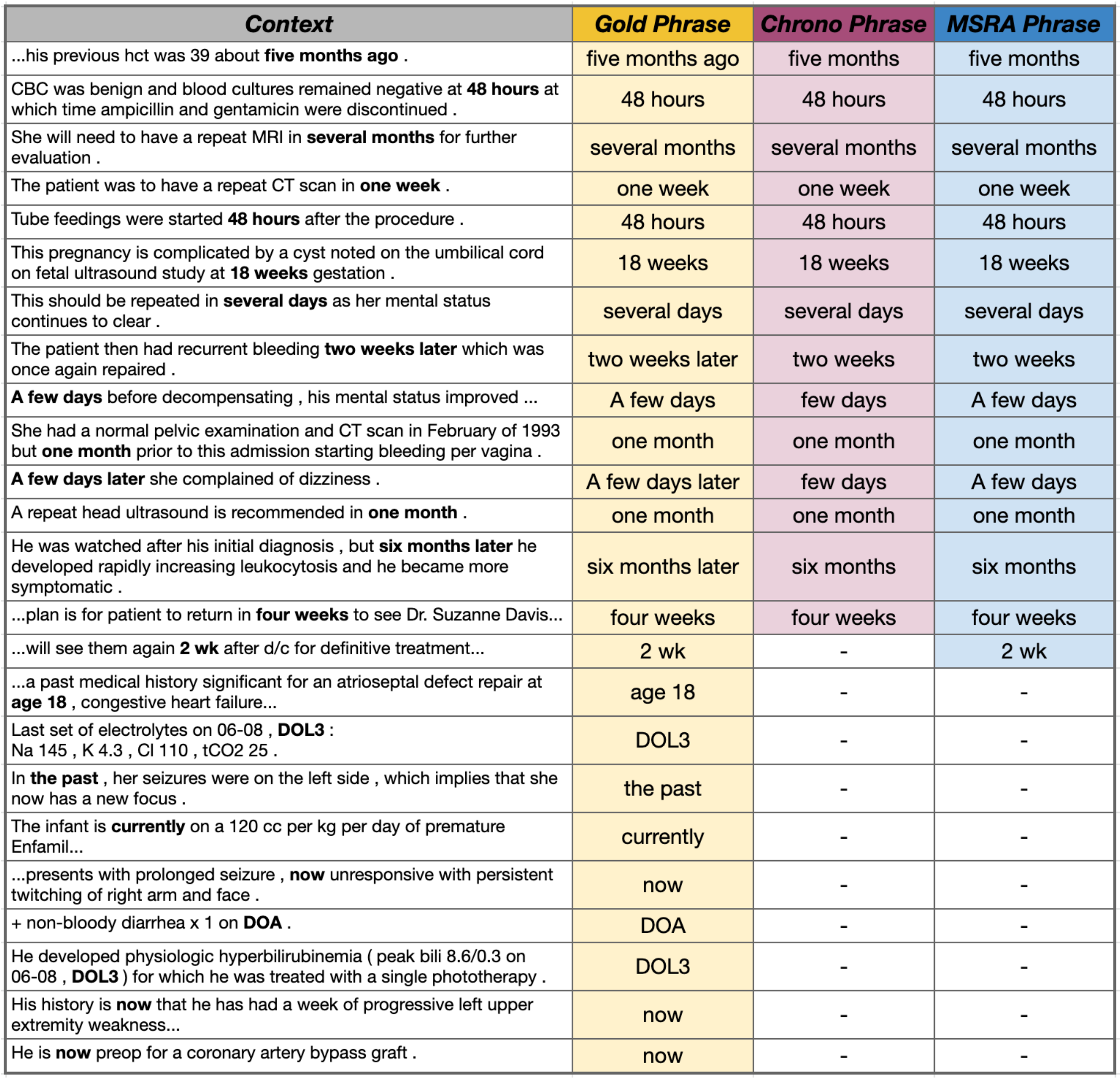
6.5.4 Error Analysis

This section provides a more detailed error analysis of Chrono+ClinBioBert-Seq2Seq-Ttype results when evaluated on the RelIV-TIMEX data set and the full End-2-End cohort to determine the next areas in need of improvement.

6.5.4.1 RelIV-TIMEX

For the RelIV-TIMEX data set, Chrono misclassified 48 phrases and failed to recognize a total of 68 phrases. In comparison, MSRA misclassified 68 phrases and failed to recognize 38. Of these, 49 phrases were either misclassified or missed by both Chrono and MSRA (24 DATEs and 25 DURATIONs). Table 32 lists the 24 DATEs that were misclassified or missed by either system. Both Chrono and MSRA identified, but misclassified, 14 phrases as DURATIONS when the gold standard lists them as DATEs. These include phrases like “five months ago”, “48 hours”, and “two weeks later”. The context around the phrases in all cases does clearly indicate these are discussing a distinct event, such as an MRI or when symptoms started. Chrono missed one phrase that MSRA identified (but also misclassified), which is the phrase “2 wk”. This was missed by Chrono as there are no rules to identify abbreviations such as “wk”. Finally, both systems failed to identify 9 DATE phrases. These include acronyms that are currently not parsed like, “DOL3” as well as vague terms that aid in ordering events such as “now”, “currently”, and “the past”, which were not consistently annotated as TIMEXs. Chrono does identify these terms, however, this function was turned off for this analysis as it introduced too many new phrases that are annotated by the SCATE schema, but are not annotated in the i2b2 data set (referred to as “Chrono w/o ordering” in Tables 30 and 31).

Similarly, there were 25 DURATION phrases either missed or misclassified by Chrono and MSRA (Table 33). Both systems identified, but misclassified, 10 DURATION phrases. These included phrases like “two days”, “the days”, and “all the night”. Chrono identified 3 phrases that MSRA missed (“6 - minute”, “this year”, “which time”), and MSRA identified 6 phrases that Chrono missed, most of which used abbreviations not recognized by Chrono, including “33 yrs ago”, “14d”, and

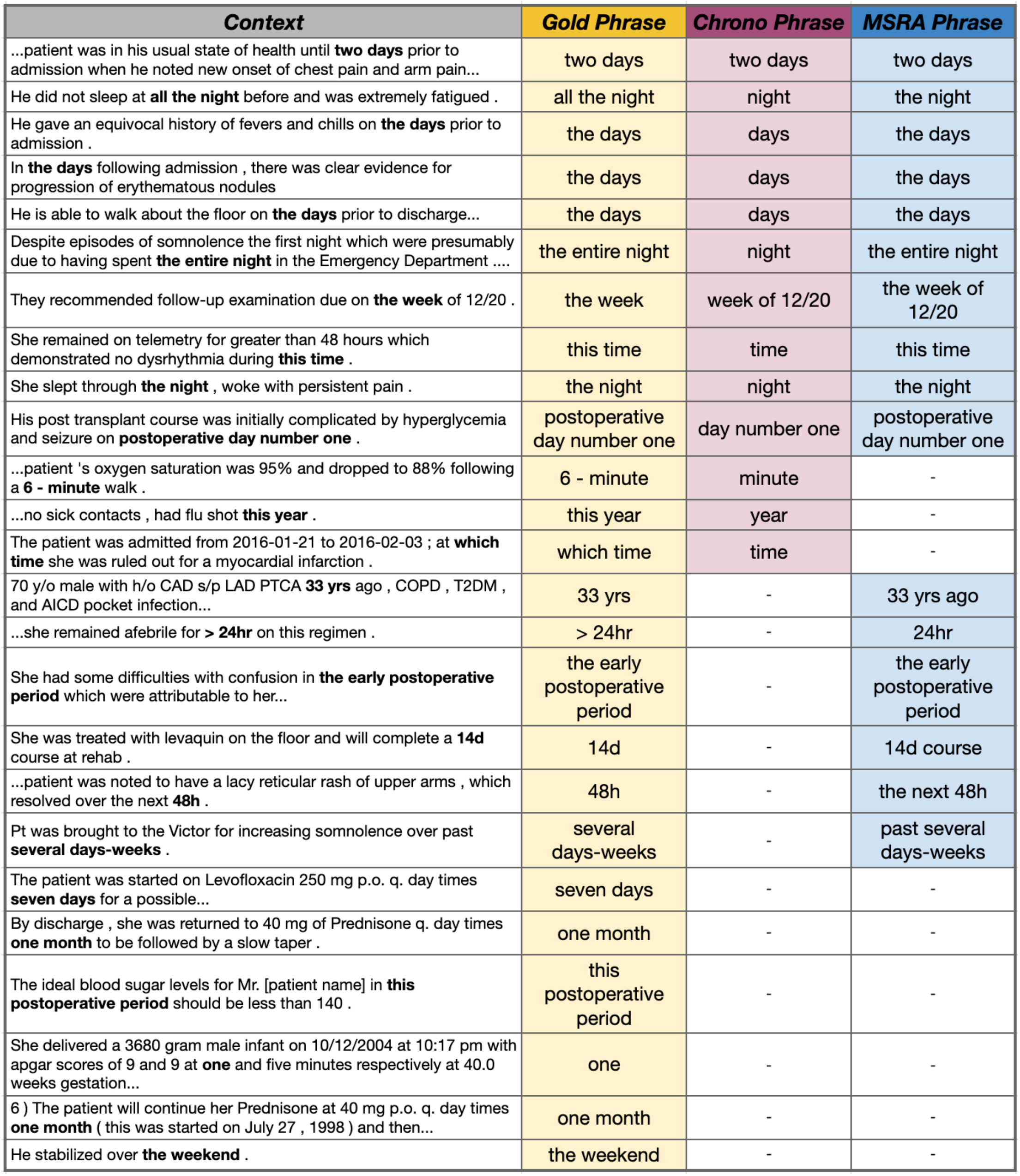


Context	*Gold Phrase*	*Chrono Phrase*	*MSRA Phrase*
...his previous hct was 39 about **five months ago** .	five months ago	five months	five months
CBC was benign and blood cultures remained negative at **48 hours** at which time ampicillin and gentamicin were discontinued .	48 hours	48 hours	48 hours
She will need to have a repeat MRI in **several months** for further evaluation .	several months	several months	several months
The patient was to have a repeat CT scan in **one week** .	one week	one week	one week
Tube feedings were started **48 hours** after the procedure .	48 hours	48 hours	48 hours
This pregnancy is complicated by a cyst noted on the umbilical cord on fetal ultrasound study at **18 weeks** gestation .	18 weeks	18 weeks	18 weeks
This should be repeated in **several days** as her mental status continues to clear .	several days	several days	several days
The patient then had recurrent bleeding **two weeks later** which was once again repaired .	two weeks later	two weeks	two weeks
A few days before decompensating , his mental status improved ...	A few days	few days	A few days
She had a normal pelvic examination and CT scan in February of 1993 but **one month** prior to this admission starting bleeding per vagina .	one month	one month	one month
A few days later she complained of dizziness .	A few days later	few days	A few days
A repeat head ultrasound is recommended in **one month** .	one month	one month	one month
He was watched after his initial diagnosis , but **six months later** he developed rapidly increasing leukocytosis and he became more symptomatic .	six months later	six months	six months
...plan is for patient to return in **four weeks** to see Dr. Suzanne Davis...	four weeks	four weeks	four weeks
...will see them again **2 wk** after d/c for definitive treatment...	2 wk	-	2 wk
...a past medical history significant for an atrioseptal defect repair at **age 18** , congestive heart failure...	age 18	-	-
Last set of electrolytes on 06-08 , **DOL3** : Na 145 , K 4.3 , Cl 110 , tCO2 25 .	DOL3	-	-
In **the past** , her seizures were on the left side , which implies that she now has a new focus .	the past	-	-
The infant is **currently** on a 120 cc per kg per day of premature Enfamil...	currently	-	-
...presents with prolonged seizure , **now** unresponsive with persistent twitching of right arm and face .	now	-	-
+ non-bloody diarrhea x 1 on **DOA** .	DOA	-	-
He developed physiologic hyperbilirubinemia (peak bili 8.6/0.3 on 06-08 , **DOL3**) for which he was treated with a single phototherapy .	DOL3	-	-
His history is **now** that he has had a week of progressive left upper extremity weakness...	now	-	-
He is **now** preop for a coronary artery bypass graft .	now	-	-

Table 32. DATE phrases missed or misclassified as DURATION by Chrono and MSRA.

“48h”. Finally, there were 6 DURATION phrases both systems failed to identify, including “seven days”, “one month”, and “one”. Looking at the context of the DURATION phrases, they are highly complex. For example, in the sentence “His post transplant course was initially complicated by hyperglycemia and seizure on postoperative day number one .” the gold standard annotated the phrase “postoperative day number one” as a DURATION, however, this could also be seen as a DATE as the complications happened on a single day. In fact most of the phrases that include the term “postoperative” are DATEs. Additionally, in the sentence “He did not sleep at all the night before and was extremely fatigued .” the gold standard annotated the phrase as “all the night”, which sounds like a duration, however, the other systems did not include the word “all” in the temporal phrase. In most instances the temporal term “night” is annotated as a DATE, whereas the phrase “overnight” is generally annotated as a DURATION. Finally, it is difficult to annotate the phrase “the week” in the sentence “They recommended follow-up examination due on the week of 12/20 .” because the full phrase actually does include a DATE. Also the semantics of the sentence really indicate a range of possible options for a follow-up examination, and are not indicating that the examination is going to occur over the whole time. This sentence provides an example of temporal semantics that are not really able to be annotated accurately by the TimeML schema, thus, one might view this annotation as subjective.

In addition to those phrases both systems misclassified or missed, Chrono misclassified 6 DURATIONs as DATEs and 15 DATEs as DURATIONs that MSRA got correct. Of the 15 DATE phrases that Chrono misclassified as DURATIONS, all of them had additional context words in the phrase that Chrono missed (Table 34) with the predominant term being “ago”. However, there were a few phrases where it could be argued that these are in fact DURATIONs. For example, in the sentence

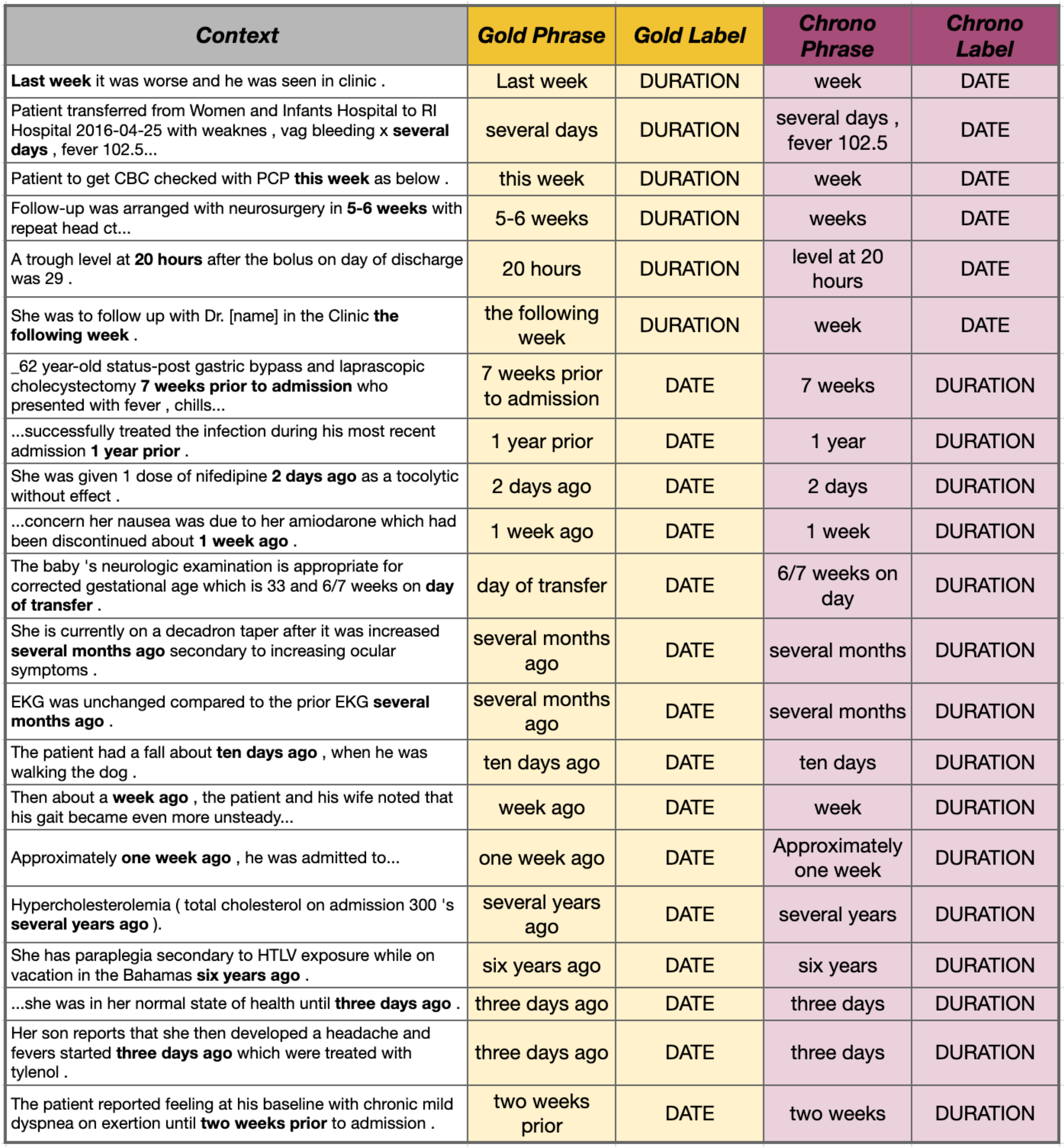


Context	Gold Phrase	Chrono Phrase	MSRA Phrase
...patient was in his usual state of health until **two days** prior to admission when he noted new onset of chest pain and arm pain...	two days	two days	two days
He did not sleep at **all the night** before and was extremely fatigued .	all the night	night	the night
He gave an equivocal history of fevers and chills on **the days** prior to admission .	the days	days	the days
In **the days** following admission , there was clear evidence for progression of erythematous nodules	the days	days	the days
He is able to walk about the floor on **the days** prior to discharge...	the days	days	the days
Despite episodes of somnolence the first night which were presumably due to having spent **the entire night** in the Emergency Department	the entire night	night	the entire night
They recommended follow-up examination due on **the week** of 12/20 .	the week	week of 12/20	the week of 12/20
She remained on telemetry for greater than 48 hours which demonstrated no dysrhythmia during **this time** .	this time	time	this time
She slept through **the night** , woke with persistent pain .	the night	night	the night
His post transplant course was initially complicated by hyperglycemia and seizure on **postoperative day number one** .	postoperative day number one	day number one	postoperative day number one
...patient 's oxygen saturation was 95% and dropped to 88% following a **6 - minute** walk .	6 - minute	minute	-
...no sick contacts , had flu shot **this year** .	this year	year	-
The patient was admitted from 2016-01-21 to 2016-02-03 ; at **which time** she was ruled out for a myocardial infarction .	which time	time	-
70 y/o male with h/o CAD s/p LAD PTCA **33 yrs** ago , COPD , T2DM , and AICD pocket infection...	33 yrs	-	33 yrs ago
...she remained afebrile for **> 24hr** on this regimen .	> 24hr	-	24hr
She had some difficulties with confusion in **the early postoperative period** which were attributable to her...	the early postoperative period	-	the early postoperative period
She was treated with levaquin on the floor and will complete a **14d** course at rehab .	14d	-	14d course
...patient was noted to have a lacy reticular rash of upper arms , which resolved over the next **48h** .	48h	-	the next 48h
Pt was brought to the Victor for increasing somnolence over past **several days-weeks** .	several days-weeks	-	past several days-weeks
The patient was started on Levofloxacin 250 mg p.o. q. day times **seven days** for a possible...	seven days	-	-
By discharge , she was returned to 40 mg of Prednisone q. day times **one month** to be followed by a slow taper .	one month	-	-
The ideal blood sugar levels for Mr. [patient name] in **this postoperative period** should be less than 140 .	this postoperative period	-	-
She delivered a 3680 gram male infant on 10/12/2004 at 10:17 pm with apgar scores of 9 and 9 at **one** and five minutes respectively at 40.0 weeks gestation...	one	-	-
6) The patient will continue her Prednisone at 40 mg p.o. q. day times **one month** (this was started on July 27 , 1998) and then...	one month	-	-
He stabilized over **the weekend** .	the weekend	-	-

Table 33. DURATION phrases missed or misclassified as DATE by Chrono and MSRA.

"...she was in her normal state of health until three days ago ." it could be argued that her un-normal state of health endured over three days, which is in fact a DURATION. Additionally, in the sentence "Her son reports that she then developed a headache and fevers started three days ago which were treated with tylenol ." the phrase "three days ago" refers to when the fevers started, which could be interpreted as they haven't ended and thus this phrase is in fact a DURATION. With respect to DATEs, the majority include the term "week" or "weeks" and Chrono again did not include some context words that were annotated in the gold standard. These are the precise types of relative terms that are difficult to pin to a timeline. Finally, there were 46 phrases that Chrono did not identify as a DATE or DURATION but MSRA got correct. The vast majority of these include acronyms that Chrono does not recognize, for example as "POD#6", "HD#3", and "14 d". It also included 4 phrases of "the second day". Chrono actually did recognize these phrases, but because they included the term "second" they were classified as a TIME type, which was counted as missing for the RelIV-TIMEX data set.

Overall, with the temporal disambiguation module, Chrono performs better for DURATION type than the state-of-the-art systems, which over-classify DATE types. For those phrases that both Chrono and MSRA classified incorrectly the reader has to utilize prior knowledge and the full context of the sentence to make a determination on if the phrase is a DATE or DURATION. This includes knowledge of symptoms and other medical events that are provided in the sentence context that rule-base and supervised learning systems may not be able to utilize effectively at this point. This is also true for many of the phrases that MSRA classified correctly but Chrono missed; however, it could be debated that the annotation could go either way for some phrases, which shows that is it also difficult for human annotators to identify temporal types in some instances. For phrases that Chrono missed completely, the



Context	*Gold Phrase*	*Gold Label*	*Chrono Phrase*	*Chrono Label*
Last week it was worse and he was seen in clinic .	Last week	DURATION	week	DATE
Patient transferred from Women and Infants Hospital to RI Hospital 2016-04-25 with weaknes , vag bleeding x **several days** , fever 102.5...	several days	DURATION	several days , fever 102.5	DATE
Patient to get CBC checked with PCP **this week** as below .	this week	DURATION	week	DATE
Follow-up was arranged with neurosurgery in **5-6 weeks** with repeat head ct...	5-6 weeks	DURATION	weeks	DATE
A trough level at **20 hours** after the bolus on day of discharge was 29 .	20 hours	DURATION	level at 20 hours	DATE
She was to follow up with Dr. [name] in the Clinic **the following week** .	the following week	DURATION	week	DATE
_62 year-old status-post gastric bypass and laprascopic cholecystectomy **7 weeks prior to admission** who presented with fever , chills...	7 weeks prior to admission	DATE	7 weeks	DURATION
...successfully treated the infection during his most recent admission **1 year prior** .	1 year prior	DATE	1 year	DURATION
She was given 1 dose of nifedipine **2 days ago** as a tocolytic without effect .	2 days ago	DATE	2 days	DURATION
...concern her nausea was due to her amiodarone which had been discontinued about **1 week ago** .	1 week ago	DATE	1 week	DURATION
The baby 's neurologic examination is appropriate for corrected gestational age which is 33 and 6/7 weeks on **day of transfer** .	day of transfer	DATE	6/7 weeks on day	DURATION
She is currently on a decadron taper after it was increased **several months ago** secondary to increasing ocular symptoms .	several months ago	DATE	several months	DURATION
EKG was unchanged compared to the prior EKG **several months ago** .	several months ago	DATE	several months	DURATION
The patient had a fall about **ten days ago** , when he was walking the dog .	ten days ago	DATE	ten days	DURATION
Then about a **week ago** , the patient and his wife noted that his gait became even more unsteady...	week ago	DATE	week	DURATION
Approximately **one week ago** , he was admitted to...	one week ago	DATE	Approximately one week	DURATION
Hypercholesterolemia (total cholesterol on admission 300 's **several years ago**).	several years ago	DATE	several years	DURATION
She has paraplegia secondary to HTLV exposure while on vacation in the Bahamas **six years ago** .	six years ago	DATE	six years	DURATION
...she was in her normal state of health until **three days ago** .	three days ago	DATE	three days	DURATION
Her son reports that she then developed a headache and fevers started **three days ago** which were treated with tylenol .	three days ago	DATE	three days	DURATION
The patient reported feeling at his baseline with chronic mild dyspnea on exertion until **two weeks prior** to admission .	two weeks prior	DATE	two weeks	DURATION

Table 34. Phrases missed by Chrono, but classified correctly by MSRA.

main issue is recognizing medical acronyms and short-hand, and there were a few instances where Chrono did identify the phrase, but classified it as a TIME type, which excluded it from this analysis.

6.5.4.2 End-2-End

In the End-2-End analysis, Chrono misclassified 727 temporal phrases with 421 of these being phrases Chrono identified that were not in the gold standard (Table 35). The vast majority of these included phrases stating a person's age, such as "71-year-old" or "23 year" that were not annotated in the i2b2 Gold Standard. In addition the word "time" is consistently annotated by Chrono as a DATE whereas the gold standard sometimes has it annotated and sometimes does not, a few vital measurements were incorrectly identified as a DATE by Chrono, and Chrono annotated several terms such as "fall", "early", and "prevent" incorrectly as a TIME type. Many of the other terms Chrono identified that were not in the gold standard may be gold standard annotation errors. For example, there are 5 instances of the word "daily" that are not annotated in the gold standard, and phrases such as "night of 2018-05-30" and "morning of 5/17" are not included in the gold standard, but when looked at in context are clearly DATE types. Additionally, some dates in the footer information of notes were not annotated in the gold standard, but were identified by Chrono.

Of the 306 remaining phrases annotated in the gold standard, Chrono completely misses 161 of these. As mentioned previously, the majority of DATE and DURATION types missed are those that use acronyms not identified by Chrono, such as "POD", "HD", and "14 d". A total of 85 of these are FREQUENCY types missed by Chrono. This is understandable because Chrono's frequency module is currently dictionary-based with a limited lexicon; thus, adding in a module to detect FREQUENCY

		Chrono				
		DATE	DURATION	FREQUENCY	TIME	na
Gold	DATE	1139	31	4	9	52
	DURATION	21	302	3	2	15
	FREQUENCY	16	37	67	0	85
	TIME	14	8	0	34	9
	na	316	38	8	59	-

Table 35. Chrono's confusion matrix for the End-2-End evaluation.

phrases like "x 2", "four cycles", and "q.4.h" is needed. TIME types that Chrono missed include difficult phrases like "one" as well as "8 o'clock", which is a pattern not currently recognized by Chrono.

Finally, 145 phrases were annotated in the gold standard and by Chrono, but Chrono got the temporal type incorrect (Table 35). There were 21 DURATION types misclassified as a DATE by Chrono, which included the issues discussed previously where Chrono is not picking up the entire phrase or one must utilize more of the context of the sentence to correctly classify the phrase. In addition, 16 FREQUENCY types were misclassified as a DATE. All but 2 of these included the phrase "per day" in the gold standard, but Chrono only annotated the word "day" and it was sent to the temporal disambiguation module. There were also 14 TIME phrases misclassified as a DATE by Chrono because Chrono failed to identify the full phrase with the time portion included. For example, for the phrase "June 18 , 2006 , at 8:30 p.m." Chrono only identified the first portion of "June 18 , 2006", which is a DATE. Chrono does include some logic to connect these two phrases, but this logic must be failing for these instances. There were 31 DATE types misclassified as DURATION for reason's indicated previously, and 37 FREQUENCY types misclassified as a DURATION due to the FREQUENCY module not being fully implemented and Chrono not identifying the full phrase. A total of 8 TIME types were misclassified as a DURATION, including phrases like "16 hours of life". These phrases could be debatable as an actual DURATION. There were 4 DATE and 3 DURATION types classified as FREQUENCY by Chrono as Chrono included frequency-related terms. For example, the gold standard phrase "08-16" is labeled a DATE, but Chrono identified the phrase "bid on 08-16" and labeled it as a FREQUENCY. Likewise, with the gold standard phrase "seven days" Chrono identified the full phrase of "q. day times seven days" which does actually seem to be a FREQUENCY type. Finally, there were 9 DATE and 2 DURATION phrases misclassified by Chrono as a TIME due to Chrono either not identifying the full phrase and missing some information, or the phrase had key words in it like "second" which is deterministically classified as a TIME by Chrono.

In conclusion, there are certainly several areas of improvement for Chrono, including fully implementing the FREQUENCY module and enabling better identification of medical acronyms like "POD". One of the larger and more complex issues Chrono has, despite the implementation of contextualized embeddings, is getting the classification of relative DATE and DURATION types correct for phrases that require additional contextual knowledge, such as type of event that is being referenced. In these instances adding in features that represent the referenced event, or analyzing the sentence structure may aid in the temporal type disambiguation task.

6.6 Conclusions and Contributions

Relative temporal expressions are difficult to normalize because their value always depends on another temporal expression or some event that is either implicit knowledge or information located in another part of the document. However, before they can be normalized to a value and be placed on a timeline, their temporal type must be determined. Chapter 5 identified that the disambiguation of relative

DATE and DURATION temporal types is still a challenge for state-of-the-art systems. This chapter addressed this challenge through implementing a second temporal disambiguation module in Chrono that utilizes contextualized embeddings from temporally fine-tuned BERT models. Through this work the following contributions were made with respect to the temporal disambiguation of relative DATE and DURATION types:

Negative Findings

1. Using BERT to perform temporal type classification/disambiguation directly performs poorly.
2. Chaining fine-tuning on a simple (binary) then complex (Seq2Seq) task degrades performance for both the Seq2Seq models and the embeddings used in the classical learning models.
3. Incorporating context and attention tokens directly into a feature vector degrades performance of the learning models.

Positive Findings

4. Incorporating the contextualized word embeddings into classical learning models reaches state-of-the-art performance for the DATE/DURATION temporal disambiguation task.
5. Temporally fine-tuning BERT models on complex tasks create contextualized word embeddings that increase the performance of classical learning models on the DATE/DURATION temporal disambiguation task.
6. While adding context generally degrades performance, this feature extraction strategy can help unmodified BertBase embeddings compensate for domain shifts.

Research Products

7. Two focused Training and Evaluation data sets developed from the 2012 i2b2

Temporal Challenge formatted for 3 tasks[‖]: ISO-TimeML XML format, temporal sentence classification, and Seq2Seq.

