NLP Whack-A-Mole: Challenges in Cross-Domain Temporal Expression Extraction

Amy L. Olex, Luke G. Maffey, and Bridget T. McInnes
Department of Computer Science
Virginia Commonwealth University
401 East Main Street, Richmond VA 23298
alolex,maffeyl,btmcinnes@vcu.edu

Abstract

Incorporating domain knowledge is vital in building successful natural language processing (NLP) applications. Many times, cross-domain application of a tool results in poor performance as the tool does not account for domain-specific attributes. The clinical domain is challenging in this aspect due to specialized medical terms and nomenclature, shorthand notation, fragmented text, and a variety of writing styles used by different medical units. Temporal resolution is an NLP task that, in general, is domain-agnostic because temporal information is represented using a limited lexicon. However, domain-specific aspects of temporal resolution are present in clinical texts. Here we explore parsing issues that arose when running our system, a tool built on Newswire text, on clinical notes in the THYME corpus. Many parsing issues were straightforward to correct; however, a few code changes resulted in a cascading series of parsing errors that had to be resolved before an improvement in performance was observed, revealing the complexity of temporal resolution and rule-based parsing. Our system now outperforms current state-of-the-art systems on the THYME corpus with little change in its performance on Newswire texts.

1 Introduction

Temporal resolution is required for comprehending many types of communication, including written texts. This is especially true in clinical texts as patient narratives revolve around when an event happened, such as when a symptom occurred or the frequency a drug was administered (Lee et al., 2017; Sun et al., 2013b). Understanding the temporal component in texts is vital for many NLP systems (Tissot et al., 2015) to accurately interpret a patient narrative (Sun et al., 2013b).

Some temporal expressions could be considered domain agnostic as there are limited ways to represent information about time, such as formatted dates or days of the week. However, there are many lexical variations of these standard tokens. Additionally, vague temporal expressions, relative times, and event durations require contextual or implicit knowledge of the subject area for resolution (Sun et al., 2013b). Clinical texts include all these types of temporal expressions, and also contain domain-specific challenges to temporal expression identification and normalization, such as differentiating between dosage and time. Additionally, clinical texts frequently use repeated phrases such as "At this time" that are infrequently used in the general domain. These phrases are vague, relative, and require contextual knowledge of the subject area and the time of events to be resolved (Sun et al., 2013b).

In this work we focus on identification of temporal expressions in clinical texts using Chrono–a hybrid system that normalizes temporal expressions into the SCATE Schema (Bethard and Parker, 2016). Originally designed on general domain Newswire texts, we evaluate Chrono's performance on the clinical THYME corpus (Styler et al., 2014) "out-of-the-box" with no modifications, perform an error analysis, algorithm updates, and then re-evaluate on THYME. This analysis reveals six aspects of temporal expression extraction that should be considered when using a general domain tool in the clinical domain.

2 Related Work

State-of-the-art temporal expression extraction and normalization tools have emerged from temporal parsing challenges such as TempEval (Verhagen et al., 2007, 2010; UzZaman et al., 2013) and i2b2 (Sun et al., 2013a). Strategies utilized by these tools range from rule-based (SUTime (Chang and Manning, 2012), HeidelTime

Proceedings of NAACL-HLT 2019, pages 3682–3692
Minneapolis, Minnesota, June 2 - June 7, 2019.