8. A Python script that can take the ISO-TimeML results from any other system that parsed the 2012 i2b2 data set and filter it to those elements in the RelIV-TIMEX evaluation data set, or any other filtered subset of the gold standard. Available in the "gold-standard-utils" repository on the OlexLab GitHub page[**].
9. Six temporally fine-tuned BertBase and ClinBioBert models available in the "temporal-bert" repository on the OlexLab GitHub page along with associated fine-tuning code.
10. A Python object-oriented framework for extracting and summarizing contextualized embeddings for temporal phrases, including context and attention tokens. To be made available in the "summarize-bert-embeddings" repository on the OlexLab GitHub page upon publication.
11. A novel algorithm for summarizing BERT attention weight matrices to identify to which tokens an entire temporal phrase is paying the most attention. To be made available in the "summarize-bert-embeddings" repository on the OlexLab GitHub page upon publication.
12. Chrono, the first TERN system to normalize temporal expressions to both the SCATE and ISO-TimeML annotation schemes and implement a temporal disambiguation module that utilizes temporally fine-tuned contextualized embeddings. Chrono is now the state-of-the-art for disambiguating relative DATE/DURATION temporal phrases. Available on GitHub[††].

[‖] Available upon request and after being approved for access to the i2b2 corpus.

[**] https://github.com/OlexLab/

[††] https://github.com/AmyOlex/Chrono

In conclusion, this work has made progress in the area of temporal recognition and normalization by 1) showing that temporal information can be infused in contextualized embeddings extracted from BERT models, 2) improving the ability of systems to disambiguate DATE and DURATION relative temporal phrases, and 3) providing the first dual-parsing TERN system, Chrono, that normalizes temporal expressions into both the SCATE and ISO-TimeML schemes. Future work will include improving the classification of relative temporal expressions that require a deeper understanding of semantics, such as the type of event.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

The long-term goal of this research is to build a system that can reconstruct and visualize a patient's clinical timeline (symptoms, diagnoses, treatments, tests, etc) from unstructured and structured EHR data. Currently, there are only a few systems that can process unstructured clinical notes to extract a patient's medical timeline; however, their performance is poor according to current evaluation metrics or their results are not evaluated quantitatively. It is challenging to get good performance on a high-level task like timeline extraction because it depends on the output of many lower-level tasks. If the lower level tasks contain errors in output, then a cascade of errors will result and be present in the extracted timelines.

This work has performed a survey of the progress already made for each of the components of timeline extraction, identifying areas of future work in each. This dissertation focused on temporal expression recognition and normalization (TERN), specifically the temporal type disambiguation task, which is a fundamental component of timeline extraction. This is a tangential direction to the current focus of the temporal reasoning literature in the clinical and general domain, which are focused on temporal relation extraction and direct temporal expressions, leaving much needed work in the area of relative temporal expression normalization (see Chapter 2).

Recent work on TERN in clinical NLP has shown rule-base or hybrid methods perform the best for this task (Chapter 2). However, a detailed error analysis of the top i2b2 systems participating in the 2012 i2b2 Temporal Challenge (Chapter 5) revealed that, while the algorithms for the recognition of temporal expressions perform well, the normalization of these expressions was lacking. Specifically, the normalization of relative, vague, and implicit temporal expressions to an explicit date, time, frequency, or duration. This normalization is important for correctly placing events on a timeline. While many temporal expressions are explicit dates or times, clinical notes often contain relative expressions, such as “two weeks ago”, that also need to be normalized to a specific date. The performance of top systems for the i2b2 corpus indicate that getting the correct normalized value for relative temporal expressions is challenging.

This dissertation work detailed the development of Chrono, the first TERN system capable of normalizing temporal expressions into the SCATE and ISO-TimeML schemes (Chapters 3, 4, and 5), and the first temporal type disambiguation module to utilize contextualized embeddings extracted from temporally fine-tuned BERT models (Chapter 6). This work concludes that fine-tuning BERT models utilizing a single complex task (Figure 14E) generated contextualized embeddings that are more applicable to temporal type disambiguation than using a simpler temporal task (Figure 14B), or chaining fine-tuning tasks (Figure 14C and D). In addition, this work found that extracting contextualized embeddings from a fine-tuned BERT model for temporal type classification using classical learning models such as SVMs and CNNs outperform fine-tuning a BERT model and performing temporal type classification directly. This work also found that adding in context and attention tokens degrades the performance of SVM and CNN classifiers for the temporal type disambiguation task. After implementing the best performing temporal type disambiguation module, Chrono is now the state-of-the-art in disambiguating DATE and DURATION temporal types from relative temporal expressions, however there is still much work to be done. The following sections lay out a few areas of what should be focused on next.

7.1 Improving Chrono's Dictionary and the Gold Standard

From the End-2-End analysis it is clear Chrono needs a stronger FREQUENCY normalization module, and better recognition of clinical acronyms and abbreviations. These elements need to be parsed into both the SCATE and ISO-TimeML schemes; however, Chapter 2 identified that many SCATE errors are due to gold standard issues. In order for Chrono to improve it's performance the gold standard for the SCATE schema will need to be 1) error checked and 2) expanded to include more documents. Chrono could be used as a silver standard to jump start this effort. Additionally, Chrono may benefit from the integration of external knowledge bases for recognizing clinical acronyms and abbreviations, which has previously been show to be useful in detecting clinical abbreviations in admission notes [150].

7.2 Support for Different Clinical Document Types

This work utilized temporally annotated clinical notes from the i2b2 and THYME temporal challenge corpora, which are limited in both note type and domain. The i2b2 corpus only consists of discharge summaries, which generally contain multiple sections that are temporally dense such as patient history and clinical course [21]. The THYME corpus is limited to brain and colon cancer patient notes and pathology reports [12]. However, there are many other note types written for various purposes and audiences [151] (i.e. other care providers, billing, etc.) that contain important temporal information about a patients medical timeline. The HL7 FHIR [152] US Core DocumentReference Type documentation* specifically lists 1,001 different types of clinical notes that each are assigned their own LOINC (Logical Observation Identifiers Names and Codes) code [153]. The top 5 types are know as "Common Clinical

* http://hl7.org/fhir/us/core/2019Jan/ValueSet-us-core-documentreference-type.html

Notes" and include Consultation Note, Discharge Summary, History & Physical Note, Procedure Note, and Progress Note. Other HL7 FHIR supported note types include Diagnostic Reports (Cardiology, Pathology, Radiology), Referral Note, Surgical Operation Note and Nurse Note[†]. While temporal concepts are relatively domain agnostic, the ways in which these concepts are expressed can differ across note types and domains, which this work demonstrated when Chrono was adapted to the i2b2 corpus (discharge summaries) from the THYME corpus (clinical notes and pathology reports for cancer patients) in Chapter 5. Thus, it will be important to train and test Chrono on multiple clinical document types so that it can accurately extract temporal information from a patients entire clinical record instead of just specific document types, which may not capture all pertinent medical events.

7.3 Incorporate Temporally Fine-Tuned Contextualized Embeddings into the SCATE Temporal Disambiguation Module.

This work has demonstrated that temporally fine-tuned contextualized embeddings can be used to perform temporal type disambiguation and reach state-of-the-art performance. The DATE/DURATION Temporal Type Disambiguation module is the second TTD module used by Chrono. The first utilizes hand-crafted feature vectors that include context clues to disambiguate Period and Calendar-Interval SCATE entities (Chapter 3). Work done by Xu et al. [146] (the only other known work to utilize contextualized embeddings for the temporal type classification task) utilizes character-level pre-trained contextualized embeddings from Flair [154] to classify SCATE temporal types, and found that contextualized embeddings are more robust to term variability and remove the need to utilize features such as part of speech and

[†] http://hl7.org/fhir/us/core/2019Jan/clinical-notes-guidance.html

capitalization. Thus, future work on Chrono could include the training and utilization of a classifier that uses temporally fine-tuned contextualized embeddings for the Period and Calendar-Interval disambiguation task.

7.4 Incorporating EVENTs in Temporal Disambiguation

Temporally fine-tuned contextualized embeddings have made it easier to distinguish between DATE and DURATION types, however, they still have difficulty classifying some relative temporal expressions that require a deeper understanding of semantics, such as the type of event. For example, in the sentence "...patient was in his usual state of health until **two days** prior to admission when he noted new onset of chest pain and arm pain..." the temporal phrase "two days" is annotated in gold as a DURATION type, but Chrono annotated it as a DATE. To disambiguate this properly, one must understand the context of "usual state of health" and "new onset of chest pain and arm pain". This context appears both before and after the temporal phrase and implies that the patient was not in his "usual state of health" *starting* "two days prior to admission" and not ending as the patient was ultimately admitted to the hospital. Incorporating information about EVENTs surrounding the temporal expression may help in the disambiguation task; however, this requires EVENT parsing and EVENT-TIMEX relations to be identified. Much attention has been paid to this area in recent years, thus, a next step for Chrono is to incorporate state-of-the-art modules to perform these tasks and then utilize this information to perform temporal disambiguation of relative expressions.

7.5 Identifying Anchor Times for Relative Temporal Expressions

Relative temporal expressions that are classified as a DATE type require additional processing to normalize them to the correct calendar day, month, and/or year.

This is a complex problem because the anchor time could be one of many different dates from a clinical note. Frequently, the anchor time is the admission date, and verb tense can be a strong indicator of whether the time is before, during, or after the admission date. However, some notes are written over many days, so the verb tense cannot always be relied upon. For example, within a single note one passage may refer to the admission date with the token "today", while another passage later on may be referring to the discharge date using the same token. Another indicator of if the anchor time for an expression is the admission date, or some other date, is the section type. For example, a section on past medical history is probably referring to all events prior to admission, whereas text in the discharge summary are referring to events either at discharge, during the encounter, or after discharge. However, this is still not a stead fast rule for all cases because some text can be discussing events that happened relative to past events. While there is much complexity here, a human reading these clinical notes can easily deduce the anchor time of a relative expression. This is due to humans being able to identify different sentences or sections of text being members of different *temporal contexts* where each temporal context has a defined anchor time.

If a segment of text is divided into temporal contexts and an anchor time is assigned to each context, then it should be possible to assign the correct value of relative temporal expressions within each segment. Segmenting text based on temporal context is not necessarily new, but has historically been focused on ordering events and not normalizing temporal expressions. Bramsen et al. [155] used a machine learning approach to divide discharge summaries into temporal segments, and then identify high-level temporal relationships between each pair (before, after, etc) to induce event order. However, Bramsen et al. utilized discharge summaries that had been re-written into a narrative style instead of using the raw physician-generated text, and was focused on event ordering rather than identifying specific anchor times for the temporal segments. More recently, Raghavan [116] takes a similar high-level binning strategy, however, only uses the annotated events rather than breaking the text into temporal segments.

As future work, we can take this binning strategy one step further to not only place relative temporal expressions into high-level bins, but to also assign anchor times to temporal context segments, which can then be used to normalize the relative temporal expressions. Identifying temporal segments will require a layered approach. First, direct and relative temporal expressions will first need to be identified and classified. Next, these annotated temporal expressions will be used as boundaries for the temporal segments. Third, verb tense, section type, admission/discharge time, and existing direct temporal expressions will be used to identify the anchor time for each segment. Finally, the segments with relative temporal expressions will be further processed to calculate the explicit date referred to by each relative temporal expression. This could be implemented through using a rule-based approach as well as a hybrid approach that incorporates machine learning into the anchor time selection for each segment. The 2012 i2b2 Temporal Challenge development cohort can be used to build out these modules.

7.6 Consider Ensemble Classifiers

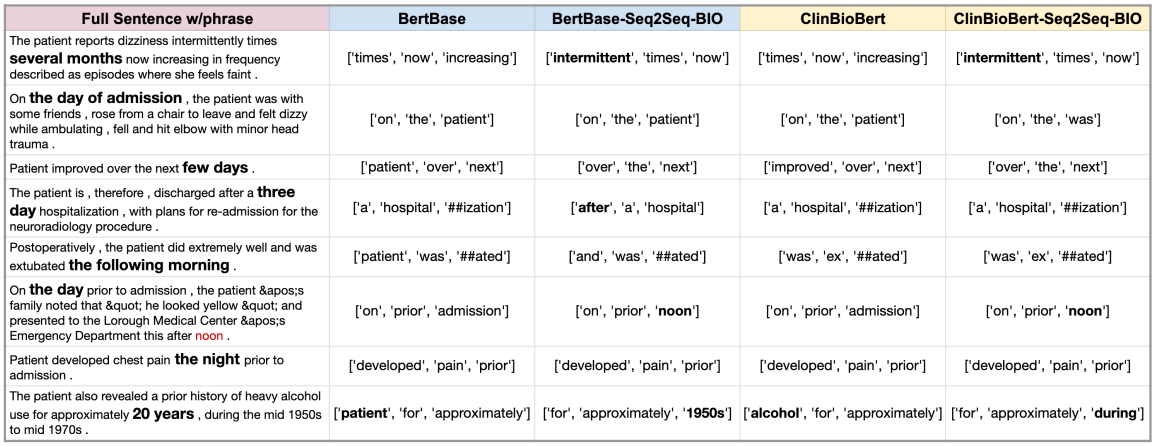
The performance for many of the strategies explored in this work were very close, which made it difficult to choose the best model to move forward. In addition, moving from the first to the second phase of evaluations resulted in the ClinBioBert-Seq2Seq-Ttype strategy outperforming the BIO strategy, which was the best performer in the first evaluation phase. This makes it difficult to choose a single best classifier, as they all perform well and may bring different strengths to the table. Thus, future work should include the exploration of utilizing these strategies in an ensemble classifier to avoid having to pick a single best model.

7.7 Do we need Attention?

This work introduced a novel approach to identifying which tokens an *entire temporal phrase* is attending to by summarizing the attention weight matrices output by BERT. While adding in attention tokens did not benefit this work, the algorithm presented does summarize the attention weight matrices to identify the top N words that are attended to by the entire temporal phrase. Browsing through these reveals that some are focused on key context works like "prior" and others are focused on medical events. In addition, comparing the top 3 attention tokens extracted for the same phrase before and after fine-tuning revealed that temporal fine-tuning results is tokens that are more temporally focused. For example, in Table 36 the first example with the phrase "several months" attends to the tokens 'times', 'now', and 'increasing' for both the baseline BertBase and ClinBioBert models, but the 'increasing' token is replaced by 'intermittent' in the temporally fine-tuned models. This method also has the potential to retrieve long-distance relationships as shown in the sixth example where the attention token 'admission' is replaced by the temporal token 'noon'. While this is an incorrect relation, it does demonstrate the potential for retrieving these types of long distance relationships. Future work will involve exploring whether this algorithm can contribute towards tasks like EVENT-TIMEX relation linking or anchor time identification.

7.8 Modifying/Adding Attention Heads to Focus on Temporal Features

The attention mechanism is key to obtaining contextualized word embeddings. Within the BERT models utilized in this work there are 12 layers, each with 12 atten-



Full Sentence w/phrase	BertBase	BertBase-Seq2Seq-BIO	ClinBioBert	ClinBioBert-Seq2Seq-BIO
The patient reports dizziness intermittently times **several months** now increasing in frequency described as episodes where she feels faint .	['times', 'now', 'increasing']	['**intermittent**', 'times', 'now']	['times', 'now', 'increasing']	['**intermittent**', 'times', 'now']
On **the day of admission** , the patient was with some friends , rose from a chair to leave and felt dizzy while ambulating , fell and hit elbow with minor head trauma .	['on', 'the', 'patient']	['on', 'the', 'patient']	['on', 'the', 'patient']	['on', 'the', 'was']
Patient improved over the next **few days** .	['patient', 'over', 'next']	['over', 'the', 'next']	['improved', 'over', 'next']	['over', 'the', 'next']
The patient is , therefore , discharged after a **three day** hospitalization , with plans for re-admission for the neuroradiology procedure .	['a', 'hospital', '##ization']	['**after**', 'a', 'hospital']	['a', 'hospital', '##ization']	['a', 'hospital', '##ization']
Postoperatively , the patient did extremely well and was extubated **the following morning** .	['patient', 'was', '##ated']	['and', 'was', '##ated']	['was', 'ex', '##ated']	['was', 'ex', '##ated']
On **the day** prior to admission , the patient &apos;s family noted that &quot; he looked yellow &quot; and presented to the Lorough Medical Center &apos;s Emergency Department this after noon .	['on', 'prior', 'admission']	['on', 'prior', '**noon**']	['on', 'prior', 'admission']	['on', 'prior', '**noon**']
Patient developed chest pain **the night** prior to admission .	['developed', 'pain', 'prior']	['developed', 'pain', 'prior']	['developed', 'pain', 'prior']	['developed', 'pain', 'prior']
The patient also revealed a prior history of heavy alcohol use for approximately **20 years** , during the mid 1950s to mid 1970s .	['**patient**', 'for', 'approximately']	['for', 'approximately', '**1950s**']	['**alcohol**', 'for', 'approximately']	['for', 'approximately', '**during**']

Table 36. Examples of how temporal fine-tuning alters a temporal phrase's attention.

tion heads. Each layer and head utilizes projection methods that focus on different aspects of the input sentence to create the final contextualized embedding. Altering or implementing new attention heads that are designed to specifically focus on temporal information and/or related events could provide contextualized embeddings that further improve the performance of temporal reasoning tasks.

7.9 Final Notes on Future Work for Timeline Extraction

During the course of this work, we reviewed the current state of temporal reasoning with respect to timeline extraction in the clinical domain. While much progress has been made, the current state-of-the-art still has a ways to go before practical application in the clinical setting will be possible. This work has identified several areas of research that are necessary to make this possible. First, the correct and complete identification of temporal expressions is fundamental to determining when events happened and for placement on a timeline. Temporal expression taggers must be able to identify all types of temporal expressions, including relative, vague, and implicit expressions. In addition, systems need to be able to normalize these expressions to a point on the timeline. This is difficult with relative, vague and implicit TIMEXs, and will require some sort of integrated approach that includes event identification, co-reference resolution, and temporal relation classification in order to determine where an event occurred in a patient's history. A second area of needed work is in developing temporal relation identification systems that perform well on inter-sentence relations and can control for event ordering conflicts. This may require a paradigm shift from looking at pair-wise relations to another framework, such as ranked lists, to control for this issue and reduce computational complexity. Third, constructing a patient's timeline over their entire medical history will require the processing of multiple types of documents, which will have duplicated information. Clinical temporal reasoning systems will need to be able to process this redundant data, which will include cross-document co-reference resolution, in order to limit displaying duplicated events to a clinician. Current work in this area is limited and has room for much progress. Fourth, visualization tools will need to be interoperatable and be able to integrate multiple data types for ease of use by clinicians. Another area of improvement is the development of consistent evaluation methods for timeline extraction systems as a whole so that they can be more easily compared and evaluated. Finally, timeline extraction systems should be able to integrate both structured and unstructured data into the timeline creation process. There are many tools that just use structured data, however, there is much information hidden in the unstructured texts. Being able to integrate this data will be of great benefit to future timeline extraction systems for clinical data.

In conclusion, temporal reasoning over clinical texts has come a long way since the first clinical temporal challenges, however, there is still room for improvement before these systems will be useful to clinicians. Because a patient's medical history is buried in multiple notes with multiple note types and grammar that is not always going to follow traditional rules, future timeline extraction systems should be flexible in processing this diverse data, as well as able to deal with the high level of redundancy in the EHR by integrating this data into a single contiguous timeline through robust co-reference resolution. As the field moves towards annotating the more difficult temporal information, such as relative and implicit temporal expressions, new methods that integrate the normalization of temporal expressions with temporal relation identification and co-reference resolution will be needed.

CHAPTER 8

SUMMARY OF CONTRIBUTIONS

In this chapter the contributions this dissertation work has made to the field of Clinical Natural Language Processing and Temporal Reasoning are summarized by chapter.

8.1 Chapter 2

Portions of Chapter 2 provide a comprehensive review of the current state of temporal reasoning in the clinical domain, and highlights several areas in need of attention as future work for the field. This was published in the Journal of Biomedical Informatics [8].

- Olex AL, McInnes BT. Review of Temporal Reasoning in the Clinical Domain for Timeline Extraction: Where we are and where we need to be. Journal of Biomedical Informatics 2021;118:103784.

8.2 Chapter 3

Chapter 3 described the first hybrid framework for normalizing fine-grained temporal information into the SCATE scheme, which is implemented in the tool Chrono and available on GitHub*. This chapter was presented as an oral presentation and poster at the 2018 SemEval Workshop, and published as a full-length, peer-reviewed paper [42].

- Olex A, Maffey L, Morgan N et al. Chrono at SemEval-2018 Task 6: A System for Normalizing Temporal Expressions. Proceedings of The 12th International

*https://github.com/AmyOlex/Chrono

Workshop on Semantic Evaluation. New Orleans, Louisiana: Association for Computational Linguistics, 2018, 97–101.

- Olex A, Maffey L, Morton N et al. Chrono: A System for Normalizing Temporal Expressions. The 12th International Workshop on Semantic Evaluation 2018. Poster and Oral presentation by Amy Olex.

8.3 Chapter 4

Chapter 4 demonstrated that clinical domain texts pose additional challenges to TERN systems, and identified 6 aspects of temporal parsing one should consider when migrating a system from the general to clinical domain. This chapter was presented as an oral presentation at the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HTL), and published as a full-length, peer-reviewed paper in the proceedings [45].

- Olex A, Maffey L, McInnes B. NLP Whack-A-Mole: Challenges in Cross-Domain Temporal Expression Extraction. 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). Minneapolis, Minnesota: Association for Computational Linguistics, 2019, 3682–92. Full-length manuscript and Oral Presentation by Amy Olex.

8.4 Chapter 5

Contributions of Chapter 5 include 1) implementation of 3 strategies to convert SCATE annotations to ISO-TimeML, 2) development of the first system to parse temporal phrases into both the SCATE and ISO-TimeML schemes, and 3) identification of 6 types of errors state-of-the-art systems make when processing the 2012 i2b2 data set, which sets the stage for future work in this area. This chapter is currently in preparation for submission as a journal article.

8.5 Chapter 6

Chapter 6 includes many contributions that span positive and negative findings regarding the use of temporally fine-tuned contextualized embeddings for temporal type disambiguation, and several research products that can be utilized by others. Of particular note is that *this work is the first to temporally fine-tune BERT models and then use the subsequent temporally fine-tuned contextualized embeddings for the TTD task.* Only two other works have utilized contextualized embeddings in temporal reasoning tasks [146, 71], but neither of them fine-tuned the language models to a temporal task, and this is the first work to utilize these embeddings specifically for the TTD task in the ISO-TimeML scheme.

Negative Findings

1. Using BERT to perform temporal type classification/disambiguation directly performs poorly.
2. Chaining fine-tuning on a simple (binary) then complex (Seq2Seq) task degrades performance for both the Seq2Seq models and the embeddings used in the classical learning models.
3. Incorporating context and attention tokens directly into a feature vector degrades performance of the learning models.

Positive Findings

4. Incorporating the contextualized word embeddings into classical learning models reaches state-of-the-art performance for the DATE/DURATION temporal disambiguation task.
5. Temporally fine-tuning BERT models on complex tasks create contextualized

word embeddings that increase the performance of classical learning models on the DATE/DURATION temporal disambiguation task.

6. While adding context generally degrades performance, this feature extraction strategy can help unmodified BertBase embeddings compensate for domain shifts.

Research Products

7. Two focused Training and Evaluation data sets developed from the 2012 i2b2 Temporal Challenge formatted for 3 tasks[†]: ISO-TimeML XML format, temporal sentence classification, and Seq2Seq.
8. A Python script that can take the ISO-TimeML results from any other system that parsed the 2012 i2b2 data set and filter it to those elements in the RelIV-TIMEX evaluation data set, or any other filtered subset of the gold standard. Available in the "gold-standard-utils" repository on the OlexLab GitHub page[‡].
9. Six temporally fine-tuned BertBase and ClinBioBert models available in the "temporal-bert" repository on the OlexLab GitHub page along with associated fine-tuning code.
10. A Python object-oriented framework for extracting and summarizing contextualized embeddings for temporal phrases, including context and attention tokens. To be made available in the "summarize-bert-embeddings" repository on the OlexLab GitHub page upon publication.
11. A novel algorithm for summarizing BERT attention weight matrices to identify to which tokens an entire temporal phrase is paying the most attention. To be made available in the "summarize-bert-embeddings" repository on the OlexLab GitHub page upon publication.

[†] Available upon request and after being approved for access to the i2b2 corpus.

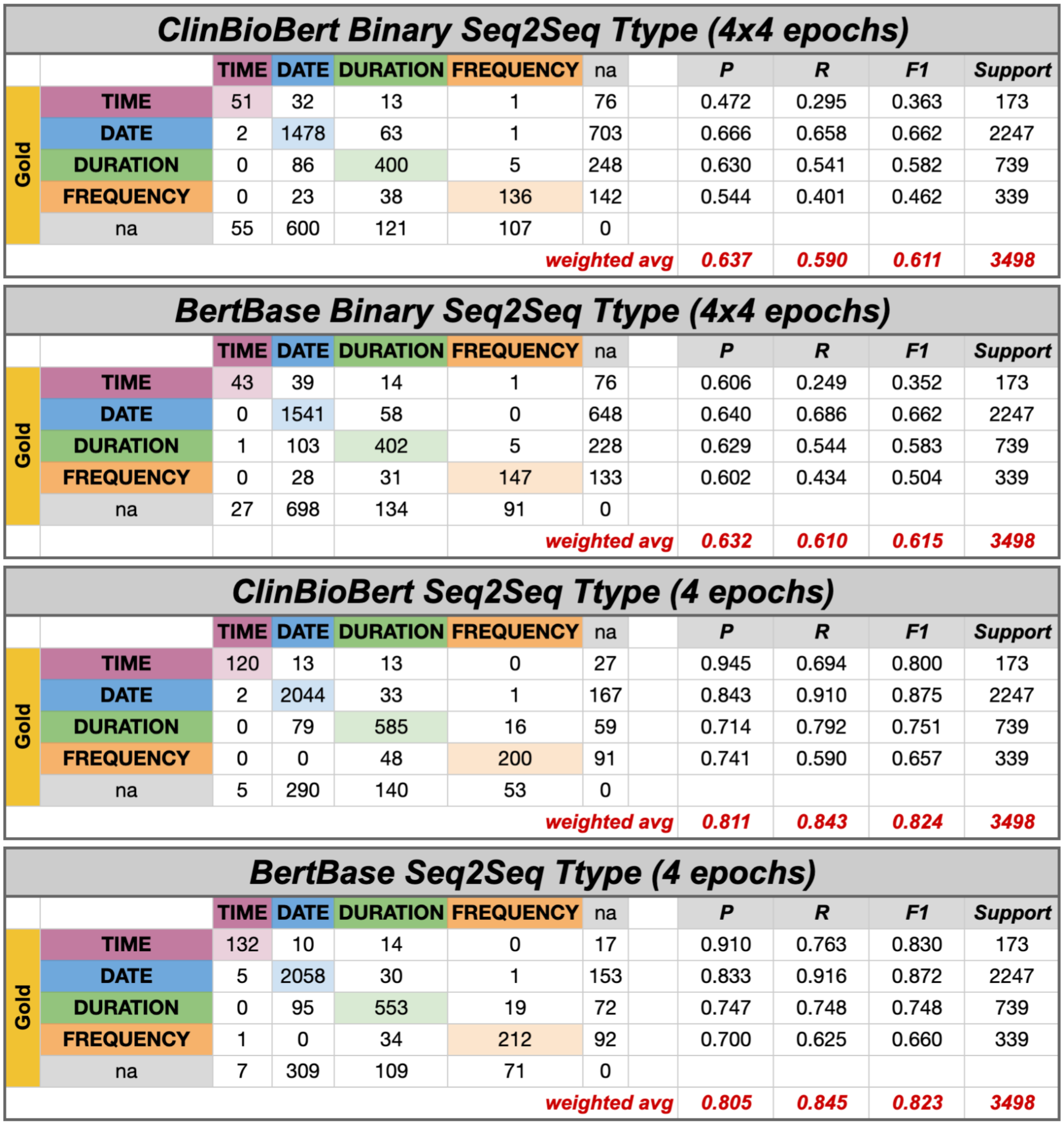
[‡] https://github.com/OlexLab/

12. Chrono, the first TERN system to normalize temporal expressions to both the SCATE and ISO-TimeML annotation schemes and implement a temporal disambiguation module that utilizes temporally fine-tuned contextualized embeddings. Chrono is now the state-of-the-art for disambiguating relative DATE/DURATION temporal phrases. Available on GitHub[§].

[§] https://github.com/AmyOlex/Chrono

ID	File	Phrase	Gold Value	Mayo Value	Vanderbilt Value	MSRA Value
1	32	Yesterday morning , he developed...	5/12/2006	11/16/2006	5/13/2006	2003
2	32	On physical exam today	5/16/2006	11/16/2006	5/13/2006	5/13/2006
3	32	Prior to discharge today	5/16/2006	6/18/2006	6/18/2006	5/16/2006
4	73	Cholangiogram on postoperative day number two showed...	8/26/2009	8/19/2009	8/26/2009	8/24/2009
5	73	Cholangiogram on postoperative day number two...At the time...	8/26/2009	-	8/26/2009	8/24/2009
	73	Cholangiogram on postoperative day number two...At the time...At the same time	8/26/2009	8/19/2009	-	8/24/2009
	73	On postoperative day number eight...	9/1/2009	8/25/2009	9/1/2009	8/24/2009
	73	On postoperative day number eight...Chest x-ray and sputum culture obtained at the time...	9/1/2009	-	9/1/2009	-
	73	On postoperative day number ten...	9/3/2009	8/27/2009	9/3/2009	8/24/2009
6	73	On postoperative day number 17...	9/10/2009	9/3/2009	9/10/2009	9/10/2009
7	73	On postoperative day number 17...At the time...	9/10/2009	-	9/10/2009	9/10/2009
	137	...during his most recent admission 1 year prior .	10/10/2014	-	-	10/10/2015
	137	...status-post gastric bypass ...7 weeks prior to admission who presented...	8/17/2015	-	-	2/7/2015
	208	...which had been discontinued about 1 week ago .	5/19/2018	5/1/2018	5/19/2018	4/17/2018
	233	...started earlier in the day...	6/4/2015	-	6/8/2015	6/4/2015
	233	...pain was intermittent through the day...	6/4/2015	-	6/8/2015	6/4/2015
	253	...HSV outbreak occurred on 2017-09-13 approximately one week prior to delivery .	9/15/2017	-	-	9/8/2017
	253	...serum bilirubin obtained on day of life three...	9/24/2017	9/24/2017	9/24/2017	9/22/2017
11	253	Antibiotics were discontinued on day of life three...	9/24/2017	9/24/2017	9/24/2017	9/22/2017
12	402	...required a dilt gtt on the day prior to call-out...	2/17/2013	-	2/21/2013	2/18/2013
	402	...was transitioned to PO diltiazem and has been in NSR since this time .	2/18/2013	-	2/21/2013	2/18/2013
13	402	...transitioned to PO diltiazem on the day of call-out .	2/18/2013	-	2/21/2013	2/21/2013
14	402	...was followed by urology during her stay and will see them again 2 wk after d/c...At this time , urology will coordinate removal of...	3/13/2013	2/21/2013	2/21/2013	2/21/2013
	527	...and was discharged to rehab on day 34/42 of the vancomycin .	2/19/2017	3/9/2017	-	-
	527	...the plan was for steroid taper : 60 mg x 10 days (already completed) , 40 mg x 14 d (already completed) , 20 mg x 14 d (now day 11-24) , 10 mg x 10 d , 5 mg x 10 d .	3/4/2017	2/14/2017	11/24/2017	2/14/2017
	537	...daughter says that on the day PTA...	1/19/2014	-	2/3/2014	1/20/2014
15	737	...underwent cardiac catheterization today...	6/10/2015	5/4/2015	5/4/2015	9/2/2015
	737	He is now preop for...	6/10/2015	-	9/2/2015	-
16	767	...until one and a half weeks prior to admission ... was prescribed cortisone drops . A few days later she complained of dizziness .	12/21/2009	1/3/2010	12/30/2009	-
17	817	...with chronic mild dyspnea on exertion until two weeks prior to admission .	4/6/2012	-	4/6/2012	4/19/2012
8	142	Mother presented on day of delivery with preterm labor...	2016-05-05	-	-	2016-05-05
9	142	day of life two	2016-05-07	2016-05-06	2016-05-06	2016-05-05
	142	day of life four	2016-05-09	2016-05-08	2016-05-08	2016-05-05
	142	day of life six	2016-05-11	2016-05-10	2016-05-10	2016-05-05
	142	day of life six	2016-05-11	2016-05-10	2016-05-10	2016-05-05
10	142	day of life 18	2016-05-23	2016-05-22	2016-05-22	2016-05-05
	142	day of life four	2016-05-09	2016-05-08	2016-05-08	2016-05-05
	142	day of life seven	2016-05-12	2016-05-11	2016-05-11	2016-05-05
	142	day of life 11	2016-05-16	2016-05-15	2016-05-15	2016-05-05
	142	day of life five	2016-05-10	2016-05-09	2016-05-09	2016-05-05
	142	day of life 25	2016-05-30	2016-05-29	2016-05-29	2016-05-05
	142	day of life two	2016-05-07	2016-05-06	2016-05-06	2016-05-05
	142	day of life six	2016-05-11	2016-05-10	2016-05-10	2016-05-05
	142	day of life seven	2016-05-12	2016-05-11	2016-05-11	2016-05-05
	142	day of life two	2016-05-07	2016-05-06	2016-05-06	2016-05-05
	142	day of life two	2016-05-07	2016-05-06	2016-05-06	2016-05-05
	142	day of life three	2016-05-08	2016-05-07	2016-05-07	2016-05-05
	142	day of life seven	2016-05-12	2016-05-11	2016-05-11	2016-05-05
	142	day of life 33	2016-06-07	2016-06-06	2016-06-06	2016-06-07

Table S1. Expanded list of temporal phrases for which it was hard to correctly identify the Anchor Time and/or Delta Value. The ‘ID’ column lists the phrase ID from Table 13.



ClinBioBert Binary Seq2Seq Ttype (4x4 epochs)

		TIME	DATE	DURATION	FREQUENCY	na		P	R	F1	Support
Gold	TIME	51	32	13	1	76		0.472	0.295	0.363	173
	DATE	2	1478	63	1	703		0.666	0.658	0.662	2247
	DURATION	0	86	400	5	248		0.630	0.541	0.582	739
	FREQUENCY	0	23	38	136	142		0.544	0.401	0.462	339
	na	55	600	121	107	0					
							weighted avg	0.637	0.590	0.611	3498

BertBase Binary Seq2Seq Ttype (4x4 epochs)

		TIME	DATE	DURATION	FREQUENCY	na		P	R	F1	Support
Gold	TIME	43	39	14	1	76		0.606	0.249	0.352	173
	DATE	0	1541	58	0	648		0.640	0.686	0.662	2247
	DURATION	1	103	402	5	228		0.629	0.544	0.583	739
	FREQUENCY	0	28	31	147	133		0.602	0.434	0.504	339
	na	27	698	134	91	0					
							weighted avg	0.632	0.610	0.615	3498

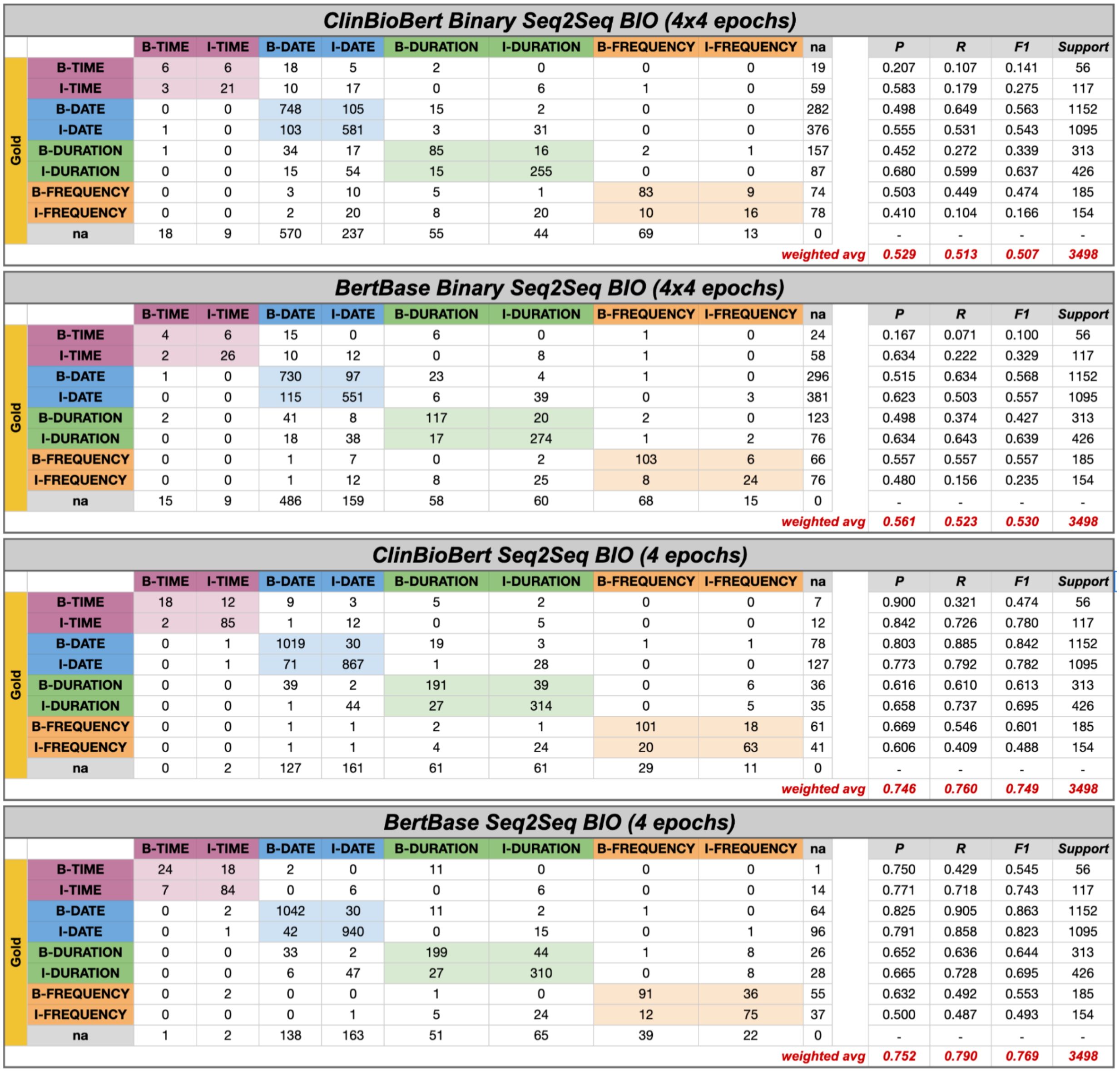
ClinBioBert Seq2Seq Ttype (4 epochs)

		TIME	DATE	DURATION	FREQUENCY	na		P	R	F1	Support
Gold	TIME	120	13	13	0	27		0.945	0.694	0.800	173
	DATE	2	2044	33	1	167		0.843	0.910	0.875	2247
	DURATION	0	79	585	16	59		0.714	0.792	0.751	739
	FREQUENCY	0	0	48	200	91		0.741	0.590	0.657	339
	na	5	290	140	53	0					
							weighted avg	0.811	0.843	0.824	3498

BertBase Seq2Seq Ttype (4 epochs)

		TIME	DATE	DURATION	FREQUENCY	na		P	R	F1	Support
Gold	TIME	132	10	14	0	17		0.910	0.763	0.830	173
	DATE	5	2058	30	1	153		0.833	0.916	0.872	2247
	DURATION	0	95	553	19	72		0.747	0.748	0.748	739
	FREQUENCY	1	0	34	212	92		0.700	0.625	0.660	339
	na	7	309	109	71	0					
							weighted avg	0.805	0.845	0.823	3498

Table S2. Results of fine-tuning BertBase and ClinBioBert baseline and binary models on the Seq2Seq multi-label classification of temporal types using the Ttype classes. Metric abbreviations: P:Precision, R:Recall



ClinBioBert Binary Seq2Seq BIO (4x4 epochs)

		B-TIME	I-TIME	B-DATE	I-DATE	B-DURATION	I-DURATION	B-FREQUENCY	I-FREQUENCY	na		P	R	F1	Support
Gold	B-TIME	6	6	18	5	2	0	0	0	19		0.207	0.107	0.141	56
	I-TIME	3	21	10	17	0	6	1	0	59		0.583	0.179	0.275	117
	B-DATE	0	0	748	105	15	2	0	0	282		0.498	0.649	0.563	1152
	I-DATE	1	0	103	581	3	31	0	0	376		0.555	0.531	0.543	1095
	B-DURATION	1	0	34	17	85	16	2	1	157		0.452	0.272	0.339	313
	I-DURATION	0	0	15	54	15	255	0	0	87		0.680	0.599	0.637	426
	B-FREQUENCY	0	0	3	10	5	1	83	9	74		0.503	0.449	0.474	185
	I-FREQUENCY	0	0	2	20	8	20	10	16	78		0.410	0.104	0.166	154
	na	18	9	570	237	55	44	69	13	0		-	-	-	-
											weighted avg	0.529	0.513	0.507	3498

BertBase Binary Seq2Seq BIO (4x4 epochs)

		B-TIME	I-TIME	B-DATE	I-DATE	B-DURATION	I-DURATION	B-FREQUENCY	I-FREQUENCY	na		P	R	F1	Support
Gold	B-TIME	4	6	15	0	6	0	1	0	24		0.167	0.071	0.100	56
	I-TIME	2	26	10	12	0	8	1	0	58		0.634	0.222	0.329	117
	B-DATE	1	0	730	97	23	4	1	0	296		0.515	0.634	0.568	1152
	I-DATE	0	0	115	551	6	39	0	3	381		0.623	0.503	0.557	1095
	B-DURATION	2	0	41	8	117	20	2	0	123		0.498	0.374	0.427	313
	I-DURATION	0	0	18	38	17	274	1	2	76		0.634	0.643	0.639	426
	B-FREQUENCY	0	0	1	7	0	2	103	6	66		0.557	0.557	0.557	185
	I-FREQUENCY	0	0	1	12	8	25	8	24	76		0.480	0.156	0.235	154
	na	15	9	486	159	58	60	68	15	0		-	-	-	-
											weighted avg	0.561	0.523	0.530	3498

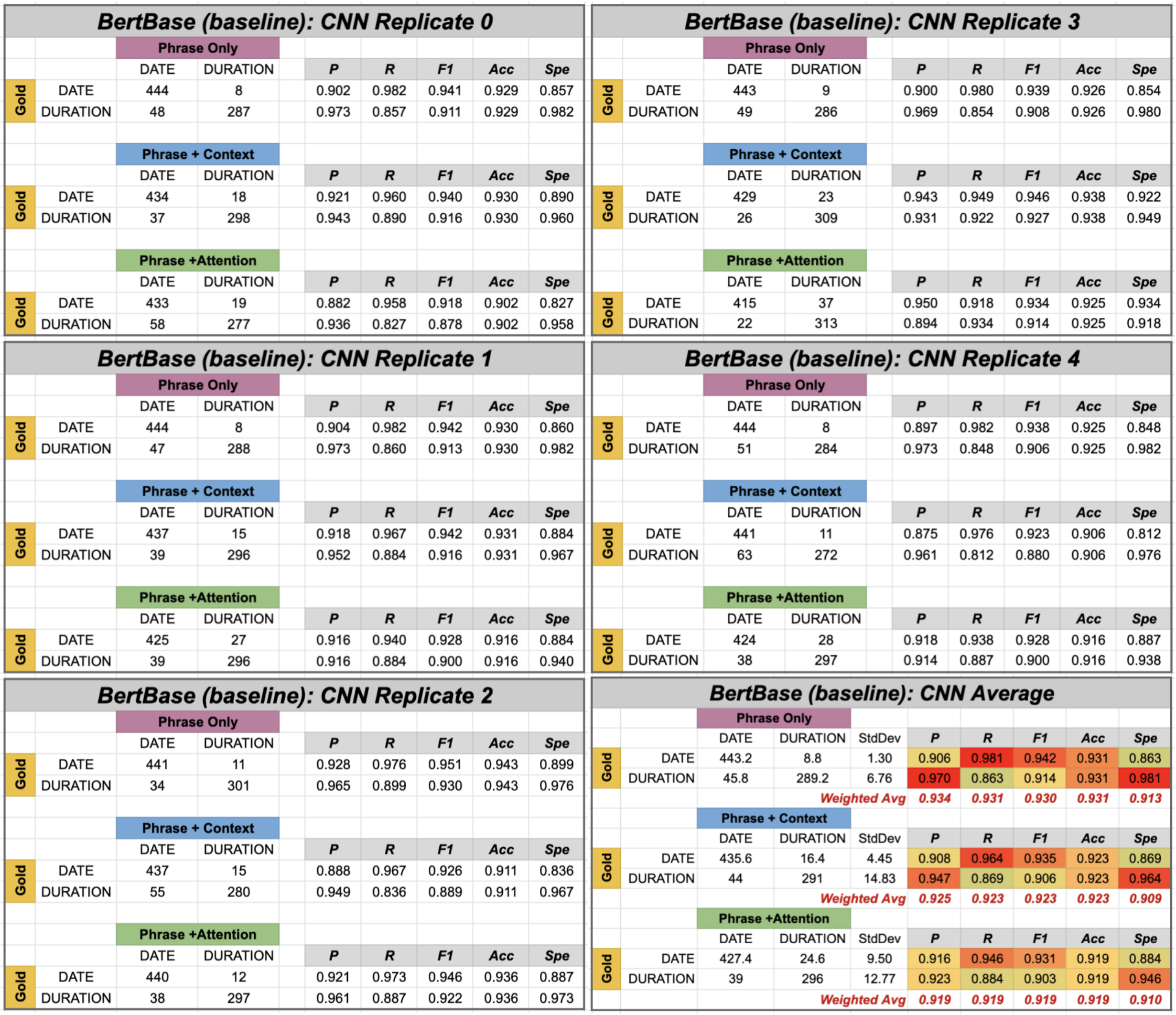
ClinBioBert Seq2Seq BIO (4 epochs)

		B-TIME	I-TIME	B-DATE	I-DATE	B-DURATION	I-DURATION	B-FREQUENCY	I-FREQUENCY	na		P	R	F1	Support
Gold	B-TIME	18	12	9	3	5	2	0	0	7		0.900	0.321	0.474	56
	I-TIME	2	85	1	12	0	5	0	0	12		0.842	0.726	0.780	117
	B-DATE	0	1	1019	30	19	3	1	1	78		0.803	0.885	0.842	1152
	I-DATE	0	1	71	867	1	28	0	0	127		0.773	0.792	0.782	1095
	B-DURATION	0	0	39	2	191	39	0	6	36		0.616	0.610	0.613	313
	I-DURATION	0	0	1	44	27	314	0	5	35		0.658	0.737	0.695	426
	B-FREQUENCY	0	0	1	1	2	1	101	18	61		0.669	0.546	0.601	185
	I-FREQUENCY	0	0	1	1	4	24	20	63	41		0.606	0.409	0.488	154
	na	0	2	127	161	61	61	29	11	0		-	-	-	-
											weighted avg	0.746	0.760	0.749	3498

BertBase Seq2Seq BIO (4 epochs)

		B-TIME	I-TIME	B-DATE	I-DATE	B-DURATION	I-DURATION	B-FREQUENCY	I-FREQUENCY	na		P	R	F1	Support
Gold	B-TIME	24	18	2	0	11	0	0	0	1		0.750	0.429	0.545	56
	I-TIME	7	84	0	6	0	6	0	0	14		0.771	0.718	0.743	117
	B-DATE	0	2	1042	30	11	2	1	0	64		0.825	0.905	0.863	1152
	I-DATE	0	1	42	940	0	15	0	1	96		0.791	0.858	0.823	1095
	B-DURATION	0	0	33	2	199	44	1	8	26		0.652	0.636	0.644	313
	I-DURATION	0	0	6	47	27	310	0	8	28		0.665	0.728	0.695	426
	B-FREQUENCY	0	2	0	0	1	0	91	36	55		0.632	0.492	0.553	185
	I-FREQUENCY	0	0	0	1	5	24	12	75	37		0.500	0.487	0.493	154
	na	1	2	138	163	51	65	39	22	0		-	-	-	-
											weighted avg	0.752	0.790	0.769	3498

Table S3. Results of fine-tuning BertBase and ClinBioBert baseline and binary models on the Seq2Seq multi-label classification of temporal types using the Beginning-Inside-Outside (BIO) classes. Metric abbreviations: P:Precision, R:Recall



BertBase (baseline): CNN Replicate 0

		DATE	DURATION	P	R	F1	Acc	Spe
		Phrase Only						
Gold	DATE	444	8	0.902	0.982	0.941	0.929	0.857
	DURATION	48	287	0.973	0.857	0.911	0.929	0.982
		Phrase + Context						
Gold	DATE	434	18	0.921	0.960	0.940	0.930	0.890
	DURATION	37	298	0.943	0.890	0.916	0.930	0.960
		Phrase +Attention						
Gold	DATE	433	19	0.882	0.958	0.918	0.902	0.827
	DURATION	58	277	0.936	0.827	0.878	0.902	0.958

BertBase (baseline): CNN Replicate 3

		DATE	DURATION	P	R	F1	Acc	Spe
		Phrase Only						
Gold	DATE	443	9	0.900	0.980	0.939	0.926	0.854
	DURATION	49	286	0.969	0.854	0.908	0.926	0.980
		Phrase + Context						
Gold	DATE	429	23	0.943	0.949	0.946	0.938	0.922
	DURATION	26	309	0.931	0.922	0.927	0.938	0.949
		Phrase +Attention						
Gold	DATE	415	37	0.950	0.918	0.934	0.925	0.934
	DURATION	22	313	0.894	0.934	0.914	0.925	0.918

BertBase (baseline): CNN Replicate 1

		DATE	DURATION	P	R	F1	Acc	Spe
		Phrase Only						
Gold	DATE	444	8	0.904	0.982	0.942	0.930	0.860
	DURATION	47	288	0.973	0.860	0.913	0.930	0.982
		Phrase + Context						
Gold	DATE	437	15	0.918	0.967	0.942	0.931	0.884
	DURATION	39	296	0.952	0.884	0.916	0.931	0.967
		Phrase +Attention						
Gold	DATE	425	27	0.916	0.940	0.928	0.916	0.884
	DURATION	39	296	0.916	0.884	0.900	0.916	0.940

BertBase (baseline): CNN Replicate 4

		DATE	DURATION	P	R	F1	Acc	Spe
		Phrase Only						
Gold	DATE	444	8	0.897	0.982	0.938	0.925	0.848
	DURATION	51	284	0.973	0.848	0.906	0.925	0.982
		Phrase + Context						
Gold	DATE	441	11	0.875	0.976	0.923	0.906	0.812
	DURATION	63	272	0.961	0.812	0.880	0.906	0.976
		Phrase +Attention						
Gold	DATE	424	28	0.918	0.938	0.928	0.916	0.887
	DURATION	38	297	0.914	0.887	0.900	0.916	0.938

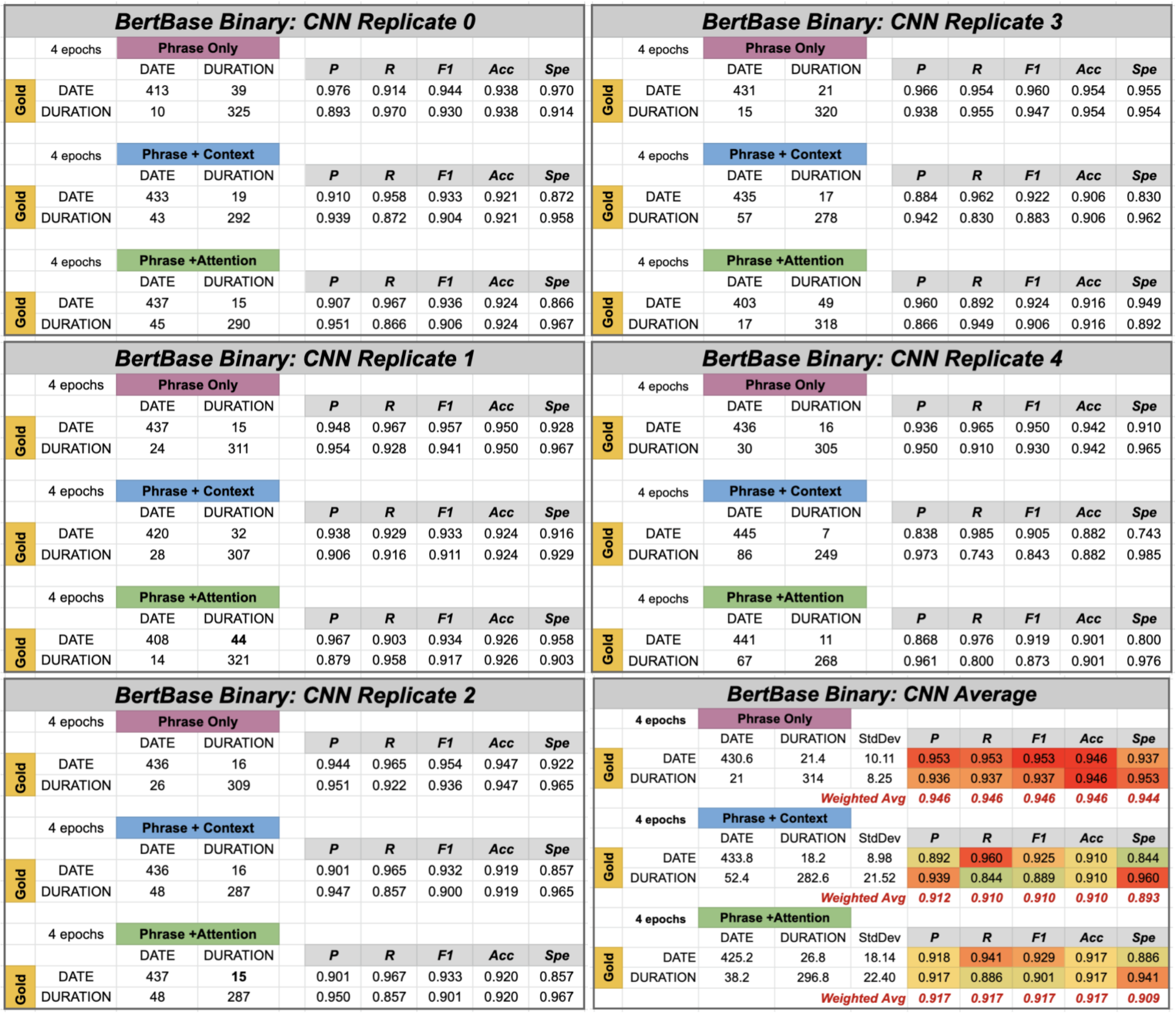
BertBase (baseline): CNN Replicate 2

		DATE	DURATION	P	R	F1	Acc	Spe
		Phrase Only						
Gold	DATE	441	11	0.928	0.976	0.951	0.943	0.899
	DURATION	34	301	0.965	0.899	0.930	0.943	0.976
		Phrase + Context						
Gold	DATE	437	15	0.888	0.967	0.926	0.911	0.836
	DURATION	55	280	0.949	0.836	0.889	0.911	0.967
		Phrase +Attention						
Gold	DATE	440	12	0.921	0.973	0.946	0.936	0.887
	DURATION	38	297	0.961	0.887	0.922	0.936	0.973

BertBase (baseline): CNN Average

		DATE	DURATION	StdDev	P	R	F1	Acc	Spe
		Phrase Only							
Gold	DATE	443.2	8.8	1.30	0.906	0.981	0.942	0.931	0.863
	DURATION	45.8	289.2	6.76	0.970	0.863	0.914	0.931	0.981
				Weighted Avg	0.934	0.931	0.930	0.931	0.913
		Phrase + Context							
Gold	DATE	435.6	16.4	4.45	0.908	0.964	0.935	0.923	0.869
	DURATION	44	291	14.83	0.947	0.869	0.906	0.923	0.964
				Weighted Avg	0.925	0.923	0.923	0.923	0.909
		Phrase +Attention							
Gold	DATE	427.4	24.6	9.50	0.916	0.946	0.931	0.919	0.884
	DURATION	39	296	12.77	0.923	0.884	0.903	0.919	0.946
				Weighted Avg	0.919	0.919	0.919	0.919	0.910

Table S4. BertBase CNN replicate model performance using RelIV-TIMEX evaluation data set. Phrase Only Hyperparameters: dropout: 0.05, kernel size1: 5, kernel size2: 2, num filters: 128, pool size: 2, stride: 2; +Context Hyperparameters: dropout: 0.1, kernel size1: 5, kernel size2: 2, num filters: 32, pool size: 3, stride: 2; +Attention Hyperparameters: dropout: 0.05, kernel size1: 5, kernel size2: 3, num filters: 128, pool size: 3, stride: 2; Metric abbreviations: P:Precision, R:Recall, Acc:Accuracy, Spe:Specificity.



BertBase Binary: CNN Replicate 0

	4 epochs	Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	413	39	0.976	0.914	0.944	0.938	0.970
	DURATION	10	325	0.893	0.970	0.930	0.938	0.914

	4 epochs	Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	433	19	0.910	0.958	0.933	0.921	0.872
	DURATION	43	292	0.939	0.872	0.904	0.921	0.958

	4 epochs	Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	437	15	0.907	0.967	0.936	0.924	0.866
	DURATION	45	290	0.951	0.866	0.906	0.924	0.967

BertBase Binary: CNN Replicate 3

	4 epochs	Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	431	21	0.966	0.954	0.960	0.954	0.955
	DURATION	15	320	0.938	0.955	0.947	0.954	0.954

	4 epochs	Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	435	17	0.884	0.962	0.922	0.906	0.830
	DURATION	57	278	0.942	0.830	0.883	0.906	0.962

	4 epochs	Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	403	49	0.960	0.892	0.924	0.916	0.949
	DURATION	17	318	0.866	0.949	0.906	0.916	0.892

BertBase Binary: CNN Replicate 1

	4 epochs	Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	437	15	0.948	0.967	0.957	0.950	0.928
	DURATION	24	311	0.954	0.928	0.941	0.950	0.967

	4 epochs	Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	420	32	0.938	0.929	0.933	0.924	0.916
	DURATION	28	307	0.906	0.916	0.911	0.924	0.929

	4 epochs	Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	408	**44**	0.967	0.903	0.934	0.926	0.958
	DURATION	14	321	0.879	0.958	0.917	0.926	0.903

BertBase Binary: CNN Replicate 4

	4 epochs	Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	436	16	0.936	0.965	0.950	0.942	0.910
	DURATION	30	305	0.950	0.910	0.930	0.942	0.965

	4 epochs	Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	445	7	0.838	0.985	0.905	0.882	0.743
	DURATION	86	249	0.973	0.743	0.843	0.882	0.985

	4 epochs	Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	441	11	0.868	0.976	0.919	0.901	0.800
	DURATION	67	268	0.961	0.800	0.873	0.901	0.976

BertBase Binary: CNN Replicate 2

	4 epochs	Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	436	16	0.944	0.965	0.954	0.947	0.922
	DURATION	26	309	0.951	0.922	0.936	0.947	0.965

	4 epochs	Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	436	16	0.901	0.965	0.932	0.919	0.857
	DURATION	48	287	0.947	0.857	0.900	0.919	0.965

	4 epochs	Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	437	**15**	0.901	0.967	0.933	0.920	0.857
	DURATION	48	287	0.950	0.857	0.901	0.920	0.967

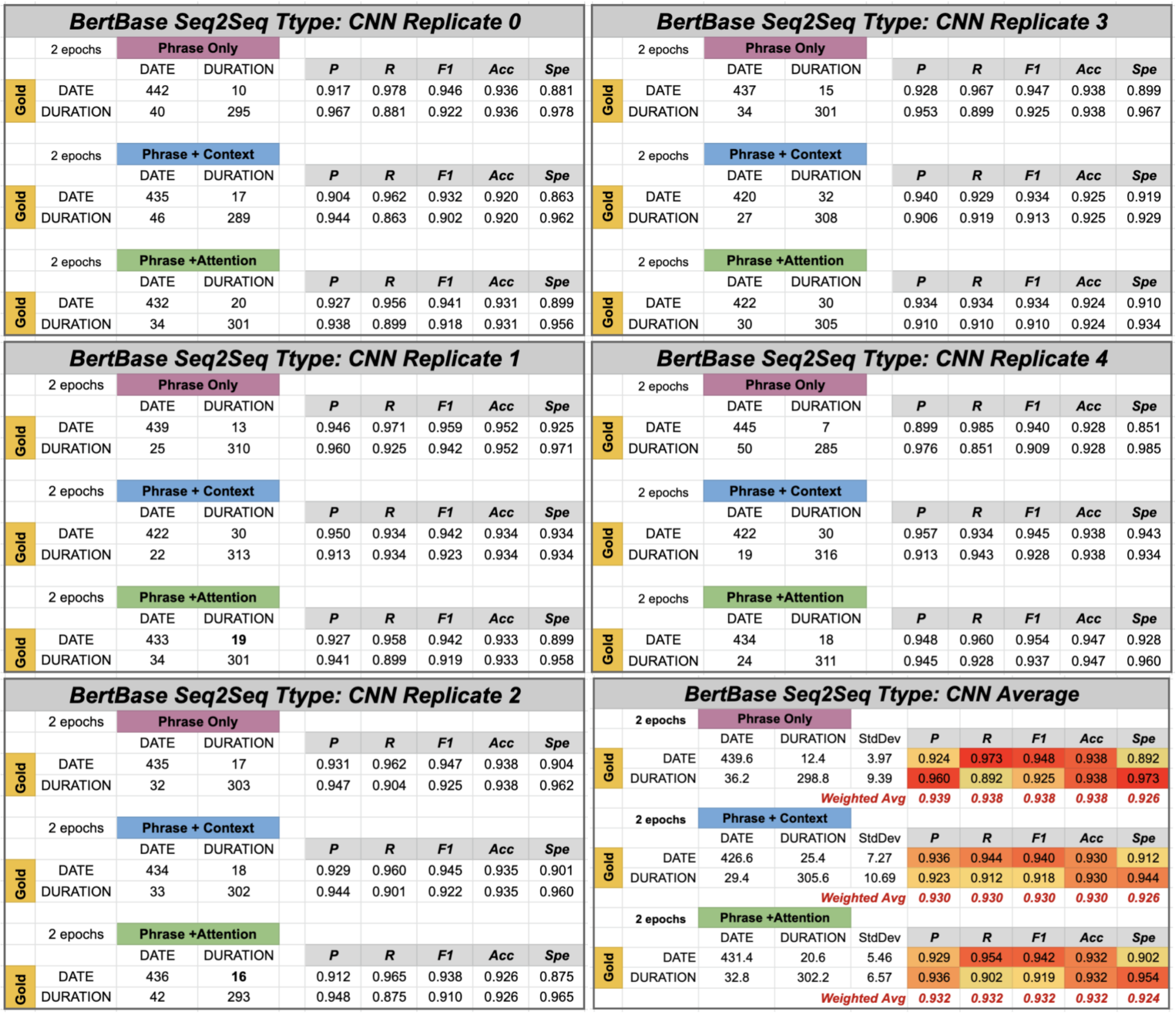
BertBase Binary: CNN Average

	4 epochs	Phrase Only							
		DATE	DURATION	StdDev	P	R	F1	Acc	Spe
Gold	DATE	430.6	21.4	10.11	0.953	0.953	0.953	0.946	0.937
	DURATION	21	314	8.25	0.936	0.937	0.937	0.946	0.953
				Weighted Avg	*0.946*	*0.946*	*0.946*	*0.946*	*0.944*

	4 epochs	Phrase + Context							
		DATE	DURATION	StdDev	P	R	F1	Acc	Spe
Gold	DATE	433.8	18.2	8.98	0.892	0.960	0.925	0.910	0.844
	DURATION	52.4	282.6	21.52	0.939	0.844	0.889	0.910	0.960
				Weighted Avg	*0.912*	*0.910*	*0.910*	*0.910*	*0.893*

	4 epochs	Phrase +Attention							
		DATE	DURATION	StdDev	P	R	F1	Acc	Spe
Gold	DATE	425.2	26.8	18.14	0.918	0.941	0.929	0.917	0.886
	DURATION	38.2	296.8	22.40	0.917	0.886	0.901	0.917	0.941
				Weighted Avg	*0.917*	*0.917*	*0.917*	*0.917*	*0.909*

Table S5. BertBase Binary CNN replicate model performance using RelIV-TIMEX evaluation data set. Phrase Only Hyperparameters: dropout: 0.1, kernel size1: 3, kernel size2: 3, num filters: 64, pool size: 2, stride: 1; +Context Hyperparameters: dropout: 0.1, kernel size1: 3, kernel size2: 2, num filters: 32, pool size: 2, stride: 2; +Attention Hyperparameters: dropout: 0.05, kernel size1: 5, kernel size2: 3, num filters: 32, pool size: 3, stride: 1; Metric abbreviations: P:Precision, R:Recall, Acc:Accuracy, Spe:Specificity.