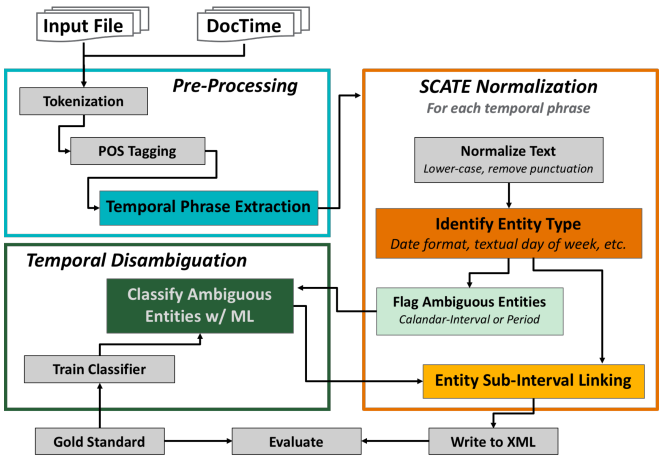

Figure 1: Overview of Chrono Workflow

(Strötgen and Gertz, 2010), NavyTime (Chambers, 2013), GUTime (Verhagen et al., 2005)) to machine learning (TRIPS and TRIOS (UzZaman and Allen, 2010), ClearTK (Bethard, 2013)) and hybrid approaches (ManTIME (Filannino et al., 2013)). For general domain texts, machine learning systems like ClearTK perform well at identifying temporal expression spans; however, rule-based and hybrid systems have better performance when taking temporal expression normalization into account (UzZaman et al., 2013).

When applied to clinical text in the 2012 i2b2 Challenge, high-ranking general domain systems SUTime, GUTime, and HeidelTime had reduced performance (Sun et al., 2013a) as compared to systems built specifically for this data (Sohn et al., 2013; Kovaevi et al., 2013; Xu et al., 2013). Regardless of the performance on general domain texts, modifications had to be made to the state-of-the-art systems to recognize clinical temporal expressions and achieve improved performance. For example, three teams utilized HeidelTime with two teams incorporating additional rules and machine learning modules on top of the default system, which achieved better performance in the 2012 i2b2 Challenge than HeidelTime with no modifications.

In addition to temporal challenges, other systems have been developed for general domain temporal parsing that utilize machine learning and complex grammars (Lee et al., 2014; Angeli et al., 2012) and rule-based methods referencing a central knowledge base (Llorens et al., 2012). SynTime (Zhong et al., 2017) takes a simplistic approach by defining a layer of syntactic token types that rules are applied to instead of processing the raw tokens. For temporal expression extraction, SynTime out-performs HeidelTime and SUTime, however, it does not attempt normalization. All these systems were built and trained on general domain texts, such as TimeBank (Pustejovsky et al., 2003) and WikiWars (Mazur and Dale, 2010) and may require adjustments to accurately capture clinical temporal expressions. In addition, these systems normalize expressions into the ISO-TimeML (Pustejovsky et al., 2010) representation, which is unable to represent expressions that don't map to a single calendar unit or are relative to an event instead of a temporal unit–both of which are frequent in clinical texts. The SCATE schema is able to faithfully represent these types of expressions, but normalization requires a more detailed approach to annotate fine-grained temporal components that are not captured by TimeML (Bethard and Parker, 2016). In this work we adapt Chrono, a novel SCATE normalization system, to the clinical domain and describe the challenges encountered when normalizing to the SCATE Schema.

3 Methods

Chrono is a hybrid rule-based and machine learning system built to identify temporal expressions in the AQUAINT corpus of Newswire texts (Graff, 2002) followed by normalization into the SCATE Schema for SemEval 2018 Task 6 (Laparra et al., 2018). Chrono consists of 3 main modules: 1) Temporal Phrase Extraction, 2) SCATE Normalization, and 3) Temporal Disambiguation (Figure 1). Briefly, the Temporal Phrase Extraction module identifies temporal/numeric tokens using a series of hierarchical rules and regular expressions. Temporal phrases are extracted based

on consecutive tagged temporal/numeric tokens. Next, the SCATE Normalization module normalizes temporal phrases into the SCATE Schema using additional rule-based logic and regular expressions to identify specific temporal entities within each phrase, and links related sub-intervals. Finally, machine learning is used in the Temporal Disambiguation module as a sub-module of SCATE Normalization to disambiguate certain SCATE entities. Details on the specific rules implemented by Chrono for SemEval 2018 can be found in the systems description paper (Olex et al., 2018), and Chrono can be downloaded from https://github.com/AmyOlex/Chrono.