BertBase Seq2Seq Ttype: CNN Replicate 0

	2 epochs	Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	442	10	0.917	0.978	0.946	0.936	0.881
	DURATION	40	295	0.967	0.881	0.922	0.936	0.978
	2 epochs	Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	435	17	0.904	0.962	0.932	0.920	0.863
	DURATION	46	289	0.944	0.863	0.902	0.920	0.962
	2 epochs	Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	432	20	0.927	0.956	0.941	0.931	0.899
	DURATION	34	301	0.938	0.899	0.918	0.931	0.956

BertBase Seq2Seq Ttype: CNN Replicate 3

	2 epochs	Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	437	15	0.928	0.967	0.947	0.938	0.899
	DURATION	34	301	0.953	0.899	0.925	0.938	0.967
	2 epochs	Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	420	32	0.940	0.929	0.934	0.925	0.919
	DURATION	27	308	0.906	0.919	0.913	0.925	0.929
	2 epochs	Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	422	30	0.934	0.934	0.934	0.924	0.910
	DURATION	30	305	0.910	0.910	0.910	0.924	0.934

BertBase Seq2Seq Ttype: CNN Replicate 1

	2 epochs	Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	439	13	0.946	0.971	0.959	0.952	0.925
	DURATION	25	310	0.960	0.925	0.942	0.952	0.971
	2 epochs	Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	422	30	0.950	0.934	0.942	0.934	0.934
	DURATION	22	313	0.913	0.934	0.923	0.934	0.934
	2 epochs	Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	433	**19**	0.927	0.958	0.942	0.933	0.899
	DURATION	34	301	0.941	0.899	0.919	0.933	0.958

BertBase Seq2Seq Ttype: CNN Replicate 4

	2 epochs	Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	445	7	0.899	0.985	0.940	0.928	0.851
	DURATION	50	285	0.976	0.851	0.909	0.928	0.985
	2 epochs	Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	422	30	0.957	0.934	0.945	0.938	0.943
	DURATION	19	316	0.913	0.943	0.928	0.938	0.934
	2 epochs	Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	434	18	0.948	0.960	0.954	0.947	0.928
	DURATION	24	311	0.945	0.928	0.937	0.947	0.960

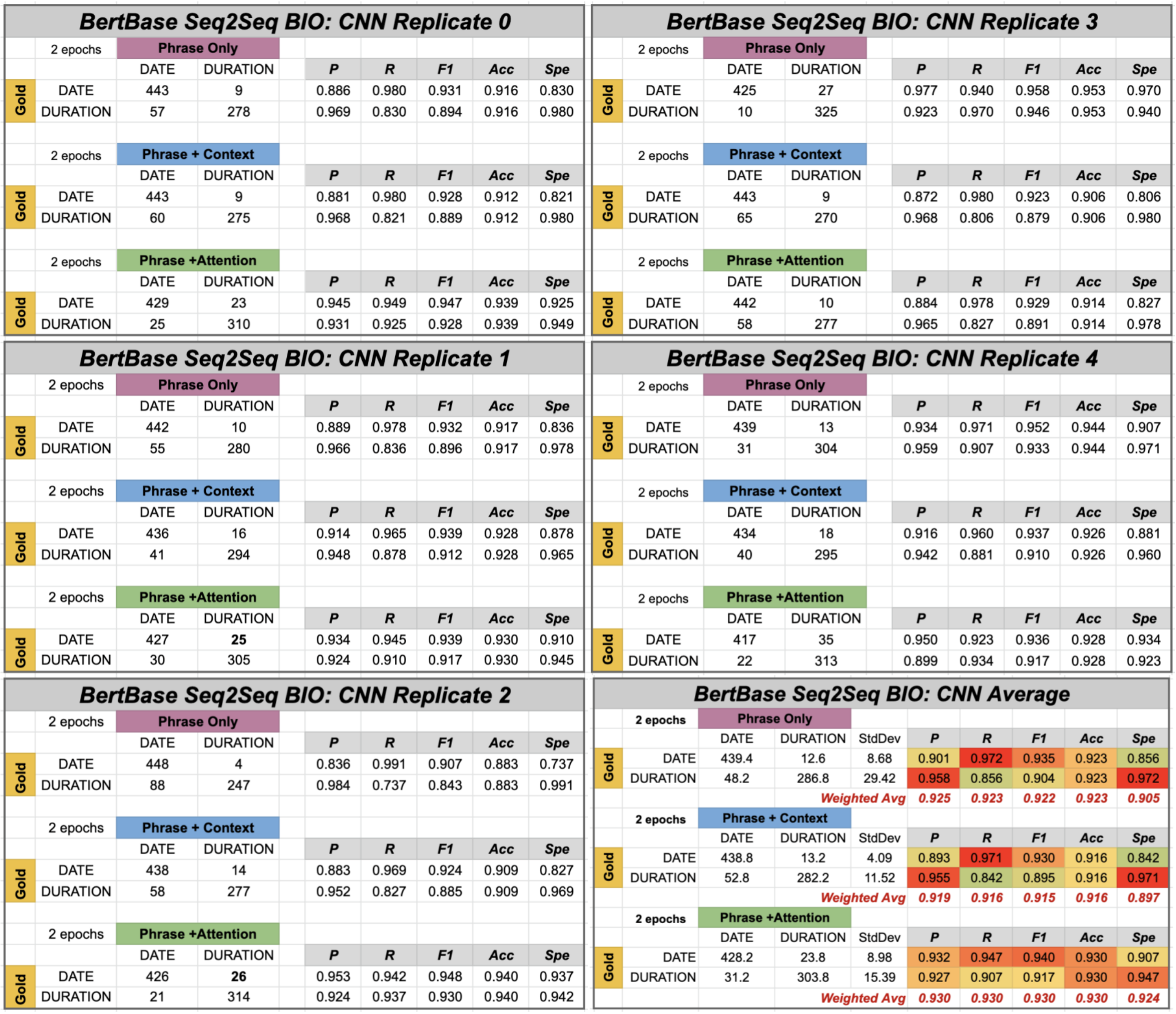
BertBase Seq2Seq Ttype: CNN Replicate 2

	2 epochs	Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	435	17	0.931	0.962	0.947	0.938	0.904
	DURATION	32	303	0.947	0.904	0.925	0.938	0.962
	2 epochs	Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	434	18	0.929	0.960	0.945	0.935	0.901
	DURATION	33	302	0.944	0.901	0.922	0.935	0.960
	2 epochs	Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	436	**16**	0.912	0.965	0.938	0.926	0.875
	DURATION	42	293	0.948	0.875	0.910	0.926	0.965

BertBase Seq2Seq Ttype: CNN Average

	2 epochs	Phrase Only							
		DATE	DURATION	StdDev	P	R	F1	Acc	Spe
Gold	DATE	439.6	12.4	3.97	0.924	0.973	0.948	0.938	0.892
	DURATION	36.2	298.8	9.39	0.960	0.892	0.925	0.938	0.973
				Weighted Avg	0.939	0.938	0.938	0.938	0.926
	2 epochs	Phrase + Context							
		DATE	DURATION	StdDev	P	R	F1	Acc	Spe
Gold	DATE	426.6	25.4	7.27	0.936	0.944	0.940	0.930	0.912
	DURATION	29.4	305.6	10.69	0.923	0.912	0.918	0.930	0.944
				Weighted Avg	0.930	0.930	0.930	0.930	0.926
	2 epochs	Phrase +Attention							
		DATE	DURATION	StdDev	P	R	F1	Acc	Spe
Gold	DATE	431.4	20.6	5.46	0.929	0.954	0.942	0.932	0.902
	DURATION	32.8	302.2	6.57	0.936	0.902	0.919	0.932	0.954
				Weighted Avg	0.932	0.932	0.932	0.932	0.924

Table S6. BertBase Seq2Seq Ttype CNN replicate model performance using RelIV–TIMEX evaluation data set. Phrase Only Hyperparameters: dropout: 0.1, kernel size1: 5, kernel size2: 2, num filters: 32, pool size: 3, stride: 1; +Context Hyperparameters: dropout: 0.1, kernel size1: 3, kernel size2: 3, num filters: 32, pool size: 3, stride: 1; +Attention Hyperparameters: dropout: 0.05, kernel size1: 5, kernel size2: 3, num filters: 128, pool size: 2, stride: 2; Metric abbreviations: P:Precision, R:Recall, Acc:Accuracy, Spe:Specificity.



BertBase Seq2Seq BIO: CNN Replicate 0

	2 epochs	Phrase Only		P	R	F1	Acc	Spe
		DATE	DURATION					
Gold	DATE	443	9	0.886	0.980	0.931	0.916	0.830
	DURATION	57	278	0.969	0.830	0.894	0.916	0.980
	2 epochs	Phrase + Context		P	R	F1	Acc	Spe
		DATE	DURATION					
Gold	DATE	443	9	0.881	0.980	0.928	0.912	0.821
	DURATION	60	275	0.968	0.821	0.889	0.912	0.980
	2 epochs	Phrase +Attention		P	R	F1	Acc	Spe
		DATE	DURATION					
Gold	DATE	429	23	0.945	0.949	0.947	0.939	0.925
	DURATION	25	310	0.931	0.925	0.928	0.939	0.949

BertBase Seq2Seq BIO: CNN Replicate 3

	2 epochs	Phrase Only		P	R	F1	Acc	Spe
		DATE	DURATION					
Gold	DATE	425	27	0.977	0.940	0.958	0.953	0.970
	DURATION	10	325	0.923	0.970	0.946	0.953	0.940
	2 epochs	Phrase + Context		P	R	F1	Acc	Spe
		DATE	DURATION					
Gold	DATE	443	9	0.872	0.980	0.923	0.906	0.806
	DURATION	65	270	0.968	0.806	0.879	0.906	0.980
	2 epochs	Phrase +Attention		P	R	F1	Acc	Spe
		DATE	DURATION					
Gold	DATE	442	10	0.884	0.978	0.929	0.914	0.827
	DURATION	58	277	0.965	0.827	0.891	0.914	0.978

BertBase Seq2Seq BIO: CNN Replicate 1

	2 epochs	Phrase Only		P	R	F1	Acc	Spe
		DATE	DURATION					
Gold	DATE	442	10	0.889	0.978	0.932	0.917	0.836
	DURATION	55	280	0.966	0.836	0.896	0.917	0.978
	2 epochs	Phrase + Context		P	R	F1	Acc	Spe
		DATE	DURATION					
Gold	DATE	436	16	0.914	0.965	0.939	0.928	0.878
	DURATION	41	294	0.948	0.878	0.912	0.928	0.965
	2 epochs	Phrase +Attention		P	R	F1	Acc	Spe
		DATE	DURATION					
Gold	DATE	427	**25**	0.934	0.945	0.939	0.930	0.910
	DURATION	30	305	0.924	0.910	0.917	0.930	0.945

BertBase Seq2Seq BIO: CNN Replicate 4

	2 epochs	Phrase Only		P	R	F1	Acc	Spe
		DATE	DURATION					
Gold	DATE	439	13	0.934	0.971	0.952	0.944	0.907
	DURATION	31	304	0.959	0.907	0.933	0.944	0.971
	2 epochs	Phrase + Context		P	R	F1	Acc	Spe
		DATE	DURATION					
Gold	DATE	434	18	0.916	0.960	0.937	0.926	0.881
	DURATION	40	295	0.942	0.881	0.910	0.926	0.960
	2 epochs	Phrase +Attention		P	R	F1	Acc	Spe
		DATE	DURATION					
Gold	DATE	417	35	0.950	0.923	0.936	0.928	0.934
	DURATION	22	313	0.899	0.934	0.917	0.928	0.923

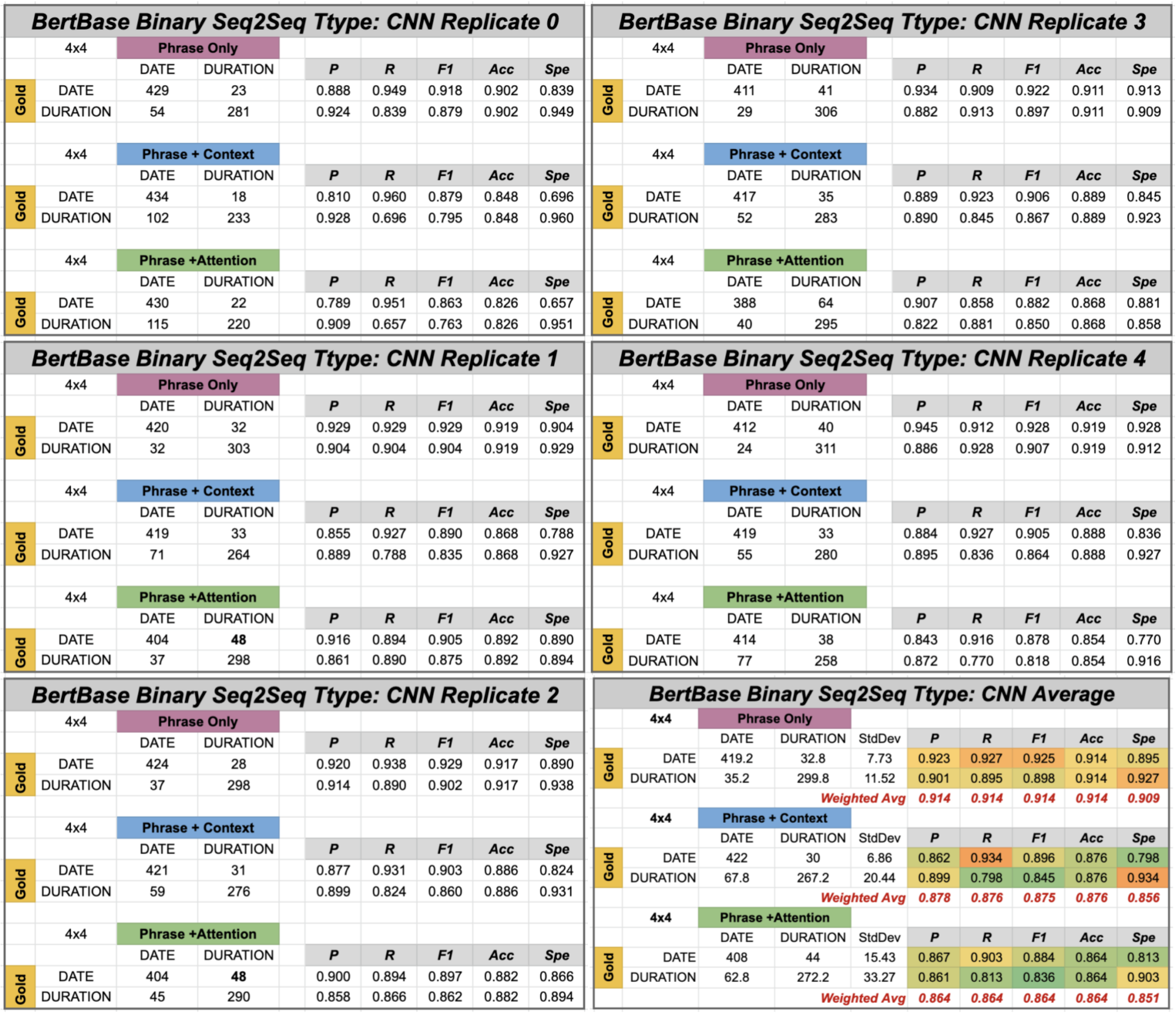
BertBase Seq2Seq BIO: CNN Replicate 2

	2 epochs	Phrase Only		P	R	F1	Acc	Spe
		DATE	DURATION					
Gold	DATE	448	4	0.836	0.991	0.907	0.883	0.737
	DURATION	88	247	0.984	0.737	0.843	0.883	0.991
	2 epochs	Phrase + Context		P	R	F1	Acc	Spe
		DATE	DURATION					
Gold	DATE	438	14	0.883	0.969	0.924	0.909	0.827
	DURATION	58	277	0.952	0.827	0.885	0.909	0.969
	2 epochs	Phrase +Attention		P	R	F1	Acc	Spe
		DATE	DURATION					
Gold	DATE	426	**26**	0.953	0.942	0.948	0.940	0.937
	DURATION	21	314	0.924	0.937	0.930	0.940	0.942

BertBase Seq2Seq BIO: CNN Average

	2 epochs	Phrase Only							
		DATE	DURATION	StdDev	P	R	F1	Acc	Spe
Gold	DATE	439.4	12.6	8.68	0.901	0.972	0.935	0.923	0.856
	DURATION	48.2	286.8	29.42	0.958	0.856	0.904	0.923	0.972
				Weighted Avg	*0.925*	*0.923*	*0.922*	*0.923*	*0.905*
	2 epochs	Phrase + Context							
		DATE	DURATION	StdDev	P	R	F1	Acc	Spe
Gold	DATE	438.8	13.2	4.09	0.893	0.971	0.930	0.916	0.842
	DURATION	52.8	282.2	11.52	0.955	0.842	0.895	0.916	0.971
				Weighted Avg	*0.919*	*0.916*	*0.915*	*0.916*	*0.897*
	2 epochs	Phrase +Attention							
		DATE	DURATION	StdDev	P	R	F1	Acc	Spe
Gold	DATE	428.2	23.8	8.98	0.932	0.947	0.940	0.930	0.907
	DURATION	31.2	303.8	15.39	0.927	0.907	0.917	0.930	0.947
				Weighted Avg	*0.930*	*0.930*	*0.930*	*0.930*	*0.924*

Table S7. BertBase Seq2Seq BIO CNN replicate model performance using RelIV–TIMEX evaluation data set. Phrase Only Hyperparameters: dropout: 0.05, kernel size1: 3, kernel size2: 3, num filters: 32, pool size: 3, stride: 2; +Context Hyperparameters: dropout: 0.05, kernel size1: 3, kernel size2: 3, num filters: 32, pool size: 2, stride: 2; +Attention Hyperparameters: dropout: 0.05, kernel size1: 3, kernel size2: 2, num filters: 64, pool size: 2, stride: 2; Metric abbreviations: P:Precision, R:Recall, Acc:Accuracy, Spe:Specificity.



BertBase Binary Seq2Seq Ttype: CNN Replicate 0								
	4x4	Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	429	23	0.888	0.949	0.918	0.902	0.839
	DURATION	54	281	0.924	0.839	0.879	0.902	0.949
	4x4	Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	434	18	0.810	0.960	0.879	0.848	0.696
	DURATION	102	233	0.928	0.696	0.795	0.848	0.960
	4x4	Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	430	22	0.789	0.951	0.863	0.826	0.657
	DURATION	115	220	0.909	0.657	0.763	0.826	0.951

BertBase Binary Seq2Seq Ttype: CNN Replicate 1								
	4x4	Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	420	32	0.929	0.929	0.929	0.919	0.904
	DURATION	32	303	0.904	0.904	0.904	0.919	0.929
	4x4	Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	419	33	0.855	0.927	0.890	0.868	0.788
	DURATION	71	264	0.889	0.788	0.835	0.868	0.927
	4x4	Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	404	**48**	0.916	0.894	0.905	0.892	0.890
	DURATION	37	298	0.861	0.890	0.875	0.892	0.894

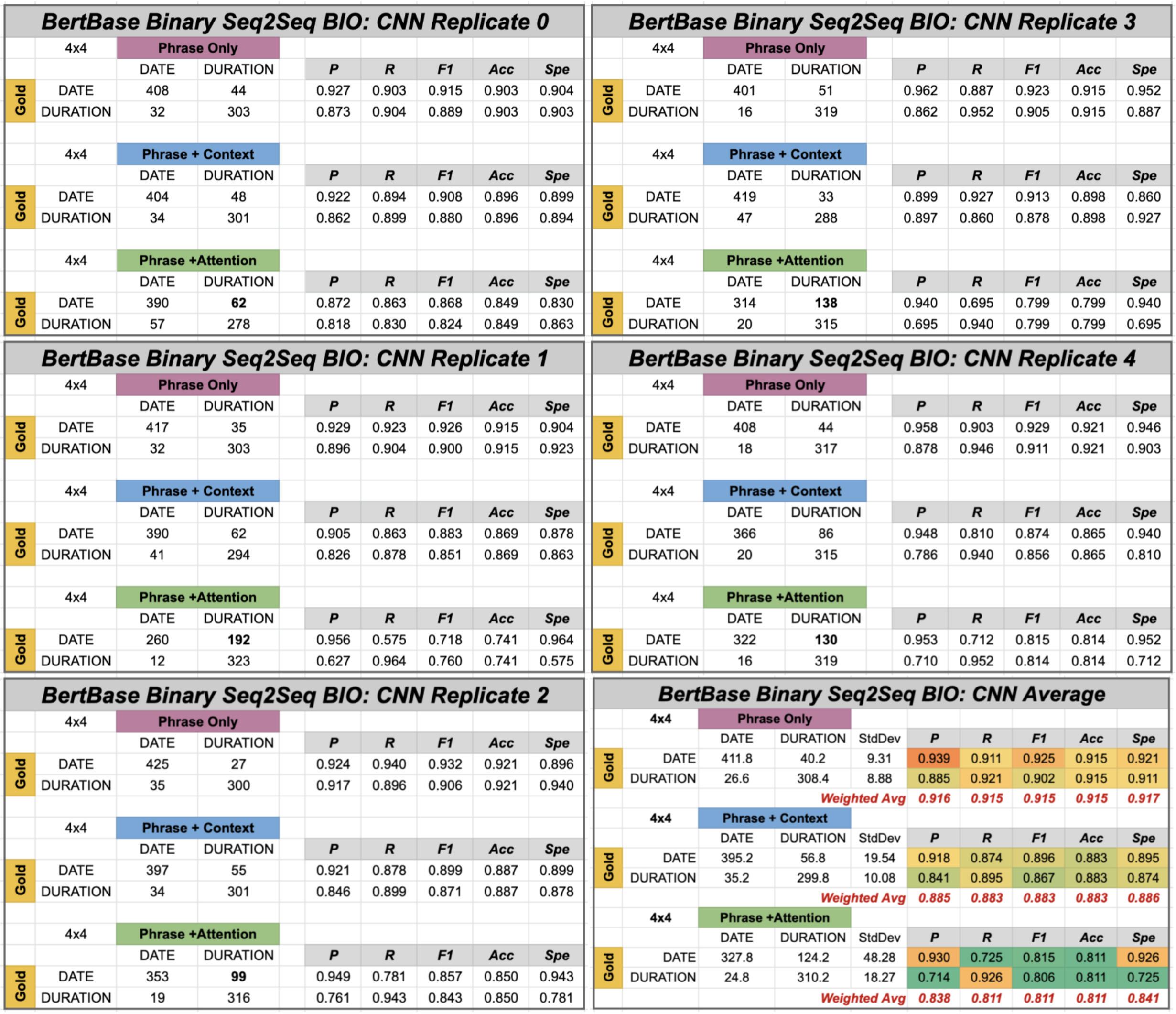
BertBase Binary Seq2Seq Ttype: CNN Replicate 2								
	4x4	Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	424	28	0.920	0.938	0.929	0.917	0.890
	DURATION	37	298	0.914	0.890	0.902	0.917	0.938
	4x4	Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	421	31	0.877	0.931	0.903	0.886	0.824
	DURATION	59	276	0.899	0.824	0.860	0.886	0.931
	4x4	Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	404	**48**	0.900	0.894	0.897	0.882	0.866
	DURATION	45	290	0.858	0.866	0.862	0.882	0.894

BertBase Binary Seq2Seq Ttype: CNN Replicate 3								
	4x4	Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	411	41	0.934	0.909	0.922	0.911	0.913
	DURATION	29	306	0.882	0.913	0.897	0.911	0.909
	4x4	Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	417	35	0.889	0.923	0.906	0.889	0.845
	DURATION	52	283	0.890	0.845	0.867	0.889	0.923
	4x4	Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	388	64	0.907	0.858	0.882	0.868	0.881
	DURATION	40	295	0.822	0.881	0.850	0.868	0.858

BertBase Binary Seq2Seq Ttype: CNN Replicate 4								
	4x4	Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	412	40	0.945	0.912	0.928	0.919	0.928
	DURATION	24	311	0.886	0.928	0.907	0.919	0.912
	4x4	Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	419	33	0.884	0.927	0.905	0.888	0.836
	DURATION	55	280	0.895	0.836	0.864	0.888	0.927
	4x4	Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	414	38	0.843	0.916	0.878	0.854	0.770
	DURATION	77	258	0.872	0.770	0.818	0.854	0.916

BertBase Binary Seq2Seq Ttype: CNN Average									
	4x4	Phrase Only							
		DATE	DURATION	StdDev	P	R	F1	Acc	Spe
Gold	DATE	419.2	32.8	7.73	0.923	0.927	0.925	0.914	0.895
	DURATION	35.2	299.8	11.52	0.901	0.895	0.898	0.914	0.927
				Weighted Avg	*0.914*	*0.914*	*0.914*	*0.914*	*0.909*
	4x4	Phrase + Context							
		DATE	DURATION	StdDev	P	R	F1	Acc	Spe
Gold	DATE	422	30	6.86	0.862	0.934	0.896	0.876	0.798
	DURATION	67.8	267.2	20.44	0.899	0.798	0.845	0.876	0.934
				Weighted Avg	*0.878*	*0.876*	*0.875*	*0.876*	*0.856*
	4x4	Phrase +Attention							
		DATE	DURATION	StdDev	P	R	F1	Acc	Spe
Gold	DATE	408	44	15.43	0.867	0.903	0.884	0.864	0.813
	DURATION	62.8	272.2	33.27	0.861	0.813	0.836	0.864	0.903
				Weighted Avg	*0.864*	*0.864*	*0.864*	*0.864*	*0.851*

Table S8. BertBase Binary Seq2Seq Ttype CNN replicate model performance using RelIV-TIMEX evaluation data set. Phrase Only Hyperparameters: dropout: 0.1, kernel size1: 3, kernel size2: 3, num filters: 32, pool size: 3, stride: 2; +Context Hyperparameters: dropout: 0.1, kernel size1: 3, kernel size2: 2, num filters: 64, pool size: 3, stride: 1; +Attention Hyperparameters: dropout: 0.1, kernel size1: 3, kernel size2: 3, num filters: 128, pool size: 3, stride: 2; Metric abbreviations: P:Precision, R:Recall, Acc:Accuracy, Spe:Specificity.



BertBase Binary Seq2Seq BIO: CNN Replicate 0

	4x4	Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	408	44	0.927	0.903	0.915	0.903	0.904
	DURATION	32	303	0.873	0.904	0.889	0.903	0.903
	4x4	Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	404	48	0.922	0.894	0.908	0.896	0.899
	DURATION	34	301	0.862	0.899	0.880	0.896	0.894
	4x4	Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	390	**62**	0.872	0.863	0.868	0.849	0.830
	DURATION	57	278	0.818	0.830	0.824	0.849	0.863

BertBase Binary Seq2Seq BIO: CNN Replicate 1

	4x4	Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	417	35	0.929	0.923	0.926	0.915	0.904
	DURATION	32	303	0.896	0.904	0.900	0.915	0.923
	4x4	Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	390	62	0.905	0.863	0.883	0.869	0.878
	DURATION	41	294	0.826	0.878	0.851	0.869	0.863
	4x4	Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	260	**192**	0.956	0.575	0.718	0.741	0.964
	DURATION	12	323	0.627	0.964	0.760	0.741	0.575

BertBase Binary Seq2Seq BIO: CNN Replicate 2

	4x4	Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	425	27	0.924	0.940	0.932	0.921	0.896
	DURATION	35	300	0.917	0.896	0.906	0.921	0.940
	4x4	Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	397	55	0.921	0.878	0.899	0.887	0.899
	DURATION	34	301	0.846	0.899	0.871	0.887	0.878
	4x4	Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	353	**99**	0.949	0.781	0.857	0.850	0.943
	DURATION	19	316	0.761	0.943	0.843	0.850	0.781

BertBase Binary Seq2Seq BIO: CNN Replicate 3

	4x4	Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	401	51	0.962	0.887	0.923	0.915	0.952
	DURATION	16	319	0.862	0.952	0.905	0.915	0.887
	4x4	Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	419	33	0.899	0.927	0.913	0.898	0.860
	DURATION	47	288	0.897	0.860	0.878	0.898	0.927
	4x4	Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	314	**138**	0.940	0.695	0.799	0.799	0.940
	DURATION	20	315	0.695	0.940	0.799	0.799	0.695

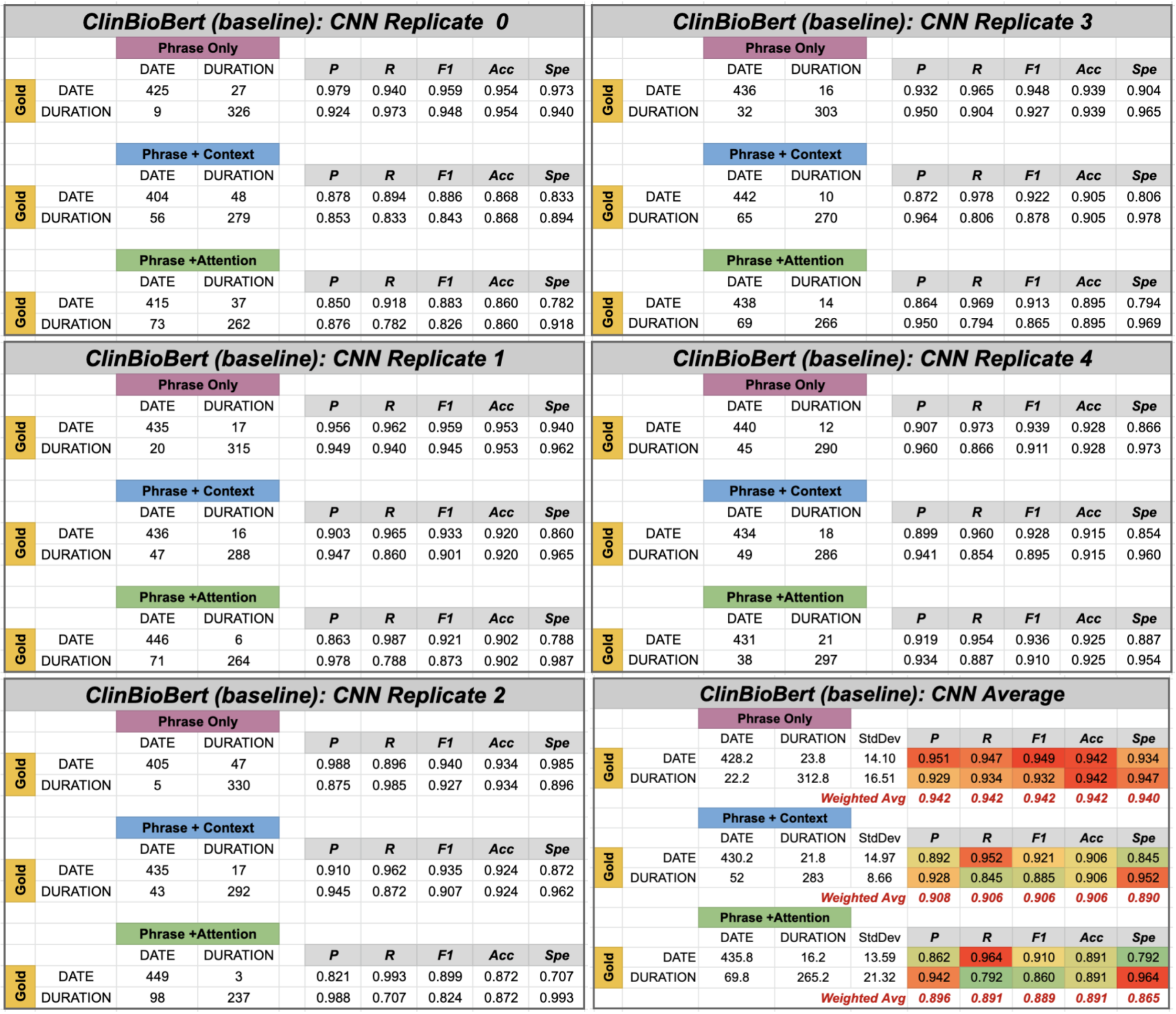
BertBase Binary Seq2Seq BIO: CNN Replicate 4

	4x4	Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	408	44	0.958	0.903	0.929	0.921	0.946
	DURATION	18	317	0.878	0.946	0.911	0.921	0.903
	4x4	Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	366	86	0.948	0.810	0.874	0.865	0.940
	DURATION	20	315	0.786	0.940	0.856	0.865	0.810
	4x4	Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	322	**130**	0.953	0.712	0.815	0.814	0.952
	DURATION	16	319	0.710	0.952	0.814	0.814	0.712

BertBase Binary Seq2Seq BIO: CNN Average

	4x4	Phrase Only							
		DATE	DURATION	StdDev	P	R	F1	Acc	Spe
Gold	DATE	411.8	40.2	9.31	0.939	0.911	0.925	0.915	0.921
	DURATION	26.6	308.4	8.88	0.885	0.921	0.902	0.915	0.911
				Weighted Avg	***0.916***	***0.915***	***0.915***	***0.915***	***0.917***
	4x4	Phrase + Context							
		DATE	DURATION	StdDev	P	R	F1	Acc	Spe
Gold	DATE	395.2	56.8	19.54	0.918	0.874	0.896	0.883	0.895
	DURATION	35.2	299.8	10.08	0.841	0.895	0.867	0.883	0.874
				Weighted Avg	***0.885***	***0.883***	***0.883***	***0.883***	***0.886***
	4x4	Phrase +Attention							
		DATE	DURATION	StdDev	P	R	F1	Acc	Spe
Gold	DATE	327.8	124.2	48.28	0.930	0.725	0.815	0.811	0.926
	DURATION	24.8	310.2	18.27	0.714	0.926	0.806	0.811	0.725
				Weighted Avg	***0.838***	***0.811***	***0.811***	***0.811***	***0.841***

Table S9. BertBase Binary Seq2Seq BIO CNN replicate model performance using RelIV-TIMEX evaluation data set. Phrase Only Hyperparameters: dropout: 0.05, kernel size1: 3, kernel size2: 3, num filters: 64, pool size: 3, stride: 1; +Context Hyperparameters: dropout: 0.05, kernel size1: 5, kernel size2: 2, num filters: 64, pool size: 2, stride: 2; +Attention Hyperparameters: dropout: 0.1, kernel size1: 5, kernel size2: 3, num filters: 64, pool size: 3, stride: 1; Metric abbreviations: P:Precision, R:Recall, Acc:Accuracy, Spe:Specificity.



ClinBioBert (baseline): CNN Replicate 0

		Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	425	27	0.979	0.940	0.959	0.954	0.973
	DURATION	9	326	0.924	0.973	0.948	0.954	0.940
		Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	404	48	0.878	0.894	0.886	0.868	0.833
	DURATION	56	279	0.853	0.833	0.843	0.868	0.894
		Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	415	37	0.850	0.918	0.883	0.860	0.782
	DURATION	73	262	0.876	0.782	0.826	0.860	0.918

ClinBioBert (baseline): CNN Replicate 1

		Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	435	17	0.956	0.962	0.959	0.953	0.940
	DURATION	20	315	0.949	0.940	0.945	0.953	0.962
		Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	436	16	0.903	0.965	0.933	0.920	0.860
	DURATION	47	288	0.947	0.860	0.901	0.920	0.965
		Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	446	6	0.863	0.987	0.921	0.902	0.788
	DURATION	71	264	0.978	0.788	0.873	0.902	0.987

ClinBioBert (baseline): CNN Replicate 2

		Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	405	47	0.988	0.896	0.940	0.934	0.985
	DURATION	5	330	0.875	0.985	0.927	0.934	0.896
		Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	435	17	0.910	0.962	0.935	0.924	0.872
	DURATION	43	292	0.945	0.872	0.907	0.924	0.962
		Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	449	3	0.821	0.993	0.899	0.872	0.707
	DURATION	98	237	0.988	0.707	0.824	0.872	0.993

ClinBioBert (baseline): CNN Replicate 3

		Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	436	16	0.932	0.965	0.948	0.939	0.904
	DURATION	32	303	0.950	0.904	0.927	0.939	0.965
		Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	442	10	0.872	0.978	0.922	0.905	0.806
	DURATION	65	270	0.964	0.806	0.878	0.905	0.978
		Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	438	14	0.864	0.969	0.913	0.895	0.794
	DURATION	69	266	0.950	0.794	0.865	0.895	0.969

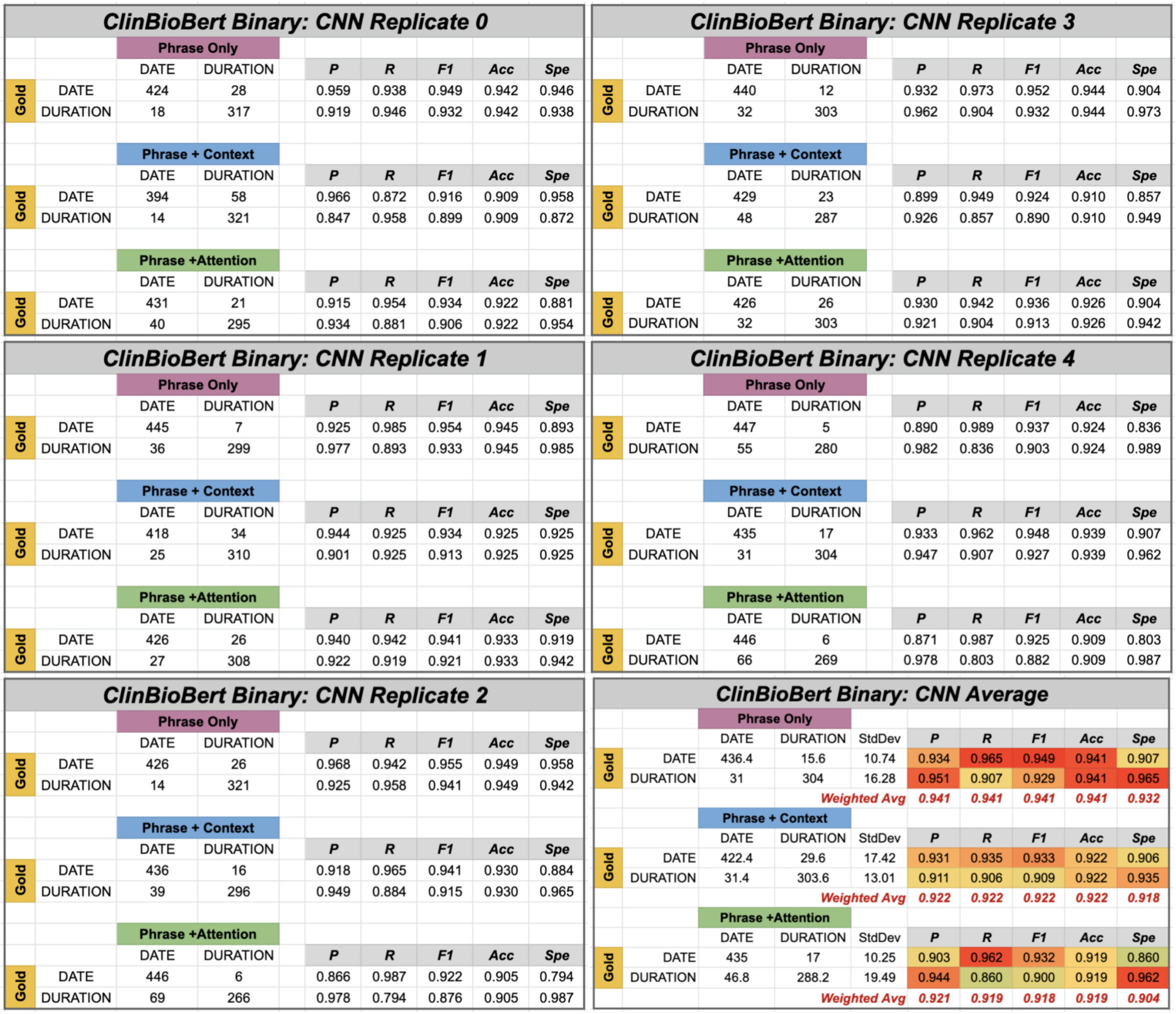
ClinBioBert (baseline): CNN Replicate 4

		Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	440	12	0.907	0.973	0.939	0.928	0.866
	DURATION	45	290	0.960	0.866	0.911	0.928	0.973
		Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	434	18	0.899	0.960	0.928	0.915	0.854
	DURATION	49	286	0.941	0.854	0.895	0.915	0.960
		Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	431	21	0.919	0.954	0.936	0.925	0.887
	DURATION	38	297	0.934	0.887	0.910	0.925	0.954

ClinBioBert (baseline): CNN Average

		Phrase Only							
		DATE	DURATION	StdDev	P	R	F1	Acc	Spe
Gold	DATE	428.2	23.8	14.10	0.951	0.947	0.949	0.942	0.934
	DURATION	22.2	312.8	16.51	0.929	0.934	0.932	0.942	0.947
				Weighted Avg	*0.942*	*0.942*	*0.942*	*0.942*	*0.940*
		Phrase + Context							
		DATE	DURATION	StdDev	P	R	F1	Acc	Spe
Gold	DATE	430.2	21.8	14.97	0.892	0.952	0.921	0.906	0.845
	DURATION	52	283	8.66	0.928	0.845	0.885	0.906	0.952
				Weighted Avg	*0.908*	*0.906*	*0.906*	*0.906*	*0.890*
		Phrase +Attention							
		DATE	DURATION	StdDev	P	R	F1	Acc	Spe
Gold	DATE	435.8	16.2	13.59	0.862	0.964	0.910	0.891	0.792
	DURATION	69.8	265.2	21.32	0.942	0.792	0.860	0.891	0.964
				Weighted Avg	*0.896*	*0.891*	*0.889*	*0.891*	*0.865*

Table S10. ClinBioBert CNN replicate model performance using RelIV-TIMEX evaluation data set. Phrase Only Hyperparameters: dropout: 0.05, kernel size1: 5, kernel size2: 3, num filters: 128, pool size: 3, stride: 1; +Context Hyperparameters: dropout: 0.05, kernel size1: 3, kernel size2: 3, num filters: 32, pool size: 2, stride: 2; +Attention Hyperparameters: dropout: 0.1, kernel size1: 3, kernel size2: 2, num filters: 64, pool size: 2, stride: 1; Metric abbreviations: P:Precision, R:Recall, Acc:Accuracy, Spe:Specificity.



ClinBioBert Binary: CNN Replicate 0

		Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	424	28	0.959	0.938	0.949	0.942	0.946
	DURATION	18	317	0.919	0.946	0.932	0.942	0.938
		Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	394	58	0.966	0.872	0.916	0.909	0.958
	DURATION	14	321	0.847	0.958	0.899	0.909	0.872
		Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	431	21	0.915	0.954	0.934	0.922	0.881
	DURATION	40	295	0.934	0.881	0.906	0.922	0.954

ClinBioBert Binary: CNN Replicate 3

		Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	440	12	0.932	0.973	0.952	0.944	0.904
	DURATION	32	303	0.962	0.904	0.932	0.944	0.973
		Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	429	23	0.899	0.949	0.924	0.910	0.857
	DURATION	48	287	0.926	0.857	0.890	0.910	0.949
		Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	426	26	0.930	0.942	0.936	0.926	0.904
	DURATION	32	303	0.921	0.904	0.913	0.926	0.942

ClinBioBert Binary: CNN Replicate 1

		Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	445	7	0.925	0.985	0.954	0.945	0.893
	DURATION	36	299	0.977	0.893	0.933	0.945	0.985
		Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	418	34	0.944	0.925	0.934	0.925	0.925
	DURATION	25	310	0.901	0.925	0.913	0.925	0.925
		Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	426	26	0.940	0.942	0.941	0.933	0.919
	DURATION	27	308	0.922	0.919	0.921	0.933	0.942

ClinBioBert Binary: CNN Replicate 4

		Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	447	5	0.890	0.989	0.937	0.924	0.836
	DURATION	55	280	0.982	0.836	0.903	0.924	0.989
		Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	435	17	0.933	0.962	0.948	0.939	0.907
	DURATION	31	304	0.947	0.907	0.927	0.939	0.962
		Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	446	6	0.871	0.987	0.925	0.909	0.803
	DURATION	66	269	0.978	0.803	0.882	0.909	0.987

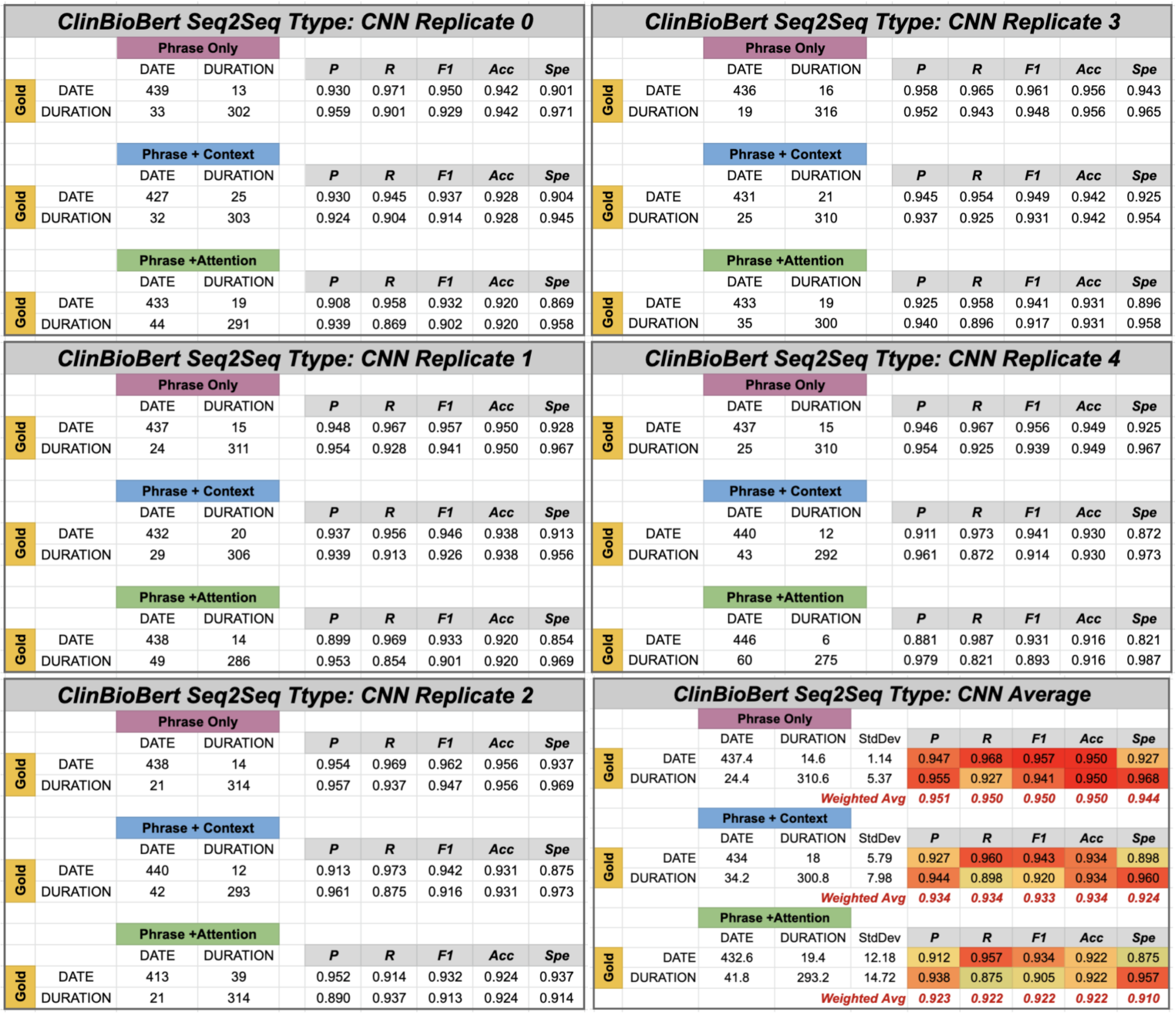
ClinBioBert Binary: CNN Replicate 2

		Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	426	26	0.968	0.942	0.955	0.949	0.958
	DURATION	14	321	0.925	0.958	0.941	0.949	0.942
		Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	436	16	0.918	0.965	0.941	0.930	0.884
	DURATION	39	296	0.949	0.884	0.915	0.930	0.965
		Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	446	6	0.866	0.987	0.922	0.905	0.794
	DURATION	69	266	0.978	0.794	0.876	0.905	0.987

ClinBioBert Binary: CNN Average

		Phrase Only							
		DATE	DURATION	StdDev	P	R	F1	Acc	Spe
Gold	DATE	436.4	15.6	10.74	0.934	0.965	0.949	0.941	0.907
	DURATION	31	304	16.28	0.951	0.907	0.929	0.941	0.965
				Weighted Avg	*0.941*	*0.941*	*0.941*	*0.941*	*0.932*
		Phrase + Context							
		DATE	DURATION	StdDev	P	R	F1	Acc	Spe
Gold	DATE	422.4	29.6	17.42	0.931	0.935	0.933	0.922	0.906
	DURATION	31.4	303.6	13.01	0.911	0.906	0.909	0.922	0.935
				Weighted Avg	*0.922*	*0.922*	*0.922*	*0.922*	*0.918*
		Phrase +Attention							
		DATE	DURATION	StdDev	P	R	F1	Acc	Spe
Gold	DATE	435	17	10.25	0.903	0.962	0.932	0.919	0.860
	DURATION	46.8	288.2	19.49	0.944	0.860	0.900	0.919	0.962
				Weighted Avg	*0.921*	*0.919*	*0.918*	*0.919*	*0.904*

Table S11. ClinBioBert Binary CNN replicate model performance using RelIV–TIMEX evaluation data set. Phrase Only Hyperparameters: dropout: 0.05, kernel size1: 5, kernel size2: 3, num filters: 32, pool size: 3, stride: 1; +Context Hyperparameters: dropout: 0.05, kernel size1: 5, kernel size2: 3, num filters: 32, pool size: 3, stride: 1; +Attention Hyperparameters: dropout: 0.1, kernel size1: 3, kernel size2: 2, num filters: 64, pool size: 3, stride: 1; Metric abbreviations: P:Precision, R:Recall, Acc:Accuracy, Spe:Specificity.



ClinBioBert Seq2Seq Ttype: CNN Replicate 0

		DATE	DURATION	P	R	F1	Acc	Spe
		Phrase Only						
Gold	DATE	439	13	0.930	0.971	0.950	0.942	0.901
	DURATION	33	302	0.959	0.901	0.929	0.942	0.971
		Phrase + Context						
Gold	DATE	427	25	0.930	0.945	0.937	0.928	0.904
	DURATION	32	303	0.924	0.904	0.914	0.928	0.945
		Phrase +Attention						
Gold	DATE	433	19	0.908	0.958	0.932	0.920	0.869
	DURATION	44	291	0.939	0.869	0.902	0.920	0.958

ClinBioBert Seq2Seq Ttype: CNN Replicate 3

		DATE	DURATION	P	R	F1	Acc	Spe
		Phrase Only						
Gold	DATE	436	16	0.958	0.965	0.961	0.956	0.943
	DURATION	19	316	0.952	0.943	0.948	0.956	0.965
		Phrase + Context						
Gold	DATE	431	21	0.945	0.954	0.949	0.942	0.925
	DURATION	25	310	0.937	0.925	0.931	0.942	0.954
		Phrase +Attention						
Gold	DATE	433	19	0.925	0.958	0.941	0.931	0.896
	DURATION	35	300	0.940	0.896	0.917	0.931	0.958

ClinBioBert Seq2Seq Ttype: CNN Replicate 1

		DATE	DURATION	P	R	F1	Acc	Spe
		Phrase Only						
Gold	DATE	437	15	0.948	0.967	0.957	0.950	0.928
	DURATION	24	311	0.954	0.928	0.941	0.950	0.967
		Phrase + Context						
Gold	DATE	432	20	0.937	0.956	0.946	0.938	0.913
	DURATION	29	306	0.939	0.913	0.926	0.938	0.956
		Phrase +Attention						
Gold	DATE	438	14	0.899	0.969	0.933	0.920	0.854
	DURATION	49	286	0.953	0.854	0.901	0.920	0.969

ClinBioBert Seq2Seq Ttype: CNN Replicate 4

		DATE	DURATION	P	R	F1	Acc	Spe
		Phrase Only						
Gold	DATE	437	15	0.946	0.967	0.956	0.949	0.925
	DURATION	25	310	0.954	0.925	0.939	0.949	0.967
		Phrase + Context						
Gold	DATE	440	12	0.911	0.973	0.941	0.930	0.872
	DURATION	43	292	0.961	0.872	0.914	0.930	0.973
		Phrase +Attention						
Gold	DATE	446	6	0.881	0.987	0.931	0.916	0.821
	DURATION	60	275	0.979	0.821	0.893	0.916	0.987

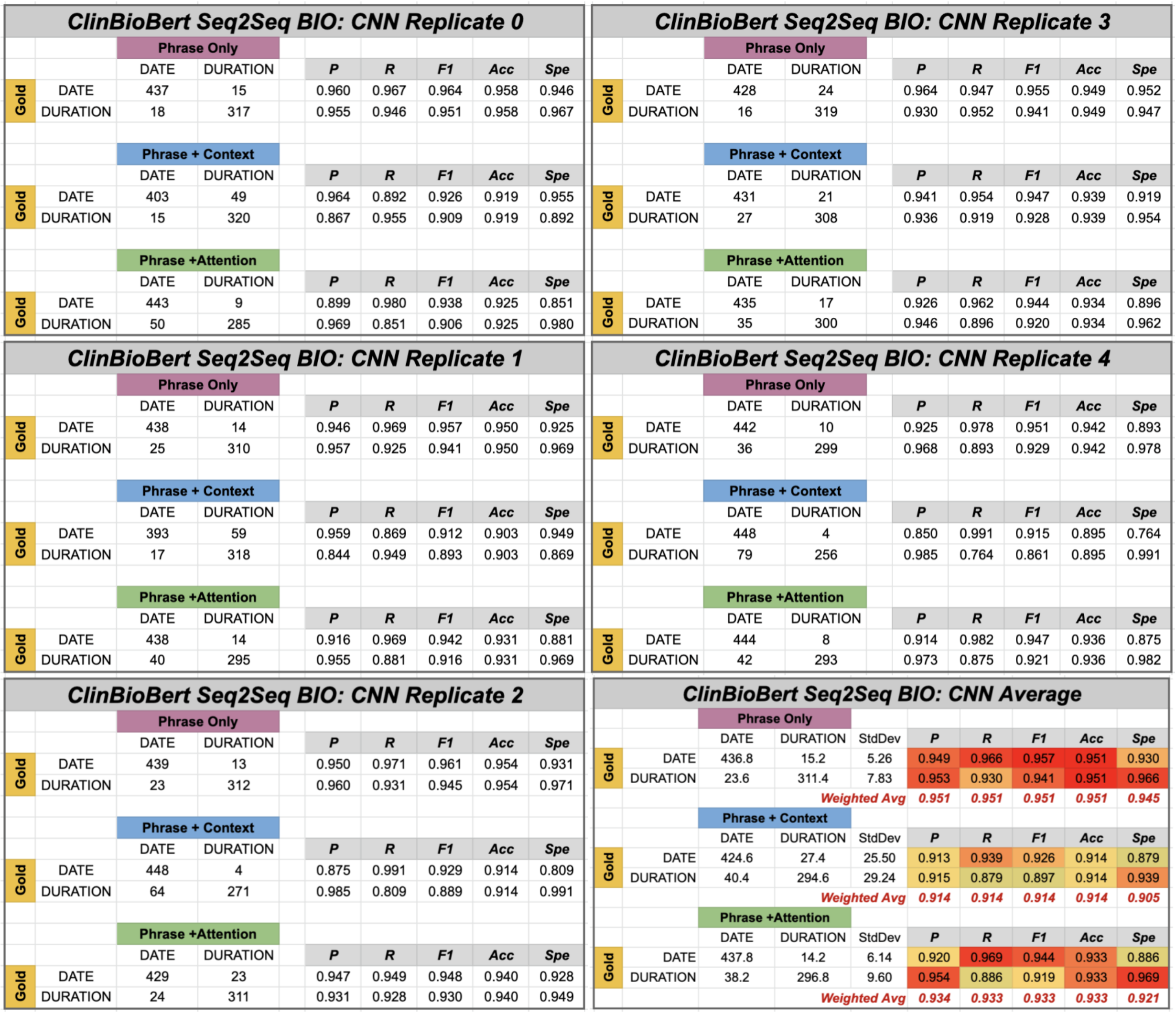
ClinBioBert Seq2Seq Ttype: CNN Replicate 2

		DATE	DURATION	P	R	F1	Acc	Spe
		Phrase Only						
Gold	DATE	438	14	0.954	0.969	0.962	0.956	0.937
	DURATION	21	314	0.957	0.937	0.947	0.956	0.969
		Phrase + Context						
Gold	DATE	440	12	0.913	0.973	0.942	0.931	0.875
	DURATION	42	293	0.961	0.875	0.916	0.931	0.973
		Phrase +Attention						
Gold	DATE	413	39	0.952	0.914	0.932	0.924	0.937
	DURATION	21	314	0.890	0.937	0.913	0.924	0.914

ClinBioBert Seq2Seq Ttype: CNN Average

		DATE	DURATION	StdDev	P	R	F1	Acc	Spe
		Phrase Only							
Gold	DATE	437.4	14.6	1.14	0.947	0.968	0.957	0.950	0.927
	DURATION	24.4	310.6	5.37	0.955	0.927	0.941	0.950	0.968
				Weighted Avg	*0.951*	*0.950*	*0.950*	*0.950*	*0.944*
		Phrase + Context							
Gold	DATE	434	18	5.79	0.927	0.960	0.943	0.934	0.898
	DURATION	34.2	300.8	7.98	0.944	0.898	0.920	0.934	0.960
				Weighted Avg	*0.934*	*0.934*	*0.933*	*0.934*	*0.924*
		Phrase +Attention							
Gold	DATE	432.6	19.4	12.18	0.912	0.957	0.934	0.922	0.875
	DURATION	41.8	293.2	14.72	0.938	0.875	0.905	0.922	0.957
				Weighted Avg	*0.923*	*0.922*	*0.922*	*0.922*	*0.910*

Table S12. ClinBioBert Seq2Seq Ttype CNN replicate model performance using RelIV-TIMEX evaluation data set. Phrase Only Hyperparameters: dropout: 0.1, kernel size1: 5, kernel size2: 3, num filters: 32, pool size: 3, stride: 1; +Context Hyperparameters: dropout: 0.1, kernel size1: 5, kernel size2: 2, num filters: 64, pool size: 3, stride: 1; +Attention Hyperparameters: dropout: 0.05, kernel size1: 5, kernel size2: 3, num filters: 32, pool size: 3, stride: 2; Metric abbreviations: P:Precision, R:Recall, Acc:Accuracy, Spe:Specificity.