4 THYME Corpus

The THYME corpus consists of de-identified clinical notes and pathology reports for colon and brain cancer patients. 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 (Laparra et al., 2018). 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. In this work, Chrono is first run on the THYME Evaluation Corpus before modifications are made, then the THYME Training Corpus is used to identify problem areas in need of improvement. Finally, Chrono is run on the Evaluation Corpus again after making improvements. Data in the Evaluation Corpus remained hidden through the entire process.

5 Evaluation

Evaluation of Chrono's performance on the Training Corpus utilized python scripts provided by AnaforaTools[†] that compare Anafora XML (Chen and Styler, 2013) 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 Corpus was uploaded to the Post-Evaluation submission system for SemEval 2018 Task 6, and overall Precision, Recall, and F1 measures are reported in Tables 1 and 3.

[†]https://github.com/bethard/anaforatools

6 Results and Discussion

This section first discusses Chrono's "out-of-the-box" performance on the THYME Evaluation Corpus prior to any code changes. The next section presents parsing issues encountered using the Training Corpus that fall into six main categories: 1) lexical, 2) entity frequency, 3) numeric disambiguation, 4) machine learning training data, 5) writing style, and 6) document structure. While fixing some of these issues was straightforward, more complex issues resulted in debugging an error cascade before performance increased. Finally, a discussion of Chrono's improved performance on the THYME Evaluation Corpus is presented.

6.1 Out-of-the-Box Performance on THYME

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 1). This is due to Chrono having only been trained on Newswire text, thus, it saw a limited 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.

Dataset	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 1: Baseline performance, excluding "Event", on THYME Training and Evaluation corpora using SVM.

6.2 NLP Whack-A-Mole - Resolving Cross-Domain NLP Challenges

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.

6.2.1 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 (Meystre et al., 2008). 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 CalendarInterval 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 2).

6.2.2 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 2). 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 2).

6.2.3 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

Entity	Chrono Implements	Newswire Avg Freq	Clinical 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 2: 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).

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.

6.2.4 Cross-Domain Machine 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 CalendarInterval. First, rule-based logic identifies if an entity is a possible Period or CalendarInterval, 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.

6.2.5 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 exam-

ple, 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".

6.2.6 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 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.

6.3 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 3). 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

Dataset	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 3: Improved performance on THYME Corpora using SVM, excluding "Event".

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.

7 Conclusions and Future Work

In conclusion, clinical domain texts posed additional challenges that were either not present in the Newswire corpus, or not frequent enough to prioritize highly when initially building Chrono. Application to the THMYE Training Corpus brought these limitations to light, such as the consistent use of temporal expressions that utilize frequency, highly repeated temporal phrases, dosage values being annotated as temporal expressions, and additional lexical elements. As temporal information is relatively domain agnostic, improvements made to Chrono for THYME should improve performance on other domains. An advantage of utilizing clinical texts is that it encounters a variety of writing styles from different practitioners who may prefer specifying temporal information in different ways. Additionally, different medical forms, such as pathology reports versus clinical notes, have specific ways to convey dates. Thus, the range of temporal expressions Chrono now identifies has been significantly expanded due to the variety incorporated in the clinical texts.

While Chrono's performance on SCATE Normalization has improved, there are still many areas for further development. These include identifying frequency, disambiguating dosage versus 4-digit year or 24-hour time, implementing more sophisticated approaches to parsing metadata, and performing a more detailed investigation at the entity level to identify which SCATE properties are being missed or incorrectly assigned in order to bring the 100% Correct Entity performance closer to the Span Only performance. These updates will require the implementation of additional rule-sets as well as the addition of machine learning modules and more complex contextual parsing. One approach to augmenting current rule sets is the automated generation of regular expressions (Redd et al., 2015) based on annotated gold standards, which has the potential to expand Chrono's capabilities without time-consuming human review of missed expressions. Finally, Chrono outputs normalized temporal expressions in the SCATE schema format, which limits our ability to evaluate its performance on corpora in other domains. Currently, only select subsets of the AQUAINT and THYME corpora are annotated with SCATE, and the complete conversion of TimeML to the SCATE schema is difficult as TimeML lacks details required by SCATE. Thus, implementation of a method to convert SCATE XML to the standard TimeML format will allow Chrono to be evaluated on additional cross-domain corpora and classic benchmark temporal corpora such as i2b2 (Sun et al., 2013a), TempEval (Verhagen et al., 2007, 2010; UzZaman et al., 2013), and Clinical TempEval (Bethard et al., 2015).