ClinBioBert Seq2Seq BIO: CNN Replicate 0								
		Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	437	15	0.960	0.967	0.964	0.958	0.946
	DURATION	18	317	0.955	0.946	0.951	0.958	0.967
		Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	403	49	0.964	0.892	0.926	0.919	0.955
	DURATION	15	320	0.867	0.955	0.909	0.919	0.892
		Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	443	9	0.899	0.980	0.938	0.925	0.851
	DURATION	50	285	0.969	0.851	0.906	0.925	0.980

ClinBioBert Seq2Seq BIO: CNN Replicate 3								
		Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	428	24	0.964	0.947	0.955	0.949	0.952
	DURATION	16	319	0.930	0.952	0.941	0.949	0.947
		Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	431	21	0.941	0.954	0.947	0.939	0.919
	DURATION	27	308	0.936	0.919	0.928	0.939	0.954
		Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	435	17	0.926	0.962	0.944	0.934	0.896
	DURATION	35	300	0.946	0.896	0.920	0.934	0.962

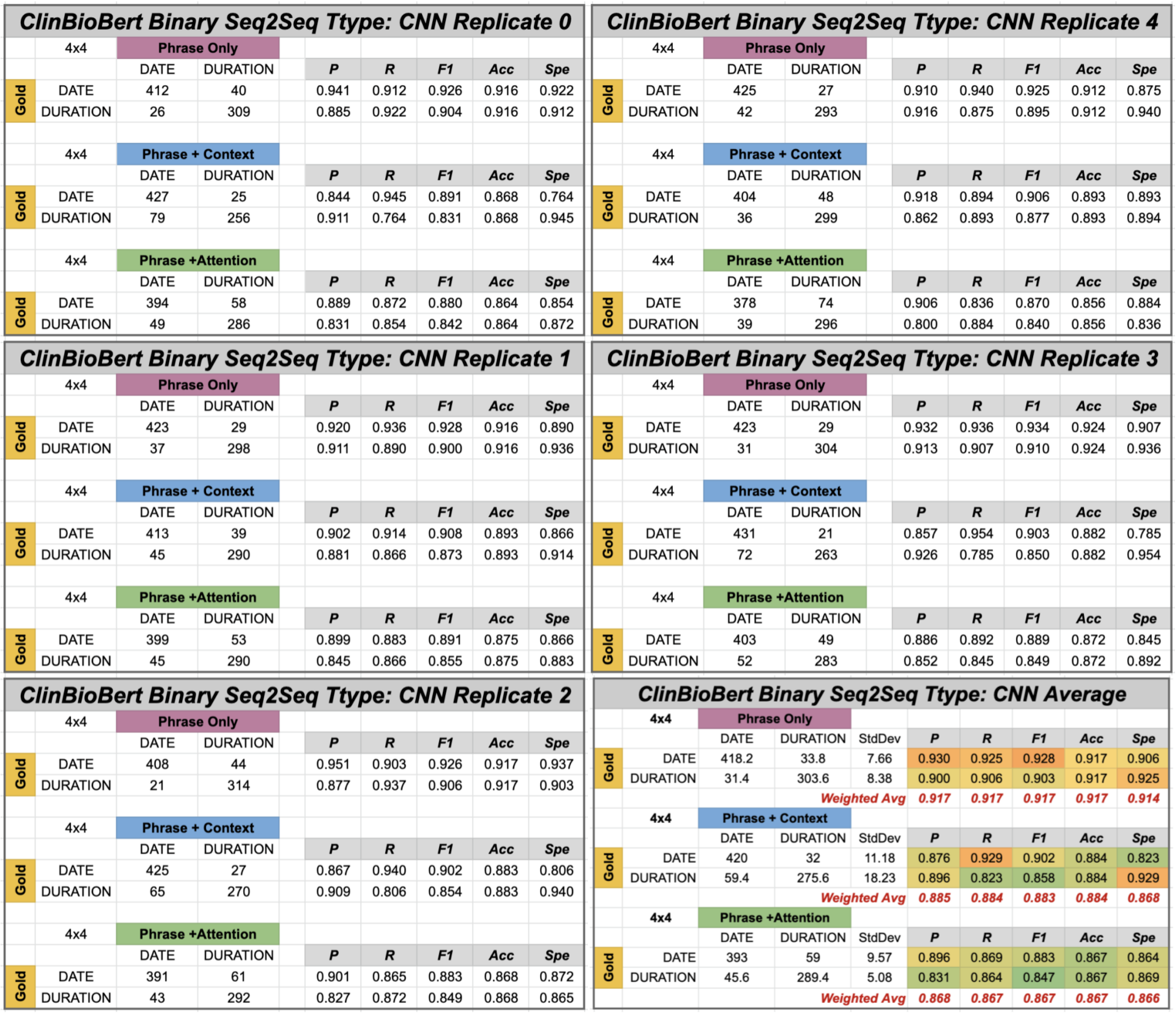
ClinBioBert Seq2Seq BIO: CNN Replicate 1								
		Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	438	14	0.946	0.969	0.957	0.950	0.925
	DURATION	25	310	0.957	0.925	0.941	0.950	0.969
		Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	393	59	0.959	0.869	0.912	0.903	0.949
	DURATION	17	318	0.844	0.949	0.893	0.903	0.869
		Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	438	14	0.916	0.969	0.942	0.931	0.881
	DURATION	40	295	0.955	0.881	0.916	0.931	0.969

ClinBioBert Seq2Seq BIO: CNN Replicate 4								
		Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	442	10	0.925	0.978	0.951	0.942	0.893
	DURATION	36	299	0.968	0.893	0.929	0.942	0.978
		Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	448	4	0.850	0.991	0.915	0.895	0.764
	DURATION	79	256	0.985	0.764	0.861	0.895	0.991
		Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	444	8	0.914	0.982	0.947	0.936	0.875
	DURATION	42	293	0.973	0.875	0.921	0.936	0.982

ClinBioBert Seq2Seq BIO: CNN Replicate 2								
		Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	439	13	0.950	0.971	0.961	0.954	0.931
	DURATION	23	312	0.960	0.931	0.945	0.954	0.971
		Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	448	4	0.875	0.991	0.929	0.914	0.809
	DURATION	64	271	0.985	0.809	0.889	0.914	0.991
		Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	429	23	0.947	0.949	0.948	0.940	0.928
	DURATION	24	311	0.931	0.928	0.930	0.940	0.949

ClinBioBert Seq2Seq BIO: CNN Average									
		Phrase Only							
		DATE	DURATION	StdDev	P	R	F1	Acc	Spe
Gold	DATE	436.8	15.2	5.26	0.949	0.966	0.957	0.951	0.930
	DURATION	23.6	311.4	7.83	0.953	0.930	0.941	0.951	0.966
				Weighted Avg	*0.951*	*0.951*	*0.951*	*0.951*	*0.945*
		Phrase + Context							
		DATE	DURATION	StdDev	P	R	F1	Acc	Spe
Gold	DATE	424.6	27.4	25.50	0.913	0.939	0.926	0.914	0.879
	DURATION	40.4	294.6	29.24	0.915	0.879	0.897	0.914	0.939
				Weighted Avg	*0.914*	*0.914*	*0.914*	*0.914*	*0.905*
		Phrase +Attention							
		DATE	DURATION	StdDev	P	R	F1	Acc	Spe
Gold	DATE	437.8	14.2	6.14	0.920	0.969	0.944	0.933	0.886
	DURATION	38.2	296.8	9.60	0.954	0.886	0.919	0.933	0.969
				Weighted Avg	*0.934*	*0.933*	*0.933*	*0.933*	*0.921*

Table S13. ClinBioBert Seq2Seq BIO CNN replicate model performance using RelIV--TIMEX evaluation data set. Phrase Only Hyperparameters: dropout: 0.05, kernel size1: 5, kernel size2: 3, num filters: 32, pool size: 3, stride: 2; +Context Hyperparameters: dropout: 0.05, kernel size1: 5, kernel size2: 3, num filters: 32, pool size: 2, stride: 2; +Attention Hyperparameters: dropout: 0.1, kernel size1: 3, kernel size2: 2, num filters: 64, pool size: 3, stride: 2; Metric abbreviations: P:Precision, R:Recall, Acc:Accuracy, Spe:Specificity.



ClinBioBert Binary Seq2Seq Ttype: CNN Replicate 0

	4x4	Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	412	40	0.941	0.912	0.926	0.916	0.922
	DURATION	26	309	0.885	0.922	0.904	0.916	0.912
	4x4	Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	427	25	0.844	0.945	0.891	0.868	0.764
	DURATION	79	256	0.911	0.764	0.831	0.868	0.945
	4x4	Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	394	58	0.889	0.872	0.880	0.864	0.854
	DURATION	49	286	0.831	0.854	0.842	0.864	0.872

ClinBioBert Binary Seq2Seq Ttype: CNN Replicate 4

	4x4	Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	425	27	0.910	0.940	0.925	0.912	0.875
	DURATION	42	293	0.916	0.875	0.895	0.912	0.940
	4x4	Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	404	48	0.918	0.894	0.906	0.893	0.893
	DURATION	36	299	0.862	0.893	0.877	0.893	0.894
	4x4	Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	378	74	0.906	0.836	0.870	0.856	0.884
	DURATION	39	296	0.800	0.884	0.840	0.856	0.836

ClinBioBert Binary Seq2Seq Ttype: CNN Replicate 1

	4x4	Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	423	29	0.920	0.936	0.928	0.916	0.890
	DURATION	37	298	0.911	0.890	0.900	0.916	0.936
	4x4	Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	413	39	0.902	0.914	0.908	0.893	0.866
	DURATION	45	290	0.881	0.866	0.873	0.893	0.914
	4x4	Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	399	53	0.899	0.883	0.891	0.875	0.866
	DURATION	45	290	0.845	0.866	0.855	0.875	0.883

ClinBioBert Binary Seq2Seq Ttype: CNN Replicate 3

	4x4	Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	423	29	0.932	0.936	0.934	0.924	0.907
	DURATION	31	304	0.913	0.907	0.910	0.924	0.936
	4x4	Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	431	21	0.857	0.954	0.903	0.882	0.785
	DURATION	72	263	0.926	0.785	0.850	0.882	0.954
	4x4	Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	403	49	0.886	0.892	0.889	0.872	0.845
	DURATION	52	283	0.852	0.845	0.849	0.872	0.892

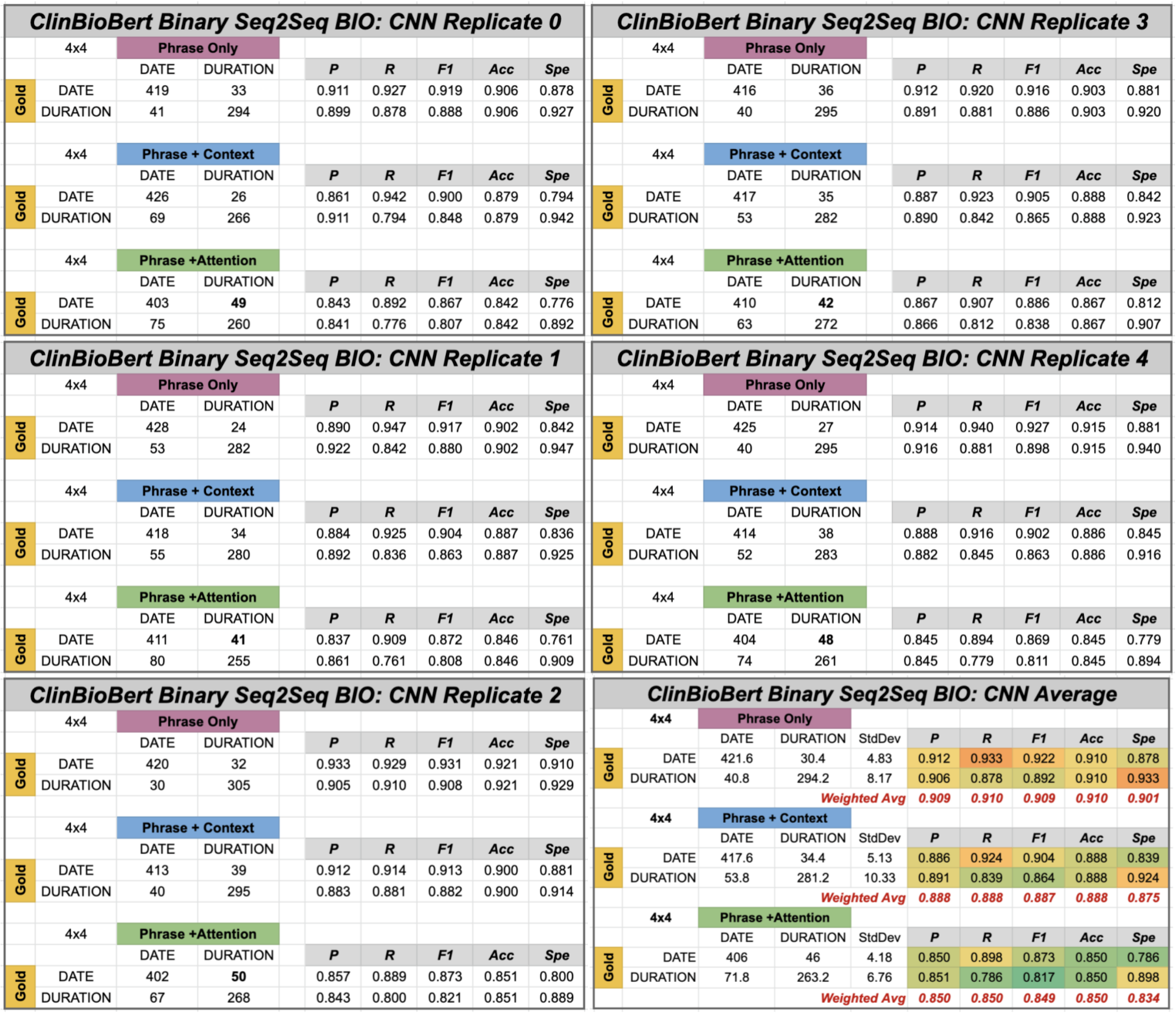
ClinBioBert Binary Seq2Seq Ttype: CNN Replicate 2

	4x4	Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	408	44	0.951	0.903	0.926	0.917	0.937
	DURATION	21	314	0.877	0.937	0.906	0.917	0.903
	4x4	Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	425	27	0.867	0.940	0.902	0.883	0.806
	DURATION	65	270	0.909	0.806	0.854	0.883	0.940
	4x4	Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	391	61	0.901	0.865	0.883	0.868	0.872
	DURATION	43	292	0.827	0.872	0.849	0.868	0.865

ClinBioBert Binary Seq2Seq Ttype: CNN Average

	4x4	Phrase Only							
		DATE	DURATION	StdDev	P	R	F1	Acc	Spe
Gold	DATE	418.2	33.8	7.66	0.930	0.925	0.928	0.917	0.906
	DURATION	31.4	303.6	8.38	0.900	0.906	0.903	0.917	0.925
				Weighted Avg	0.917	0.917	0.917	0.917	0.914
	4x4	Phrase + Context							
		DATE	DURATION	StdDev	P	R	F1	Acc	Spe
Gold	DATE	420	32	11.18	0.876	0.929	0.902	0.884	0.823
	DURATION	59.4	275.6	18.23	0.896	0.823	0.858	0.884	0.929
				Weighted Avg	0.885	0.884	0.883	0.884	0.868
	4x4	Phrase +Attention							
		DATE	DURATION	StdDev	P	R	F1	Acc	Spe
Gold	DATE	393	59	9.57	0.896	0.869	0.883	0.867	0.864
	DURATION	45.6	289.4	5.08	0.831	0.864	0.847	0.867	0.869
				Weighted Avg	0.868	0.867	0.867	0.867	0.866

Table S14. ClinBioBert Binary Seq2Seq Ttype CNN replicate model performance using RelIV-TIMEX evaluation data set. Phrase Only Hyperparameters: dropout: 0.05, kernel size1: 5, kernel size2: 3, num filters: 32, pool size: 2, stride: 2; +Context Hyperparameters: dropout: 0.1, kernel size1: 5, kernel size2: 3, num filters: 64, pool size: 3, stride: 2; +Attention Hyperparameters: dropout: 0.05, kernel size1: 3, kernel size2: 3, num filters: 64, pool size: 2, stride: 1; Metric abbreviations: P:Precision, R:Recall, Acc:Accuracy, Spe:Specificity.



ClinBioBert Binary Seq2Seq BIO: CNN Replicate 0

	4x4	Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	419	33	0.911	0.927	0.919	0.906	0.878
	DURATION	41	294	0.899	0.878	0.888	0.906	0.927
	4x4	Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	426	26	0.861	0.942	0.900	0.879	0.794
	DURATION	69	266	0.911	0.794	0.848	0.879	0.942
	4x4	Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	403	**49**	0.843	0.892	0.867	0.842	0.776
	DURATION	75	260	0.841	0.776	0.807	0.842	0.892

ClinBioBert Binary Seq2Seq BIO: CNN Replicate 3

	4x4	Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	416	36	0.912	0.920	0.916	0.903	0.881
	DURATION	40	295	0.891	0.881	0.886	0.903	0.920
	4x4	Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	417	35	0.887	0.923	0.905	0.888	0.842
	DURATION	53	282	0.890	0.842	0.865	0.888	0.923
	4x4	Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	410	**42**	0.867	0.907	0.886	0.867	0.812
	DURATION	63	272	0.866	0.812	0.838	0.867	0.907

ClinBioBert Binary Seq2Seq BIO: CNN Replicate 1

	4x4	Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	428	24	0.890	0.947	0.917	0.902	0.842
	DURATION	53	282	0.922	0.842	0.880	0.902	0.947
	4x4	Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	418	34	0.884	0.925	0.904	0.887	0.836
	DURATION	55	280	0.892	0.836	0.863	0.887	0.925
	4x4	Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	411	**41**	0.837	0.909	0.872	0.846	0.761
	DURATION	80	255	0.861	0.761	0.808	0.846	0.909

ClinBioBert Binary Seq2Seq BIO: CNN Replicate 4

	4x4	Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	425	27	0.914	0.940	0.927	0.915	0.881
	DURATION	40	295	0.916	0.881	0.898	0.915	0.940
	4x4	Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	414	38	0.888	0.916	0.902	0.886	0.845
	DURATION	52	283	0.882	0.845	0.863	0.886	0.916
	4x4	Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	404	**48**	0.845	0.894	0.869	0.845	0.779
	DURATION	74	261	0.845	0.779	0.811	0.845	0.894

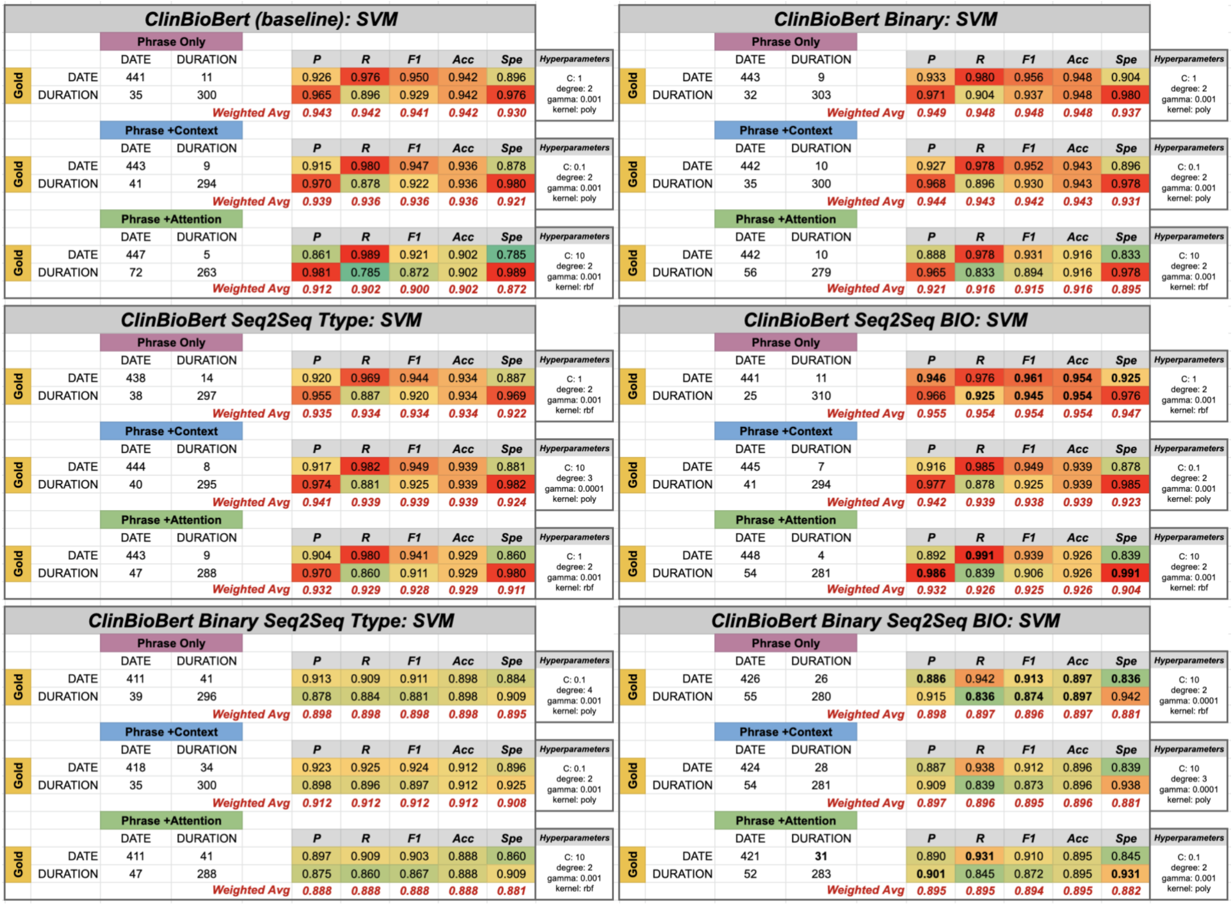
ClinBioBert Binary Seq2Seq BIO: CNN Replicate 2

	4x4	Phrase Only						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	420	32	0.933	0.929	0.931	0.921	0.910
	DURATION	30	305	0.905	0.910	0.908	0.921	0.929
	4x4	Phrase + Context						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	413	39	0.912	0.914	0.913	0.900	0.881
	DURATION	40	295	0.883	0.881	0.882	0.900	0.914
	4x4	Phrase +Attention						
		DATE	DURATION	P	R	F1	Acc	Spe
Gold	DATE	402	**50**	0.857	0.889	0.873	0.851	0.800
	DURATION	67	268	0.843	0.800	0.821	0.851	0.889

ClinBioBert Binary Seq2Seq BIO: CNN Average

	4x4	Phrase Only							
		DATE	DURATION	StdDev	P	R	F1	Acc	Spe
Gold	DATE	421.6	30.4	4.83	0.912	0.933	0.922	0.910	0.878
	DURATION	40.8	294.2	8.17	0.906	0.878	0.892	0.910	0.933
				Weighted Avg	0.909	0.910	0.909	0.910	0.901
	4x4	Phrase + Context							
		DATE	DURATION	StdDev	P	R	F1	Acc	Spe
Gold	DATE	417.6	34.4	5.13	0.886	0.924	0.904	0.888	0.839
	DURATION	53.8	281.2	10.33	0.891	0.839	0.864	0.888	0.924
				Weighted Avg	0.888	0.888	0.887	0.888	0.875
	4x4	Phrase +Attention							
		DATE	DURATION	StdDev	P	R	F1	Acc	Spe
Gold	DATE	406	46	4.18	0.850	0.898	0.873	0.850	0.786
	DURATION	71.8	263.2	6.76	0.851	0.786	0.817	0.850	0.898
				Weighted Avg	0.850	0.850	0.849	0.850	0.834

Table S15. ClinBioBert Binary Seq2Seq BIO CNN replicate model performance using RelIV-TIMEX evaluation data set. Phrase Only Hyperparameters: dropout: 0.05, kernel size1: 5, kernel size2: 3, num filters: 64, pool size: 3, stride: 1; +Context Hyperparameters: dropout: 0.05, kernel size1: 3, kernel size2: 2, num filters: 32, pool size: 2, stride: 1; +Attention Hyperparameters: dropout: 0.1, kernel size1: 5, kernel size2: 2, num filters: 64, pool size: 2, stride: 1; Metric abbreviations: P:Precision, R:Recall, Acc:Accuracy, Spe:Specificity.



ClinBioBert (baseline): SVM

		DATE	DURATION	P	R	F1	Acc	Spe	Hyperparameters
		Phrase Only							
Gold	DATE	441	11	0.926	0.976	0.950	0.942	0.896	C: 1 degree: 2 gamma: 0.001 kernel: poly
	DURATION	35	300	0.965	0.896	0.929	0.942	0.976	
			Weighted Avg	0.943	0.942	0.941	0.942	0.930	
		Phrase +Context							
Gold	DATE	443	9	0.915	0.980	0.947	0.936	0.878	C: 0.1 degree: 2 gamma: 0.001 kernel: poly
	DURATION	41	294	0.970	0.878	0.922	0.936	0.980	
			Weighted Avg	0.939	0.936	0.936	0.936	0.921	
		Phrase +Attention							
Gold	DATE	447	5	0.861	0.989	0.921	0.902	0.785	C: 10 degree: 2 gamma: 0.001 kernel: rbf
	DURATION	72	263	0.981	0.785	0.872	0.902	0.989	
			Weighted Avg	0.912	0.902	0.900	0.902	0.872	

ClinBioBert Binary: SVM

		DATE	DURATION	P	R	F1	Acc	Spe	Hyperparameters
		Phrase Only							
Gold	DATE	443	9	0.933	0.980	0.956	0.948	0.904	C: 1 degree: 2 gamma: 0.001 kernel: poly
	DURATION	32	303	0.971	0.904	0.937	0.948	0.980	
			Weighted Avg	0.949	0.948	0.948	0.948	0.937	
		Phrase +Context							
Gold	DATE	442	10	0.927	0.978	0.952	0.943	0.896	C: 0.1 degree: 2 gamma: 0.001 kernel: poly
	DURATION	35	300	0.968	0.896	0.930	0.943	0.978	
			Weighted Avg	0.944	0.943	0.942	0.943	0.931	
		Phrase +Attention							
Gold	DATE	442	10	0.888	0.978	0.931	0.916	0.833	C: 10 degree: 2 gamma: 0.001 kernel: rbf
	DURATION	56	279	0.965	0.833	0.894	0.916	0.978	
			Weighted Avg	0.921	0.916	0.915	0.916	0.895	

ClinBioBert Seq2Seq Ttype: SVM

		DATE	DURATION	P	R	F1	Acc	Spe	Hyperparameters
		Phrase Only							
Gold	DATE	438	14	0.920	0.969	0.944	0.934	0.887	C: 1 degree: 2 gamma: 0.001 kernel: rbf
	DURATION	38	297	0.955	0.887	0.920	0.934	0.969	
			Weighted Avg	0.935	0.934	0.934	0.934	0.922	
		Phrase +Context							
Gold	DATE	444	8	0.917	0.982	0.949	0.939	0.881	C: 10 degree: 3 gamma: 0.0001 kernel: poly
	DURATION	40	295	0.974	0.881	0.925	0.939	0.982	
			Weighted Avg	0.941	0.939	0.939	0.939	0.924	
		Phrase +Attention							
Gold	DATE	443	9	0.904	0.980	0.941	0.929	0.860	C: 1 degree: 2 gamma: 0.001 kernel: rbf
	DURATION	47	288	0.970	0.860	0.911	0.929	0.980	
			Weighted Avg	0.932	0.929	0.928	0.929	0.911	

ClinBioBert Seq2Seq BIO: SVM

		DATE	DURATION	P	R	F1	Acc	Spe	Hyperparameters
		Phrase Only							
Gold	DATE	441	11	**0.946**	0.976	**0.961**	**0.954**	**0.925**	C: 1 degree: 2 gamma: 0.001 kernel: rbf
	DURATION	25	310	0.966	**0.925**	**0.945**	**0.954**	0.976	
			Weighted Avg	0.955	0.954	0.954	0.954	0.947	
		Phrase +Context							
Gold	DATE	445	7	0.916	0.985	0.949	0.939	0.878	C: 0.1 degree: 2 gamma: 0.001 kernel: poly
	DURATION	41	294	0.977	0.878	0.925	0.939	0.985	
			Weighted Avg	0.942	0.939	0.938	0.939	0.923	
		Phrase +Attention							
Gold	DATE	448	4	0.892	**0.991**	0.939	0.926	0.839	C: 10 degree: 2 gamma: 0.001 kernel: rbf
	DURATION	54	281	**0.986**	0.839	0.906	0.926	**0.991**	
			Weighted Avg	0.932	0.926	0.925	0.926	0.904	

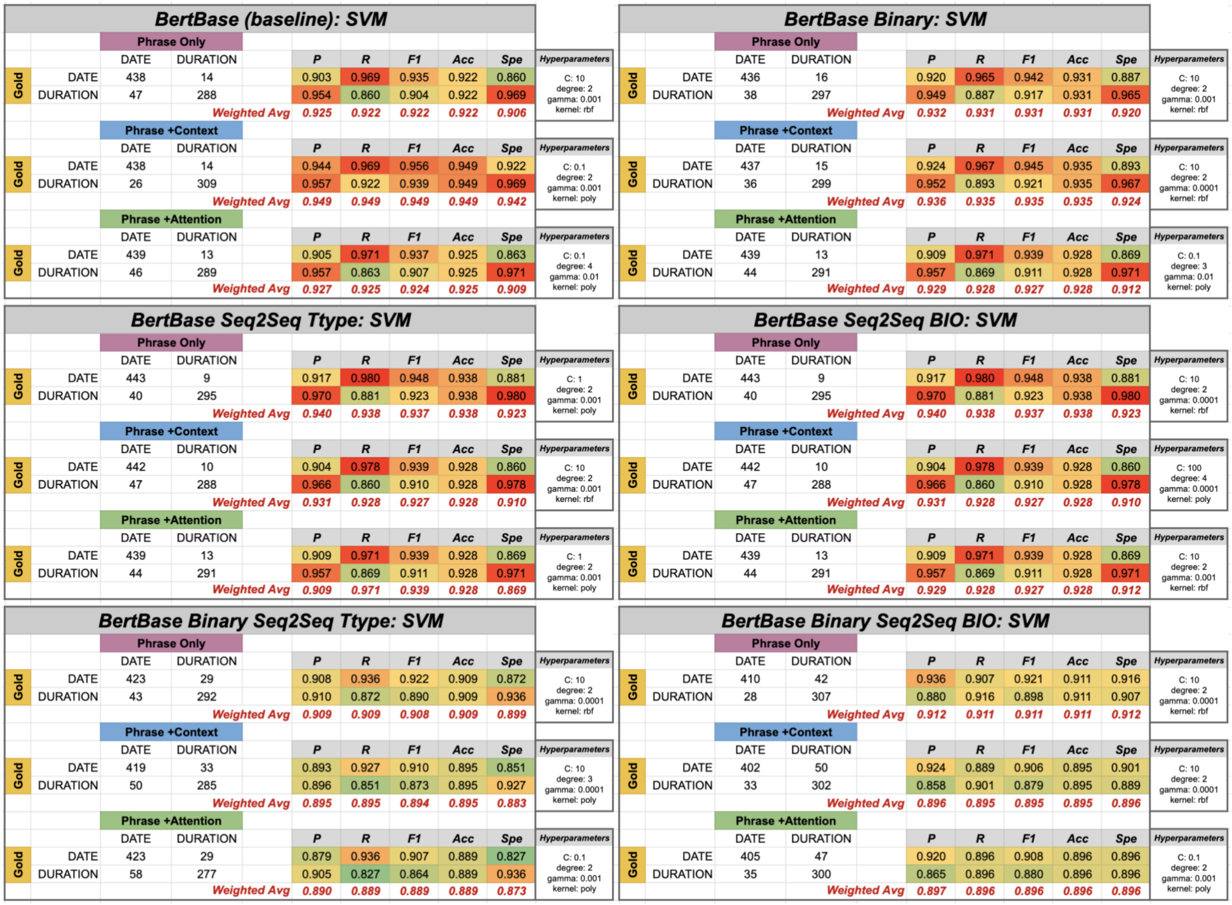
ClinBioBert Binary Seq2Seq Ttype: SVM

		DATE	DURATION	P	R	F1	Acc	Spe	Hyperparameters
		Phrase Only							
Gold	DATE	411	41	0.913	0.909	0.911	0.898	0.884	C: 0.1 degree: 4 gamma: 0.001 kernel: poly
	DURATION	39	296	0.878	0.884	0.881	0.898	0.909	
			Weighted Avg	0.898	0.898	0.898	0.898	0.895	
		Phrase +Context							
Gold	DATE	418	34	0.923	0.925	0.924	0.912	0.896	C: 0.1 degree: 2 gamma: 0.001 kernel: poly
	DURATION	35	300	0.898	0.896	0.897	0.912	0.925	
			Weighted Avg	0.912	0.912	0.912	0.912	0.908	
		Phrase +Attention							
Gold	DATE	411	41	0.897	0.909	0.903	0.888	0.860	C: 10 degree: 2 gamma: 0.001 kernel: rbf
	DURATION	47	288	0.875	0.860	0.867	0.888	0.909	
			Weighted Avg	0.888	0.888	0.888	0.888	0.881	

ClinBioBert Binary Seq2Seq BIO: SVM

		DATE	DURATION	P	R	F1	Acc	Spe	Hyperparameters
		Phrase Only							
Gold	DATE	426	26	**0.886**	0.942	**0.913**	**0.897**	**0.836**	C: 10 degree: 2 gamma: 0.0001 kernel: rbf
	DURATION	55	280	0.915	**0.836**	**0.874**	**0.897**	0.942	
			Weighted Avg	0.898	0.897	0.896	0.897	0.881	
		Phrase +Context							
Gold	DATE	424	28	0.887	0.938	0.912	0.896	0.839	C: 10 degree: 3 gamma: 0.0001 kernel: poly
	DURATION	54	281	0.909	0.839	0.873	0.896	0.938	
			Weighted Avg	0.897	0.896	0.895	0.896	0.881	
		Phrase +Attention							
Gold	DATE	421	**31**	0.890	**0.931**	0.910	0.895	0.845	C: 0.1 degree: 2 gamma: 0.001 kernel: poly
	DURATION	52	283	**0.901**	0.845	0.872	0.895	**0.931**	
			Weighted Avg	0.895	0.895	0.894	0.895	0.882	

Table S16. ClinBioBert SVM model performance using RelIV-TIMEX evaluation data set. Metric abbreviations: P:Precision, R:Recall, Acc:Accuracy, Spe:Specificity.