The process of improving Chrono brought to light several aspects of cross-domain application of temporal parsing: 1) lexical differences, 2) the frequency of temporal entity usage, 3) disambiguating numerical phrases, 4) appropriate machine learning data, 5) lexical variation of concepts, and 6) differences in document structure. While the concept of time is the same regardless of the domain, its representation can vary. Thus, temporal parsing provides a good backdrop for determining the challenges of cross-domain application, which is difficult for many NLP applications. The aspects of cross-domain application discussed herein provide a foundation for designing adaptable NLP tools that can be utilized across domains.

References

Gabor Angeli, Christopher Manning, and Daniel Jurafsky. 2012. Parsing Time: Learning to Interpret Time Expressions. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 446–455, Montral, Canada. Association for Computational Linguistics.

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

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

Steven Bethard and Jonathan Parker. 2016. A Semantically Compositional Annotation Scheme for Time Normalization. In *LREC*.

Nathanael Chambers. 2013. NavyTime: Event and Time Ordering from Raw Text. Technical report, NAVAL ACADEMY ANNAPOLIS MD.

Angel X. Chang and Christopher D. Manning. 2012. Sutime: A library for recognizing and normalizing time expressions. In *Lrec*, volume 2012, pages 3735–3740.

Wei-Te Chen and Will Styler. 2013. Anafora: A Web-based General Purpose Annotation Tool. In *Proceedings of the 2013 NAACL HLT Demonstration Session*, pages 14–19, Atlanta, Georgia. Association for Computational Linguistics.

Michele Filannino, Gavin Brown, and Goran Nenadic. 2013. ManTIME: Temporal expression identification and normalization in the TempEval-3 challenge. *arXiv:1304.7942 [cs]*. ArXiv: 1304.7942.

David Graff. 2002. The AQUAINT Corpus of English News Text LDC2002t31. Philadelphia: Linguistic Data Consortium, 2002.

Aleksandar Kovaevi, Azad Dehghan, Michele Filannino, John A. Keane, and Goran Nenadic. 2013. Combining rules and machine learning for extraction of temporal expressions and events from clinical narratives. *Journal of the American Medical Informatics Association*, 20(5):859–866.

Egoitz Laparra, Dongfang Xu, Ahmed Elsayed, Steven Bethard, and Martha Palmer. 2018. SemEval 2018 Task 6: Parsing Time Normalizations. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 88–96, New Orleans, Louisiana. Association for Computational Linguistics.

Hee-Jin Lee, Yaoyun Zhang, Jun Xu, Cui Tao, Hua Xu, and Min Jiang. 2017. Towards practical temporal relation extraction from clinical notes: An analysis of direct temporal relations. In *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1272–1275.

Kenton Lee, Yoav Artzi, Jesse Dodge, and Luke Zettlemoyer. 2014. Context-dependent Semantic Parsing for Time Expressions. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1437–1447, Baltimore, Maryland. Association for Computational Linguistics.

Hector Llorens, Leon Derczynski, Robert Gaizauskas, and Estela Saquete. 2012. TIMEN: An Open Temporal Expression Normalisation Resource. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*. European Language Resources Association (ELRA). Event-place: Istanbul, Turkey.

Pawet Mazur and Robert Dale. 2010. WikiWars: A New Corpus for Research on Temporal Expressions. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 913–922, Stroudsburg, PA, USA. Association for Computational Linguistics. Event-place: Cambridge, Massachusetts.