BertBase (baseline): SVM

Phrase Only

		DATE	DURATION		P	R	F1	Acc	Spe	Hyperparameters
Gold	DATE	438	14		0.903	0.969	0.935	0.922	0.860	C: 10 degree: 2 gamma: 0.001 kernel: rbf
	DURATION	47	288		0.954	0.860	0.904	0.922	0.969	
				Weighted Avg	0.925	0.922	0.922	0.922	0.906	

Phrase +Context

		DATE	DURATION		P	R	F1	Acc	Spe	Hyperparameters
Gold	DATE	438	14		0.944	0.969	0.956	0.949	0.922	C: 0.1 degree: 2 gamma: 0.001 kernel: poly
	DURATION	26	309		0.957	0.922	0.939	0.949	0.969	
				Weighted Avg	0.949	0.949	0.949	0.949	0.942	

Phrase +Attention

		DATE	DURATION		P	R	F1	Acc	Spe	Hyperparameters
Gold	DATE	439	13		0.905	0.971	0.937	0.925	0.863	C: 0.1 degree: 4 gamma: 0.01 kernel: poly
	DURATION	46	289		0.957	0.863	0.907	0.925	0.971	
				Weighted Avg	0.927	0.925	0.924	0.925	0.909	

BertBase Binary: SVM

Phrase Only

		DATE	DURATION		P	R	F1	Acc	Spe	Hyperparameters
Gold	DATE	436	16		0.920	0.965	0.942	0.931	0.887	C: 10 degree: 2 gamma: 0.001 kernel: rbf
	DURATION	38	297		0.949	0.887	0.917	0.931	0.965	
				Weighted Avg	0.932	0.931	0.931	0.931	0.920	

Phrase +Context

		DATE	DURATION		P	R	F1	Acc	Spe	Hyperparameters
Gold	DATE	437	15		0.924	0.967	0.945	0.935	0.893	C: 10 degree: 2 gamma: 0.0001 kernel: rbf
	DURATION	36	299		0.952	0.893	0.921	0.935	0.967	
				Weighted Avg	0.936	0.935	0.935	0.935	0.924	

Phrase +Attention

		DATE	DURATION		P	R	F1	Acc	Spe	Hyperparameters
Gold	DATE	439	13		0.909	0.971	0.939	0.928	0.869	C: 0.1 degree: 3 gamma: 0.01 kernel: poly
	DURATION	44	291		0.957	0.869	0.911	0.928	0.971	
				Weighted Avg	0.929	0.928	0.927	0.928	0.912	

BertBase Seq2Seq Ttype: SVM

Phrase Only

		DATE	DURATION		P	R	F1	Acc	Spe	Hyperparameters
Gold	DATE	443	9		0.917	0.980	0.948	0.938	0.881	C: 1 degree: 2 gamma: 0.001 kernel: poly
	DURATION	40	295		0.970	0.881	0.923	0.938	0.980	
				Weighted Avg	0.940	0.938	0.937	0.938	0.923	

Phrase +Context

		DATE	DURATION		P	R	F1	Acc	Spe	Hyperparameters
Gold	DATE	442	10		0.904	0.978	0.939	0.928	0.860	C: 10 degree: 2 gamma: 0.001 kernel: rbf
	DURATION	47	288		0.966	0.860	0.910	0.928	0.978	
				Weighted Avg	0.931	0.928	0.927	0.928	0.910	

Phrase +Attention

		DATE	DURATION		P	R	F1	Acc	Spe	Hyperparameters
Gold	DATE	439	13		0.909	0.971	0.939	0.928	0.869	C: 1 degree: 2 gamma: 0.001 kernel: poly
	DURATION	44	291		0.957	0.869	0.911	0.928	0.971	
				Weighted Avg	0.909	0.971	0.939	0.928	0.869	

BertBase Seq2Seq BIO: SVM

Phrase Only

		DATE	DURATION		P	R	F1	Acc	Spe	Hyperparameters
Gold	DATE	443	9		0.917	0.980	0.948	0.938	0.881	C: 10 degree: 2 gamma: 0.0001 kernel: rbf
	DURATION	40	295		0.970	0.881	0.923	0.938	0.980	
				Weighted Avg	0.940	0.938	0.937	0.938	0.923	

Phrase +Context

		DATE	DURATION		P	R	F1	Acc	Spe	Hyperparameters
Gold	DATE	442	10		0.904	0.978	0.939	0.928	0.860	C: 100 degree: 4 gamma: 0.0001 kernel: poly
	DURATION	47	288		0.966	0.860	0.910	0.928	0.978	
				Weighted Avg	0.931	0.928	0.927	0.928	0.910	

Phrase +Attention

		DATE	DURATION		P	R	F1	Acc	Spe	Hyperparameters
Gold	DATE	439	13		0.909	0.971	0.939	0.928	0.869	C: 10 degree: 2 gamma: 0.001 kernel: rbf
	DURATION	44	291		0.957	0.869	0.911	0.928	0.971	
				Weighted Avg	0.929	0.928	0.927	0.928	0.912	

BertBase Binary Seq2Seq Ttype: SVM

Phrase Only

		DATE	DURATION		P	R	F1	Acc	Spe	Hyperparameters
Gold	DATE	423	29		0.908	0.936	0.922	0.909	0.872	C: 10 degree: 2 gamma: 0.0001 kernel: rbf
	DURATION	43	292		0.910	0.872	0.890	0.909	0.936	
				Weighted Avg	0.909	0.909	0.908	0.909	0.899	

Phrase +Context

		DATE	DURATION		P	R	F1	Acc	Spe	Hyperparameters
Gold	DATE	419	33		0.893	0.927	0.910	0.895	0.851	C: 10 degree: 3 gamma: 0.0001 kernel: poly
	DURATION	50	285		0.896	0.851	0.873	0.895	0.927	
				Weighted Avg	0.895	0.895	0.894	0.895	0.883	

Phrase +Attention

		DATE	DURATION		P	R	F1	Acc	Spe	Hyperparameters
Gold	DATE	423	29		0.879	0.936	0.907	0.889	0.827	C: 0.1 degree: 2 gamma: 0.001 kernel: poly
	DURATION	58	277		0.905	0.827	0.864	0.889	0.936	
				Weighted Avg	0.890	0.889	0.889	0.889	0.873	

BertBase Binary Seq2Seq BIO: SVM

Phrase Only

		DATE	DURATION		P	R	F1	Acc	Spe	Hyperparameters
Gold	DATE	410	42		0.936	0.907	0.921	0.911	0.916	C: 10 degree: 2 gamma: 0.0001 kernel: rbf
	DURATION	28	307		0.880	0.916	0.898	0.911	0.907	
				Weighted Avg	0.912	0.911	0.911	0.911	0.912	

Phrase +Context

		DATE	DURATION		P	R	F1	Acc	Spe	Hyperparameters
Gold	DATE	402	50		0.924	0.889	0.906	0.895	0.901	C: 10 degree: 2 gamma: 0.0001 kernel: rbf
	DURATION	33	302		0.858	0.901	0.879	0.895	0.889	
				Weighted Avg	0.896	0.895	0.895	0.895	0.896	

Phrase +Attention

		DATE	DURATION		P	R	F1	Acc	Spe	Hyperparameters
Gold	DATE	405	47		0.920	0.896	0.908	0.896	0.896	C: 0.1 degree: 2 gamma: 0.001 kernel: poly
	DURATION	35	300		0.865	0.896	0.880	0.896	0.896	
				Weighted Avg	0.897	0.896	0.896	0.896	0.896	

Table S17. BertBase SVM model performance using RelIV-TIMEX evaluation data set. Metric abbreviations: P:Precision, R:Recall, Acc:Accuracy, Spe:Specificity.

Bibliography

[1] Claire McCarthy MD. *Does your child need a tonsillectomy?* en-US. Mar. 2018. URL: https://www.health.harvard.edu/blog/child-need-tonsillectomy-2018032013504 (visited on 10/04/2020).

[2] Daniel Reichert et al. "Cognitive Analysis of the Summarization of Longitudinal Patient Records". In: *AMIA Annual Symposium Proceedings* 2010 (2010), pp. 667–671. ISSN: 1942-597X. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3041351/ (visited on 05/16/2020).

[3] Andres Ledesma et al. "Health timeline: an insight-based study of a timeline visualization of clinical data". en. In: *BMC Medical Informatics and Decision Making* 19.1 (Aug. 2019), p. 170. ISSN: 1472-6947. DOI: 10.1186/s12911-019-0885-x. URL: https://doi.org/10.1186/s12911-019-0885-x (visited on 09/06/2020).

[4] Artuur Leeuwenberg and Marie-Francine Moens. "A Survey on Temporal Reasoning for Temporal Information Extraction from Text". en. In: *Journal of Artificial Intelligence Research* 66 (Sept. 2019), pp. 341–380. ISSN: 1076-9757. DOI: 10.1613/jair.1.11727. URL: https://www.jair.org/index.php/jair/article/view/11727 (visited on 07/05/2020).

[5] Naman Gupta. "Temporal Information Extraction Extracting Events and Temporal Expressions A Literature Survey". en. In: (2015), p. 34. URL: http://www.cfilt.iitb.ac.in/resources/surveys/Temporal_Information_Extraction-Naman-June15.pdf.

[6] A.K. Pani and G.P. Bhattacharjee. “Temporal representation and reasoning in artificial intelligence: A review”. en. In: *Mathematical and Computer Modelling* 34.1-2 (July 2001), pp. 55–80. ISSN: 08957177. DOI: 10.1016/S0895-7177(01)00049-8. URL: https://linkinghub.elsevier.com/retrieve/pii/S0895717701000498 (visited on 08/22/2020).

[7] Chae-Gyun Lim, Young-Seob Jeong, and Ho-Jin Choi. “Survey of Temporal Information Extraction”. en. In: *Journal of Information Processing Systems* 15.4 (Aug. 2019), pp. 931–956. DOI: 10.3745/JIPS.04.0129. URL: https://doi.org/10.3745/JIPS.04.0129 (visited on 08/22/2020).

[8] Amy L. Olex and Bridget T. McInnes. “Review of Temporal Reasoning in the Clinical Domain for Timeline Extraction: Where we are and where we need to be”. en. In: *Journal of Biomedical Informatics* 118 (June 2021), p. 103784. ISSN: 1532-0464. DOI: 10.1016/j.jbi.2021.103784. URL: https://www.sciencedirect.com/science/article/pii/S1532046421001131 (visited on 05/25/2021).

[9] Li Zhou and George Hripcsak. “Temporal reasoning with medical data—A review with emphasis on medical natural language processing”. In: *Journal of Biomedical Informatics* 40.2 (Apr. 2007), pp. 183–202. ISSN: 1532-0464. DOI: 10.1016/j.jbi.2006.12.009. URL: http://www.sciencedirect.com/science/article/pii/S1532046407000032 (visited on 01/04/2019).

[10] Weiyi Sun, Anna Rumshisky, and Ozlem Uzuner. “Temporal reasoning over clinical text: the state of the art”. en. In: *Journal of the American Medical Informatics Association* 20.5 (Sept. 2013), pp. 814–819. ISSN: 1067-5027, 1527-974X. DOI: 10.1136/amiajnl-2013-001760. URL: https://academic.oup.

com/jamia/article-lookup/doi/10.1136/amiajnl-2013-001760 (visited on 11/12/2018).

[11] Steven Bethard et al. “SemEval-2015 Task 6: Clinical TempEval”. In: *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Denver, Colorado: Association for Computational Linguistics, June 2015, pp. 806–814. URL: http://www.aclweb.org/anthology/S15-2136 (visited on 11/13/2018).

[12] William F. Styler Iv et al. “Temporal Annotation in the Clinical Domain”. en. In: *Transactions of the Association for Computational Linguistics* 2.0 (Apr. 2014). Number: 0, pp. 143–154. ISSN: 2307-387X. (Visited on 05/01/2020).

[13] Jannik Strötgen and Michael Gertz. “HeidelTime: High Quality Rule-based Extraction and Normalization of Temporal Expressions”. In: *Proceedings of the 5th International Workshop on Semantic Evaluation*. SemEval ’10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 321–324.

[14] Angel X. Chang and Christopher D. Manning. “Sutime: A library for recognizing and normalizing time expressions.” In: *Lrec*. Vol. 2012. 2012, pp. 3735–3740.

[15] Weiyi Sun, Anna Rumshisky, and Ozlem Uzuner. “Normalization of relative and incomplete temporal expressions in clinical narratives”. en. In: *Journal of the American Medical Informatics Association* 22.5 (Sept. 2015), pp. 1001–1008. ISSN: 1067-5027. DOI: 10.1093/jamia/ocu004. URL: http://academic.oup.com/jamia/article/22/5/1001/928475 (visited on 11/12/2018).

[16] Lisa Ferro et al. “TIDES Temporal Annotation Guidelines Version 1.0.2”. en. In: (), p. 57.

[17] James Pustejovsky et al. "TimeML: Robust specification of event and temporal expressions in text." In: *New directions in question answering* 3 (2003), pp. 28–34.

[18] Andrea Setzer. "Temporal information in newswire articles : an annotation scheme and corpus study." phd. University of Sheffield, 2002. URL: `http://etheses.whiterose.ac.uk/14436/` (visited on 08/29/2020).

[19] Estela Saquete and James Pustejovsky. "Automatic transformation from TIDES to TimeML annotation". en. In: *Language Resources and Evaluation* 45.4 (Dec. 2011), pp. 495–523. ISSN: 1574-0218. DOI: `10.1007/s10579-011-9147-y`. URL: `https://doi.org/10.1007/s10579-011-9147-y` (visited on 08/29/2020).

[20] James Pustejovsky et al. "ISO-TimeML: An International Standard for Semantic Annotation". en. In: *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. Ed. by Nicoletta Calzolari. Valletta, Malta, 2010, p. 4.

[21] Weiyi Sun, Anna Rumshisky, and Ozlem Uzuner. "Evaluating temporal relations in clinical text: 2012 i2b2 Challenge." eng. In: *Journal of the American Medical Informatics Association : JAMIA* 20.5 (Sept. 2013), pp. 806–813. ISSN: 1067-5027. DOI: `10.1136/amiajnl-2013-001628`. URL: `https://www.ncbi.nlm.nih.govhttp://europepmc.org/articles/PMC3756273/` (visited on 11/12/2018).

[22] Steven Bethard and Jonathan Parker. "A Semantically Compositional Annotation Scheme for Time Normalization." In: *Lrec*. Vol. 2016. 2016, pp. 3779–3786.

[23] Ralph Grishman and Beth Sundheim. “Design of the MUC-6 Evaluation”. In: *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*. 1995. URL: `https://www.aclweb.org/anthology/M95-1001` (visited on 08/22/2020).

[24] Boyan Onyshkevych, Mary Ellen Okurowski, and Lynn Carlson. “Tasks, Domains, and Languages”. In: *Fifth Message Understanding Conference (MUC-5): Proceedings of a Conference Held in Baltimore, Maryland, August 25-27, 1993*. 1993. URL: `https://www.aclweb.org/anthology/M93-1002` (visited on 10/05/2020).

[25] Nancy A. Chinchor. “Overview of MUC-7”. In: *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998*. 1998. URL: `https://www.aclweb.org/anthology/M98-1001` (visited on 10/05/2020).

[26] Matteo Negri and Luca Marseglia. “Recognition and Normalization of Time Expressions: ITC-irst at TERN 2004”. In: (Jan. 2004).

[27] Marc Verhagen et al. “SemEval-2007 Task 15: TempEval Temporal Relation Identification”. In: *Proceedings of the 4th International Workshop on Semantic Evaluations*. SemEval ’07. Stroudsburg, PA, USA: Association for Computational Linguistics, 2007, pp. 75–80. URL: `http://dl.acm.org/citation.cfm?id=1621474.1621488` (visited on 01/13/2019).

[28] Marc Verhagen et al. “SemEval-2010 Task 13: TempEval-2”. In: *Proceedings of the 5th International Workshop on Semantic Evaluation*. SemEval ’10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 57–62. URL: `http://dl.acm.org/citation.cfm?id=1859664.1859674` (visited on 01/13/2019).

[29] Naushad UzZaman et al. “SemEval-2013 Task 1: TempEval-3: Evaluating Time Expressions, Events, and Temporal Relations”. In: *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*. Atlanta, Georgia, USA: Association for Computational Linguistics, June 2013, pp. 1–9. URL: http://www.aclweb.org/anthology/S13-2001 (visited on 01/13/2019).

[30] Yao Cheng et al. “Temporal relation discovery between events and temporal expressions identified in clinical narrative”. In: *Journal of Biomedical Informatics*. 2012 i2b2 NLP Challenge on Temporal Relations in Clinical Data 46 (Dec. 2013), S48–S53. ISSN: 1532-0464. DOI: 10.1016/j.jbi.2013.09.010. URL: http://www.sciencedirect.com/science/article/pii/S1532046413001512 (visited on 11/12/2018).

[31] Jennifer D’Souza and Vincent Ng. “Knowledge-rich temporal relation identification and classification in clinical notes”. en. In: *Database* 2014 (Jan. 2014). Publisher: Oxford Academic. DOI: 10.1093/database/bau109. URL: https://academic.oup.com/database/article/doi/10.1093/database/bau109/2635427 (visited on 06/20/2020).

[32] Jennifer D’Souza and Vincent Ng. “Annotating Inter-Sentence Temporal Relations in Clinical Notes”. en. In: (2014), p. 8.

[33] Wei Wang et al. “A new algorithmic approach for the extraction of temporal associations from clinical narratives with an application to medical product safety surveillance reports”. en. In: *Journal of Biomedical Informatics* 62 (Aug. 2016), pp. 78–89. ISSN: 1532-0464. DOI: 10.1016/j.jbi.2016.

06.006. URL: http://www.sciencedirect.com/science/article/pii/S1532046416300491 (visited on 07/11/2020).

[34] Hee-Jin Lee et al. "Towards practical temporal relation extraction from clinical notes: An analysis of direct temporal relations". In: *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. Nov. 2017, pp. 1272–1275. DOI: 10.1109/BIBM.2017.8217842.

[35] Hee-Jin Lee et al. "Identifying direct temporal relations between time and events from clinical notes". en. In: *BMC Medical Informatics and Decision Making* 18.2 (July 2018), p. 49. ISSN: 1472-6947. DOI: 10.1186/s12911-018-0627-5. URL: https://doi.org/10.1186/s12911-018-0627-5 (visited on 05/23/2020).

[36] Hong Guan et al. "Robustly Pre-trained Neural Model for Direct Temporal Relation Extraction". In: *arXiv:2004.06216 [cs]* (Apr. 2020). arXiv: 2004.06216. URL: http://arxiv.org/abs/2004.06216 (visited on 05/20/2020).

[37] Chen Lin et al. "Multilayered temporal modeling for the clinical domain". en. In: *Journal of the American Medical Informatics Association* 23.2 (Mar. 2016). Publisher: Oxford Academic, pp. 387–395. ISSN: 1067-5027. DOI: 10.1093/jamia/ocv113. URL: https://academic.oup.com/jamia/article/23/2/387/2572466 (visited on 05/09/2020).

[38] Ruchi Patel and Sanjay Tanwani. "TEMPORAL RELATION IDENTIFICATION FROM CLINICAL TEXT USING LSTM BASED DEEP LEARNING MODEL". en. In: 5.4 (2018), p. 5.

[39] Steven Bethard et al. "SemEval-2016 Task 12: Clinical TempEval". In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. San Diego, California: Association for Computational Linguistics, June

2016, pp. 1052–1062. URL: http://www.aclweb.org/anthology/S16-1165 (visited on 11/13/2018).

[40] Steven Bethard et al. "SemEval-2017 Task 12: Clinical TempEval". In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Vancouver, Canada: Association for Computational Linguistics, 2017, pp. 565–572. DOI: 10.18653/v1/S17-2093. URL: http://aclweb.org/anthology/S17-2093 (visited on 01/03/2019).

[41] Egoitz Laparra et al. "SemEval 2018 Task 6: Parsing Time Normalization." In: *Proceedings of the 12th International Workshop on Semantic Evaluation*. SemEval '18. New Orleans, LA, USA: Association for Computational Linguistics, 2018.

[42] Amy Olex et al. "Chrono at SemEval-2018 Task 6: A System for Normalizing Temporal Expressions". In: *Proceedings of The 12th International Workshop on Semantic Evaluation*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 97–101. URL: http://www.aclweb.org/anthology/S18-1012 (visited on 09/27/2018).

[43] Steven Bethard. "ClearTK-TimeML: A minimalist approach to TempEval 2013". In: *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*. Atlanta, Georgia, USA: Association for Computational Linguistics, June 2013, pp. 10–14. URL: http://www.aclweb.org/anthology/S13-2002 (visited on 03/24/2019).

[44] Sunghwan Sohn et al. "Comprehensive temporal information detection from clinical text: medical events, time, and TLINK identification". en. In: *Journal of the American Medical Informatics Association* 20.5 (Sept. 2013), pp. 836–

842. ISSN: 1067-5027. DOI: 10.1136/amiajnl-2013-001622. URL: https://academic.oup.com/jamia/article/20/5/836/727595 (visited on 03/24/2019).

[45] Amy Olex, Luke Maffey, and Bridget McInnes. "NLP Whack-A-Mole: Challenges in Cross-Domain Temporal Expression Extraction". 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)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 3682–3692. DOI: 10.18653/v1/N19-1369. URL: https://www.aclweb.org/anthology/N19-1369 (visited on 11/22/2019).

[46] Rajdeep Sarkar, Bisal Nayal, and Aparna Joshi. "An Approach for Temporal Ordering of Medical Case Reports". en. In: *Data Management, Analytics and Innovation*. Ed. by Valentina Emilia Balas, Neha Sharma, and Amlan Chakrabarti. Advances in Intelligent Systems and Computing. Springer Singapore, 2019, pp. 131–141. ISBN: 9789811312748.

[47] Marjan Najafabadipour et al. "Analysis of Electronic Health Records to Identify the Patient's Treatment Lines: Challenges and Opportunities". en. In: *Artificial Intelligence XXXVI*. Ed. by Max Bramer and Miltos Petridis. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019, pp. 437–442. ISBN: 978-3-030-34885-4. DOI: 10.1007/978-3-030-34885-4_33.

[48] Marjan Najafabadipour et al. "Reconstructing the patient's natural history from electronic health records". en. In: *Artificial Intelligence in Medicine* 105 (May 2020), p. 101860. ISSN: 0933-3657. DOI: 10.1016/j.artmed.2020.

101860. URL: http://www.sciencedirect.com/science/article/pii/S0933365719311467 (visited on 06/19/2020).

[49] Azad Dehghan. *Mining Patient Journeys from Healthcare Narratives*. PhD. 2014. URL: https://www.research.manchester.ac.uk/portal/files/54570636/FULL_TEXT.PDF (visited on 11/12/2018).

[50] Azad Dehghan. "Temporal ordering of clinical events". In: *arXiv:1504.03659 [cs]* (Apr. 2015). arXiv: 1504.03659. URL: http://arxiv.org/abs/1504.03659 (visited on 11/13/2018).

[51] Julien Tourille. "Extracting Clinical Event Timelines : Temporal Information Extraction and Coreference Resolution in Electronic Health Records". en. PhD thesis. Université Paris-Saclay, Dec. 2018. URL: https://tel.archives-ouvertes.fr/tel-01997223 (visited on 07/11/2020).

[52] Hee-Jin Lee et al. "UTHealth at SemEval-2016 Task 12: an End-to-End System for Temporal Information Extraction from Clinical Notes". In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. San Diego, California: Association for Computational Linguistics, June 2016, pp. 1292–1297. DOI: 10.18653/v1/S16-1201. URL: https://www.aclweb.org/anthology/S16-1201 (visited on 07/04/2020).

[53] Özlem Uzuner et al. "2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text". en. In: *Journal of the American Medical Informatics Association* 18.5 (Sept. 2011), pp. 552–556. ISSN: 1067-5027. DOI: 10.1136/amiajnl-2011-000203. URL: http://academic.oup.com/jamia/article/18/5/552/830538 (visited on 01/13/2019).

[54] Jesse O Wrenn et al. "Quantifying clinical narrative redundancy in an electronic health record". In: *Journal of the American Medical Informatics Asso-*

ciation 17.1 (Jan. 2010), pp. 49–53. ISSN: 1067-5027. DOI: 10.1197/jamia.M3390. URL: https://doi.org/10.1197/jamia.M3390 (visited on 02/09/2021).

[55] Yinhan Liu et al. “RoBERTa: A Robustly Optimized BERT Pretraining Approach”. In: *arXiv:1907.11692 [cs]* (July 2019). arXiv: 1907.11692. URL: http://arxiv.org/abs/1907.11692 (visited on 11/24/2020).

[56] Jacob Devlin et al. “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)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://www.aclweb.org/anthology/N19-1423 (visited on 11/08/2019).

[57] Natalia Viani et al. “Annotating Temporal Relations to Determine the Onset of Psychosis Symptoms”. eng. In: *Studies in Health Technology and Informatics* 264 (Aug. 2019), pp. 418–422. ISSN: 1879-8365. DOI: 10.3233/SHTI190255.

[58] Buzhou Tang et al. “A hybrid system for temporal information extraction from clinical text”. en. In: *Journal of the American Medical Informatics Association* 20.5 (Sept. 2013), pp. 828–835. ISSN: 1067-5027. DOI: 10.1136/amiajnl-2013-001635. URL: http://academic.oup.com/jamia/article/20/5/828/727128 (visited on 11/13/2018).

[59] Yan Xu et al. “An end-to-end system to identify temporal relation in discharge summaries: 2012 i2b2 challenge”. en. In: *Journal of the American Medical Informatics Association* 20.5 (Sept. 2013), pp. 849–858. ISSN: 1067-5027. DOI: 10.1136/amiajnl-2012-001607. URL: http://academic.oup.com/jamia/article/20/5/849/727891 (visited on 11/13/2018).

[60] Cyril Grouin et al. "Eventual situations for timeline extraction from clinical reports". en. In: *Journal of the American Medical Informatics Association* 20.5 (Sept. 2013), pp. 820–827. ISSN: 1067-5027. DOI: 10.1136/amiajnl-2013-001627. URL: http://academic.oup.com/jamia/article/20/5/820/726830 (visited on 11/13/2018).

[61] Jennifer D'Souza and Vincent Ng. "Temporal Relation Identification and Classification in Clinical Notes". In: *Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics.* BCB'13. Wshington DC, USA: Association for Computing Machinery, Sept. 2013, pp. 392–401. ISBN: 978-1-4503-2434-2. DOI: 10.1145/2506583.2506654. URL: http://doi.org/10.1145/2506583.2506654 (visited on 07/05/2020).

[62] Yung-Chun Chang et al. "TEMPTING system: A hybrid method of rule and machine learning for temporal relation extraction in patient discharge summaries". en. In: *Journal of Biomedical Informatics.* 2012 i2b2 NLP Challenge on Temporal Relations in Clinical Data 46 (Dec. 2013), S54–S62. ISSN: 1532-0464. DOI: 10.1016/j.jbi.2013.09.007. URL: http://www.sciencedirect.com/science/article/pii/S1532046413001482 (visited on 06/20/2020).

[63] Dmitriy Dligach et al. "Neural Temporal Relation Extraction". In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers.* Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 746–751. URL: http://www.aclweb.org/anthology/E17-2118 (visited on 11/13/2018).

[64] Julien Tourille et al. "Neural Architecture for Temporal Relation Extraction: A Bi-LSTM Approach for Detecting Narrative Containers". In: *Proceedings*

of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 224–230. DOI: `10.18653/v1/P17-2035`. URL: `https://www.aclweb.org/anthology/P17-2035` (visited on 07/11/2020).

[65] Chen Lin et al. "Self-training improves Recurrent Neural Networks performance for Temporal Relation Extraction". In: *Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 165–176. DOI: `10.18653/v1/W18-5619`. URL: `https://www.aclweb.org/anthology/W18-5619` (visited on 07/11/2020).

[66] Zhiyuan Liu, Yankai Lin, and Maosong Sun. "Representation Learning and NLP". en. In: *Representation Learning for Natural Language Processing*. Ed. by Zhiyuan Liu, Yankai Lin, and Maosong Sun. Singapore: Springer, 2020, pp. 1–11. ISBN: 9789811555732. DOI: `10.1007/978-981-15-5573-2_1`. URL: `https://doi.org/10.1007/978-981-15-5573-2_1` (visited on 10/11/2020).

[67] Chen Lin et al. "A BERT-based Universal Model for Both Within- and Cross-sentence Clinical Temporal Relation Extraction". In: *Proceedings of the 2nd Clinical Natural Language Processing Workshop*. Minneapolis, Minnesota, USA: Association for Computational Linguistics, June 2019, pp. 65–71. DOI: `10.18653/v1/W19-1908`. URL: `https://www.aclweb.org/anthology/W19-1908` (visited on 07/11/2020).

[68] Alistair E. W. Johnson et al. "MIMIC-III, a freely accessible critical care database". en. In: *Scientific Data* 3.1 (May 2016). Number: 1 Publisher: Nature Publishing Group, p. 160035. ISSN: 2052-4463. DOI: `10.1038/sdata.`

2016.35. URL: https://www.nature.com/articles/sdata201635 (visited on 03/06/2021).

[69] Chen Lin et al. "A BERT-based One-Pass Multi-Task Model for Clinical Temporal Relation Extraction". In: *Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing*. Online: Association for Computational Linguistics, July 2020, pp. 70–75. URL: https://www.aclweb.org/anthology/2020.bionlp-1.7 (visited on 07/11/2020).

[70] Haoyu Wang et al. "Extracting Multiple-Relations in One-Pass with Pre-Trained Transformers". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 1371–1377. DOI: 10.18653/v1/P19-1132. URL: https://www.aclweb.org/anthology/P19-1132 (visited on 10/11/2020).

[71] Louise Dupuis et al. "Relative and Incomplete Time Expression Anchoring for Clinical Text". In: *Proceedings of the 3rd Clinical Natural Language Processing Workshop*. Online: Association for Computational Linguistics, Nov. 2020, pp. 117–129. DOI: 10.18653/v1/2020.clinicalnlp-1.14. URL: https://www.aclweb.org/anthology/2020.clinicalnlp-1.14 (visited on 05/01/2021).

[72] Emily Alsentzer et al. "Publicly Available Clinical BERT Embeddings". In: *arXiv:1904.03323 [cs]* (Apr. 2019). arXiv: 1904.03323. URL: http://arxiv.org/abs/1904.03323 (visited on 08/26/2019).

[73] Rhea Sukthanker et al. "Anaphora and coreference resolution: A review". en. In: *Information Fusion* 59 (July 2020), pp. 139–162. ISSN: 1566-2535. DOI:

10.1016/j.inffus.2020.01.010. URL: http://www.sciencedirect.com/science/article/pii/S1566253519303677 (visited on 07/11/2020).

[74] Sameer Pradhan et al. “CoNLL-2011 Shared Task: Modeling Unrestricted Coreference in OntoNotes”. In: *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*. Portland, Oregon, USA: Association for Computational Linguistics, June 2011, pp. 1–27. URL: https://www.aclweb.org/anthology/W11-1901 (visited on 09/27/2020).

[75] Jiaping Zheng et al. “Coreference resolution: A review of general methodologies and applications in the clinical domain”. en. In: *Journal of Biomedical Informatics* 44.6 (Dec. 2011), pp. 1113–1122. ISSN: 1532-0464. DOI: 10.1016/j.jbi.2011.08.006. URL: http://www.sciencedirect.com/science/article/pii/S153204641100133X (visited on 09/27/2020).

[76] Vincent Ng. “Supervised Noun Phrase Coreference Research: The First Fifteen Years”. In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Uppsala, Sweden: Association for Computational Linguistics, July 2010, pp. 1396–1411. URL: https://www.aclweb.org/anthology/P10-1142 (visited on 09/27/2020).

[77] Ozlem Uzuner et al. “Evaluating the state of the art in coreference resolution for electronic medical records”. en. In: *Journal of the American Medical Informatics Association* 19.5 (Sept. 2012), pp. 786–791. ISSN: 1067-5027. DOI: 10.1136/amiajnl-2011-000784. URL: http://academic.oup.com/jamia/article/19/5/786/716138 (visited on 01/13/2019).

[78] Olivier Bodenreider. “The Unified Medical Language System (UMLS): integrating biomedical terminology”. In: *Nucleic Acids Research* 32.Database issue (Jan. 2004), pp. D267–D270. ISSN: 0305-1048. DOI: 10.1093/nar/gkh061.

URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC308795/ (visited on 11/24/2020).

[79] Yan Xu et al. "A classification approach to coreference in discharge summaries: 2011 i2b2 challenge". en. In: *Journal of the American Medical Informatics Association* 19.5 (Sept. 2012), pp. 897–905. ISSN: 1067-5027. DOI: 10.1136/amiajnl-2011-000734. URL: http://academic.oup.com/jamia/article/19/5/897/722288 (visited on 01/13/2019).

[80] Rui Zhang et al. "Evaluating Measures of Redundancy in Clinical Texts". In: *AMIA Annual Symposium Proceedings* 2011 (2011), pp. 1612–1620. ISSN: 1942-597X. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3243221/ (visited on 02/09/2021).

[81] David Hinote, Carlos Ramirez, and Ping Chen. "A comparative study of co-refernece resolution in clinical text". In: *Proceedings of the 2011 i2b2/VA/Cincinnati Workshop on Challenges in Natural Language Processing for Clinical Data*. Boston, MA, USA, 2011.

[82] P Anick et al. "Coreference resolution for electronic medical records". In: *Proceedings of the 2011 i2b2/VA/Cincinnati Workshop on Challenges in Natural Language Processing for Clinical Data*. Boston, MA, USA, 2011.

[83] Preethi Raghavan et al. "Cross-narrative Temporal Ordering of Medical Events". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Baltimore, Maryland: Association for Computational Linguistics, June 2014, pp. 998–1008. DOI: 10.3115/v1/P14-1094. URL: https://www.aclweb.org/anthology/P14-1094 (visited on 05/09/2020).

[84] Rhea Sukthanker et al. “Anaphora and Coreference Resolution: A Review”. In: *arXiv:1805.11824 [cs]* (May 2018). arXiv: 1805.11824. URL: http://arxiv.org/abs/1805.11824 (visited on 09/27/2020).

[85] Azad Dehghan et al. “Combining knowledge- and data-driven methods for de-identification of clinical narratives”. In: *Journal of Biomedical Informatics*. Proceedings of the 2014 i2b2/UTHealth Shared-Tasks and Workshop on Challenges in Natural Language Processing for Clinical Data 58 (Dec. 2015), S53–S59. ISSN: 1532-0464. DOI: 10.1016/j.jbi.2015.06.029. URL: http://www.sciencedirect.com/science/article/pii/S1532046415001392 (visited on 01/03/2019).

[86] Philip Bramsen et al. “Inducing Temporal Graphs”. In: *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. Sydney, Australia: Association for Computational Linguistics, July 2006, pp. 189–198. URL: https://www.aclweb.org/anthology/W06-1623 (visited on 05/23/2020).

[87] Preethi Raghavan, Eric Fosler-Lussier, and Albert M. Lai. “Exploring Semi-supervised Coreference Resolution of Medical Concepts Using Semantic and Temporal Features”. In: *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. NAACL HLT ’12. Stroudsburg, PA, USA: Association for Computational Linguistics, 2012, pp. 731–741. ISBN: 978-1-937284-20-6. URL: http://dl.acm.org/citation.cfm?id=2382029.2382148 (visited on 01/12/2019).

[88] Serena Jeblee and Graeme Hirst. “Listwise temporal ordering of events in clinical notes”. In: *Proceedings of the Ninth International Workshop on Health*

Text Mining and Information Analysis. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 177–182. DOI: 10.18653/v1/W18-5620. URL: https://www.aclweb.org/anthology/W18-5620 (visited on 05/16/2020).

[89] Susana Martins et al. *Evaluation of KNAVE-II: a Tool for Intelligent Query and Exploration of Patient Data*. 2004.

[90] Alex A. T. Bui, Denise R. Aberle, and Hooshang Kangarloo. "TimeLine: Visualizing Integrated Patient Records". In: *IEEE Transactions on Information Technology in Biomedicine* 11.4 (July 2007), pp. 462–473. ISSN: 1558-0032. DOI: 10.1109/TITB.2006.884365.

[91] Jamie S. Hirsch et al. "HARVEST, a longitudinal patient record summarizer". en. In: *Journal of the American Medical Informatics Association* 22.2 (Mar. 2015). Publisher: Oxford Academic, pp. 263–274. ISSN: 1067-5027. DOI: 10.1136/amiajnl-2014-002945. URL: https://academic.oup.com/jamia/article/22/2/263/694965 (visited on 05/09/2020).

[92] "PatientExploreR: an extensible application for dynamic visualization of patient clinical history from electronic health records in the OMOP common data model". en. In: *Bioinformatics* 35.21 (Nov. 2019). Publisher: Oxford Academic, pp. 4515–4518. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btz409. URL: https://academic.oup.com/bioinformatics/article/35/21/4515/5520433 (visited on 09/06/2020).

[93] Denis Klimov, Yuval Shahar, and Meirav Taieb-Maimon. "Intelligent visualization and exploration of time-oriented data of multiple patients". en. In: *Artificial Intelligence in Medicine* 49.1 (May 2010), pp. 11–31. ISSN: 0933-3657.

DOI: 10.1016/j.artmed.2010.02.001. URL: http://www.sciencedirect.com/science/article/pii/S0933365710000229 (visited on 05/09/2020).

[94] David Gotz, Fei Wang, and Adam Perer. “A methodology for interactive mining and visual analysis of clinical event patterns using electronic health record data”. en. In: *Journal of Biomedical Informatics* 48 (Apr. 2014), pp. 148–159. ISSN: 1532-0464. DOI: 10.1016/j.jbi.2014.01.007. URL: http://www.sciencedirect.com/science/article/pii/S1532046414000094 (visited on 09/06/2020).

[95] Heekyong Park and Jinwook Choi. “V-model: a new innovative model to chronologically visualize narrative clinical texts”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’12. New York, NY, USA: Association for Computing Machinery, May 2012, pp. 453–462. ISBN: 978-1-4503-1015-4. DOI: 10.1145/2207676.2207739. URL: http://doi.org/10.1145/2207676.2207739 (visited on 09/06/2020).

[96] Azadeh Nikfarjam, Ehsan Emadzadeh, and Graciela Gonzalez. “Towards generating a patient’s timeline: Extracting temporal relationships from clinical notes”. English. In: *Journal of Biomedical Informatics* Supplement.46 (2013), S40–S47. ISSN: 1532-0464. DOI: 10.1016/j.jbi.2013.11.001. URL: https://www.infona.pl//resource/bwmeta1.element.elsevier-7915e0d4-9dac-3930-b7d8-a0e7af77045e (visited on 11/12/2018).

[97] Adyasha Maharana. “Extraction of Clinical Timeline from Discharge Summaries using Neural Networks”. en_US. Accepted: 2018-01-20T00:57:43Z. Thesis. Dec. 2017. URL: https://digital.lib.washington.edu:443/researchworks/handle/1773/40821 (visited on 07/11/2020).

[98] Catalina Hallett. "Multi-modal presentation of medical histories". In: *Proceedings of the 13th international conference on Intelligent user interfaces.* IUI '08. Gran Canaria, Spain: Association for Computing Machinery, Jan. 2008, pp. 80–89. ISBN: 978-1-59593-987-6. DOI: 10.1145/1378773.1378785. URL: http://doi.org/10.1145/1378773.1378785 (visited on 05/30/2020).

[99] J. Rogers, C. Puleston, and A. Rector. "The CLEF Chronicle: Patient Histories Derived from Electronic Health Records". In: *22nd International Conference on Data Engineering Workshops (ICDEW'06).* Apr. 2006, pp. x109–x109. DOI: 10.1109/ICDEW.2006.144.

[100] Henk Harkema et al. *Information-Extraction-from-Clinical-Records.pdf.* 2005. URL: https://www.researchgate.net/profile/M_Hepple/publication/246471156_Information_Extraction_from_Clinical_Records/links/53e9d2720cf2dc24b3cad9bf/Information-Extraction-from-Clinical-Records.pdf (visited on 05/30/2020).

[101] Zhou Yuan et al. "Interactive Exploration of Longitudinal Cancer Patient Histories Extracted From Clinical Text". In: *JCO Clinical Cancer Informatics* 4 (May 2020). Publisher: American Society of Clinical Oncology, pp. 412–420. DOI: 10.1200/CCI.19.00115. URL: https://ascopubs.org/doi/full/10.1200/CCI.19.00115 (visited on 07/11/2020).

[102] Egoitz Laparra, Itziar Aldabe, and German Rigau. "Document Level Time-anchoring for TimeLine Extraction". en. In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers).* Beijing, China: Association for Computational Linguistics, 2015,

pp. 358–364. DOI: 10.3115/v1/P15-2059. URL: http://aclweb.org/anthology/P15-2059 (visited on 11/13/2018).

[103] Savelie Cornegruta and Andreas Vlachos. "Timeline extraction using distant supervision and joint inference". In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 1936–1942. DOI: 10.18653/v1/D16-1200. URL: https://www.aclweb.org/anthology/D16-1200 (visited on 05/16/2020).

[104] Estela Saquete Boró and Borja Navarro Colorado. "Cross-Document Event Ordering through Temporal Relation Inference and Distributional Semantic Models". eng. In: (Mar. 2017). ISSN: 1135-5948. URL: http://rua.ua.es/dspace/handle/10045/64031 (visited on 11/13/2018).

[105] Egoitz Laparra et al. "Multi-lingual and Cross-lingual timeline extraction". In: *Knowledge-Based Systems* 133 (Oct. 2017), pp. 77–89. ISSN: 0950-7051. DOI: 10.1016/j.knosys.2017.07.002. URL: http://www.sciencedirect.com/science/article/pii/S0950705117303192 (visited on 11/13/2018).

[106] Li Zhou et al. "System Architecture for Temporal Information Extraction, Representation and Reasoning in Clinical Narrative Reports". In: *AMIA Annual Symposium Proceedings* 2005 (2005), pp. 869–873. ISSN: 1942-597X. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1560711/ (visited on 05/16/2020).

[107] Hyuckchul Jung et al. "Building Timelines from Narrative Clinical Records: Initial Results Based-on Deep Natural Language Understanding". In: *Proceedings of BioNLP 2011 Workshop*. BioNLP '11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 146–154. ISBN: 978-1-932432-

91-6. URL: http://dl.acm.org/citation.cfm?id=2002902.2002924 (visited on 01/05/2019).

[108] James Allen et al. “An architecture for a generic dialogue shell”. en. In: *Natural Language Engineering* 6.3-4 (Sept. 2000). Publisher: Cambridge University Press, pp. 213–228. ISSN: 1469-8110, 1351-3249. DOI: 10.1017/S135132490000245X. URL: https://www.cambridge.org/core/journals/natural-language-engineering/article/an-architecture-for-a-generic-dialogue-shell/87FF9A27E4AC8BBB24220296D6E0E4C8 (visited on 10/11/2020).

[109] Marc Verhagen. “Temporal Closure in an Annotation Environment”. In: *Language Resources and Evaluation* 39.2/3 (2005). Publisher: Springer, pp. 211–241. ISSN: 1574-020X. URL: https://www.jstor.org/stable/30200551 (visited on 07/05/2020).

[110] William W. Cohen, Pradeep Ravikumar, and Stephen E. Fienberg. “A Comparison of String Distance Metrics for Name-Matching Tasks”. In: 2003, pp. 73–78.

[111] David Ferrucci and Adam Lally. “UIMA: an architectural approach to unstructured information processing in the corporate research environment”. In: *Natural Language Engineering* 10.3-4 (Sept. 2004), pp. 327–348. ISSN: 1351-3249. DOI: 10.1017/S1351324904003523. URL: https://doi.org/10.1017/S1351324904003523 (visited on 11/24/2020).

[112] Ernestina Menasalvas Ruiz et al. “Profiling Lung Cancer Patients Using Electronic Health Records”. en. In: *Journal of Medical Systems* 42.7 (May 2018), p. 126. ISSN: 1573-689X. DOI: 10.1007/s10916-018-0975-9. URL: https://doi.org/10.1007/s10916-018-0975-9 (visited on 11/24/2020).

[113] Marjan Najafabadipour et al. "Lung Cancer Concept Annotation from Spanish Clinical Narratives". en. In: *Data Integration in the Life Sciences*. Ed. by Sören Auer and Maria-Esther Vidal. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019, pp. 153–163. ISBN: 978-3-030-06016-9. DOI: 10.1007/978-3-030-06016-9_15.

[114] M. Najafabadipour et al. "Recognition of Time Expressions in Spanish Electronic Health Records". In: *2019 IEEE 32nd International Symposium on Computer-Based Medical Systems (CBMS)*. ISSN: 2372-9198. June 2019, pp. 69–74. DOI: 10.1109/CBMS.2019.00025.

[115] Milan Straka, Jan Hajič, and Jana Straková. "UDPipe: Trainable Pipeline for Processing CoNLL-U Files Performing Tokenization, Morphological Analysis, POS Tagging and Parsing". In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. Portorož, Slovenia: European Language Resources Association (ELRA), May 2016, pp. 4290–4297. URL: https://www.aclweb.org/anthology/L16-1680 (visited on 11/24/2020).

[116] Preethi Raghavan. "MEDICAL EVENT TIMELINE GENERATION FROM CLINICAL NARRATIVES". en. PhD thesis. The Ohio State University, 2014. URL: https://etd.ohiolink.edu/pg_10?0::NO:10:P10_ACCESSION_NUM:osu1397651496 (visited on 11/13/2018).

[117] Tomas Mikolov et al. "Distributed Representations of Words and Phrases and their Compositionality". In: *Advances in Neural Information Processing Systems 26*. Ed. by C. J. C. Burges et al. Curran Associates, Inc., 2013, pp. 3111–3119. URL: http://papers.nips.cc/paper/5021-distributed-

representations-of-words-and-phrases-and-their-compositionality.pdf (visited on 09/26/2019).

[118] Jeffrey Pennington, Richard Socher, and Christopher Manning. "Glove: Global Vectors for Word Representation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. DOI: 10.3115/v1/D14-1162. URL: https://www.aclweb.org/anthology/D14-1162 (visited on 09/09/2019).

[119] Matthew Peters et al. "Deep Contextualized Word Representations". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 2227–2237. DOI: 10.18653/v1/N18-1202. URL: https://www.aclweb.org/anthology/N18-1202 (visited on 09/27/2019).

[120] Alec Radford et al. "Improving Language Understanding by Generative Pre-Training". en. In: *Technical Report* (2018), p. 12.

[121] Chris McCormick and Nick Ryan. *BERT Fine-Tuning Tutorial with PyTorch*. July 2019. URL: https://mccormickml.com/2019/07/22/BERT-fine-tuning/ (visited on 08/30/2019).

[122] Sebastian Ruder et al. "Transfer Learning in Natural Language Processing". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 15–18. DOI: 10.18653/v1/N19-5004. URL: https://aclanthology.org/N19-5004 (visited on 04/09/2022).

[123] Ashish Vaswani et al. “Attention is All you Need”. In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon et al. Curran Associates, Inc., 2017, pp. 5998–6008. URL: `http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf` (visited on 09/16/2019).

[124] Egoitz Laparra, Dongfang Xu, and Steven Bethard. “From Characters to Time Intervals: New Paradigms for Evaluation and Neural Parsing of Time Normalizations”. In: *Transactions of the Association for Computational Linguistics* 6 (2018), pp. 343–356. ISSN: 2307-387X. URL: `https://www.transacl.org/ojs/index.php/tacl/article/view/1318` (visited on 06/30/2018).

[125] Steven Bird and Edward Loper. “NLTK: the natural language toolkit”. In: *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*. Association for Computational Linguistics. 2004, p. 31.

[126] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[127] Wei-Te Chen and Will Styler. “Anafora: A Web-based General Purpose Annotation Tool”. In: *Proceedings of the 2013 NAACL HLT Demonstration Session*. Atlanta, Georgia: Association for Computational Linguistics, June 2013, pp. 14–19. URL: `https://www.aclweb.org/anthology/N13-3004` (visited on 03/29/2019).

[128] David Graff. *The AQUAINT Corpus of English News Text LDC2002T31*. Philadelphia: Linguistic Data Consortium, 2002. 2002. URL: `https://catalog.ldc.upenn.edu/LDC2002T31` (visited on 12/08/2018).

[129] William F. Styler IV et al. “Temporal annotation in the clinical domain”. In: *Transactions of the Association for Computational Linguistics* 2 (2014), p. 143.

[130] Egoitz Laparra et al. “SemEval 2018 Task 6: Parsing Time Normalizations”. In: *Proceedings of The 12th International Workshop on Semantic Evaluation*. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 88–96. URL: http://www.aclweb.org/anthology/S18-1011 (visited on 07/03/2018).

[131] Stephane M. Meystre et al. “Extracting Information from Textual Documents in the Electronic Health Record: A Review of Recent Research”. en. In: *Yearbook of Medical Informatics* 17.1 (2008), pp. 128–144. ISSN: 0943-4747, 2364-0502. DOI: 10.1055/s-0038-1638592. URL: http://www.thieme-connect.de/DOI/DOI?10.1055/s-0038-1638592 (visited on 12/09/2018).

[132] Weiyi Sun, Anna Rumshisky, and Ozlem Uzuner. “Annotating temporal information in clinical narratives”. In: *Journal of Biomedical Informatics*. 2012 i2b2 NLP Challenge on Temporal Relations in Clinical Data 46 (Dec. 2013), S5–S12. ISSN: 1532-0464. DOI: 10.1016/j.jbi.2013.07.004. URL: http://www.sciencedirect.com/science/article/pii/S1532046413001032 (visited on 11/12/2018).

[133] Florentina Hristea and Mihaela Colhon. “The long road from performing word sense disambiguation to successfully using it in information retrieval: An overview of the unsupervised approach”. en. In: *Computational Intelligence* 36.3 (2020). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/coin.12303, pp. 1026–1062. ISSN: 1467-8640. DOI: 10.1111/coin.12303. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/coin.12303 (visited on 04/09/2022).

[134] Bridget T. McInnes and Mark Stevenson. “Determining the difficulty of Word Sense Disambiguation”. en. In: *Journal of Biomedical Informatics* 47 (Feb.

2014), pp. 83–90. ISSN: 1532-0464. DOI: 10.1016/j.jbi.2013.09.009. URL: https://www.sciencedirect.com/science/article/pii/S1532046413001500 (visited on 04/09/2022).

[135] Rui Antunes and Sérgio Matos. “Supervised Learning and Knowledge-Based Approaches Applied to Biomedical Word Sense Disambiguation”. en. In: *Journal of Integrative Bioinformatics* 14.4 (Dec. 2017). Publisher: De Gruyter. ISSN: 1613-4516. DOI: 10.1515/jib-2017-0051. URL: https://www.degruyter.com/document/doi/10.1515/jib-2017-0051/html (visited on 04/09/2022).

[136] Manabu Torii, Jung-Wei Fan, and Daniel S. Zisook. “Finding Difficult-to-Disambiguate Words: Towards an Efficient Workflow to Implement Word Sense Disambiguation”. In: *2015 International Conference on Healthcare Informatics*. Oct. 2015, pp. 448–448. DOI: 10.1109/ICHI.2015.66.

[137] Inderjeet Mani. “Recent Developments in Temporal Information Extraction”. In: *Proceedings of RANLP’03*. John Benjamins, 2004, pp. 45–60.

[138] Robert Dale et al. *Natural Language Processing – IJCNLP 2005: Second International Joint Conference, Jeju Island, Korea, October 11-13, 2005, Proceedings*. en. Springer Science & Business Media, Sept. 2005. ISBN: 978-3-540-29172-5.

[139] David A. Hanauer et al. “Complexities, variations, and errors of numbering within clinical notes: the potential impact on information extraction and cohort-identification”. en. In: *BMC Medical Informatics and Decision Making* 19.3 (Apr. 2019), p. 75. ISSN: 1472-6947. DOI: 10.1186/s12911-019-0784-1. URL: https://doi.org/10.1186/s12911-019-0784-1 (visited on 04/09/2022).

[140] Xue Wu et al. "Residents' numeric inputting error in computerized physician order entry prescription". en. In: *International Journal of Medical Informatics* 88 (Apr. 2016), pp. 25–33. ISSN: 1386-5056. DOI: 10.1016/j.ijmedinf.2016.01.002. URL: https://www.sciencedirect.com/science/article/pii/S1386505616300028 (visited on 04/09/2022).

[141] Dianbo Liu, Dmitriy Dligach, and Timothy Miller. "Two-stage Federated Phenotyping and Patient Representation Learning". In: *arXiv:1908.05596 [cs]* (Aug. 2019). arXiv: 1908.05596. URL: http://arxiv.org/abs/1908.05596 (visited on 09/09/2019).

[142] Hong Guan and Murthy Devarakonda. "Leveraging Contextual Information in Extracting Long Distance Relations from Clinical Notes". In: *AMIA Annual Symposium Proceedings* 2019 (Mar. 2020), pp. 1051–1060. ISSN: 1942-597X. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7153124/ (visited on 07/04/2020).

[143] Satya Almasian, Dennis Aumiller, and Michael Gertz. "BERT got a Date: Introducing Transformers to Temporal Tagging". In: *arXiv:2109.14927 [cs]* (Jan. 2022). arXiv: 2109.14927. URL: http://arxiv.org/abs/2109.14927 (visited on 02/12/2022).

[144] Chao Pang et al. "CEHR-BERT: Incorporating temporal information from structured EHR data to improve prediction tasks". In: *arXiv:2111.08585 [cs]* (Nov. 2021). arXiv: 2111.08585. URL: http://arxiv.org/abs/2111.08585 (visited on 02/12/2022).

[145] Thomas Wolf et al. "Transformers: State-of-the-Art Natural Language Processing". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for

Computational Linguistics, Oct. 2020, pp. 38–45. DOI: 10.18653/v1/2020.emnlp-demos.6. URL: https://aclanthology.org/2020.emnlp-demos.6 (visited on 03/20/2022).

[146] Dongfang Xu, Egoitz Laparra, and Steven Bethard. "Pre-trained Contextualized Character Embeddings Lead to Major Improvements in Time Normalization: a Detailed Analysis". In: *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019).* Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 68–74. DOI: 10.18653/v1/S19-1008. URL: https://www.aclweb.org/anthology/S19-1008 (visited on 04/25/2020).

[147] Zewen Li et al. "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects". In: *IEEE Transactions on Neural Networks and Learning Systems* (2021). Conference Name: IEEE Transactions on Neural Networks and Learning Systems, pp. 1–21. ISSN: 2162-2388. DOI: 10.1109/TNNLS.2021.3084827.

[148] Ping Li, Jianping Li, and Gongcheng Wang. "Application of Convolutional Neural Network in Natural Language Processing". In: *2018 15th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP).* ISSN: 2576-8964. Dec. 2018, pp. 120–122. DOI: 10.1109/ICCWAMTIP.2018.8632576.

[149] N. I. Widiastuti. "Convolution Neural Network for Text Mining and Natural Language Processing". en. In: *IOP Conference Series: Materials Science and Engineering* 662.5 (Nov. 2019). Publisher: IOP Publishing, p. 052010. ISSN: 1757-899X. DOI: 10.1088/1757-899X/662/5/052010. URL: https://doi.org/10.1088/1757-899x/662/5/052010 (visited on 04/09/2022).

[150] Hua Xu, Peter D. Stetson, and Carol Friedman. "A Study of Abbreviations in Clinical Notes". In: *AMIA Annual Symposium Proceedings* 2007 (2007), pp. 821–825. ISSN: 1942-597X. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2655910/ (visited on 04/16/2022).

[151] Youngjun Kim, Ellen Riloff, and John F. Hurdle. "A Study of Concept Extraction Across Different Types of Clinical Notes". In: *AMIA Annual Symposium Proceedings* 2015 (Nov. 2015), pp. 737–746. ISSN: 1942-597X. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4765588/ (visited on 04/16/2022).

[152] Rishi Kanth Saripalle. "Fast Health Interoperability Resources (FHIR): Current Status in the Healthcare System". en. In: *International Journal of E-Health and Medical Communications (IJEHMC)* 10.1 (Jan. 2019). Publisher: IGI Global, pp. 76–93. ISSN: 1947-315X. DOI: 10.4018/IJEHMC.2019010105. URL: https://www.igi-global.com/article/fast-health-interoperability-resources-fhir/www.igi-global.com/article/fast-health-interoperability-resources-fhir/215344 (visited on 04/16/2022).

[153] Clement J McDonald et al. "LOINC, a Universal Standard for Identifying Laboratory Observations: A 5-Year Update". In: *Clinical Chemistry* 49.4 (Apr. 2003), pp. 624–633. ISSN: 0009-9147. DOI: 10.1373/49.4.624. URL: https://doi.org/10.1373/49.4.624 (visited on 04/16/2022).

[154] Alan Akbik, Duncan Blythe, and Roland Vollgraf. "Contextual String Embeddings for Sequence Labeling". In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 1638–1649. URL: https://www.aclweb.org/anthology/C18-1139 (visited on 09/09/2019).

[155] Philip Bramsen et al. “Finding Temporal Order in Discharge Summaries”. In: *AMIA Annual Symposium Proceedings* 2006 (2006), pp. 81–85. ISSN: 1942-597X. URL: `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1839632/` (visited on 05/23/2020).