Stephane M. Meystre, Guergana K. Savova, Karin C. Kipper-Schuler, and John F. Hurdle. 2008. Extracting Information from Textual Documents in the Electronic Health Record: A Review of Recent Research. *Yearbook of Medical Informatics*, 17(1):128–144.

Amy Olex, Luke Maffey, Nicholas Morgan, and Bridget McInnes. 2018. Chrono at SemEval-2018 Task 6: A System for Normalizing Temporal Expressions. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 97–101, New Orleans, Louisiana. Association for Computational Linguistics.

James Pustejovsky, Patrick Hanks, Roser Saur, Andrew See, R Gaizauskas, A Setzer, Dragomir Radev, Beth Sundheim, David Day, Lisa Ferro, and Marcia Lazo. 2003. The TIMEBANK Corpus. In *Proceedings of Corpus Linguistics*, pages 647–656.

James Pustejovsky, Kiyong Lee, Harry Bunt, and Laurent Romary. 2010. ISO-TimeML: An International Standard for Semantic Annotation. In *Proceedings of the Seventh International Confer- ence on Language Resources and Evaluation (LREC10)*, page 4, Valletta, Malta.

Douglas Redd, YiJun Shaoa, Jing Yang, Guy Divita, and Qing Zeng-Treitler. 2015. Automated Learning

of Temporal Expressions. *Studies in Health Technology and Informatics*, 216:639–642.

Sunghwan Sohn, Kavishwar B. Wagholikar, Dingcheng Li, Siddhartha R. Jonnalagadda, Cui Tao, Ravikumar Komandur Elayavilli, and Hongfang Liu. 2013. Comprehensive temporal information detection from clinical text: medical events, time, and TLINK identification. *Journal of the American Medical Informatics Association*, 20(5):836–842.

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

William F. Styler, Steven Bethard, Sean Finan, Martha Palmer, Sameer Pradhan, Piet C de Groen, Brad Erickson, Timothy Miller, Chen Lin, Guergana Savova, and James Pustejovsky. 2014. Temporal Annotation in the Clinical Domain. *Transactions of the Association for Computational Linguistics*, 2:143–154.

Weiyi Sun, Anna Rumshisky, and Ozlem Uzuner. 2013a. Evaluating temporal relations in clinical text: 2012 i2b2 Challenge. *Journal of the American Medical Informatics Association : JAMIA*, 20(5):806–813.

Weiyi Sun, Anna Rumshisky, and Ozlem Uzuner. 2013b. Temporal reasoning over clinical text: the state of the art. *Journal of the American Medical Informatics Association*, 20(5):814–819.

Hegler Tissot, Angus Roberts, Leon Derczynski, Genevieve Gorrell, and Marcus Didonet Del Fabro. 2015. Analysis of temporal expressions annotated in clinical notes. In *Proceedings of the 11th Joint ACL-ISO Workshop on Interoperable Semantic Annotation (ISA-11)*.

Naushad UzZaman and James Allen. 2010. TRIPS and TRIOS System for TempEval-2: Extracting Temporal Information from Text. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 276–283, Uppsala, Sweden. Association for Computational Linguistics.

Naushad UzZaman, Hector Llorens, Leon Derczynski, James Allen, Marc Verhagen, and James Pustejovsky. 2013. SemEval-2013 Task 1: TempEval-3: Evaluating Time Expressions, Events, and Temporal Relations. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 1–9, Atlanta, Georgia, USA. Association for Computational Linguistics.

Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Graham Katz, and James Pustejovsky. 2007. SemEval-2007 Task 15: TempEval Temporal Relation Identification. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval '07, pages 75–80, Stroudsburg, PA, USA. Association for Computational Linguistics.

Marc Verhagen, Inderjeet Mani, Roser Sauri, Robert Knippen, Seok Bae Jang, Jessica Littman, Anna Rumshisky, John Phillips, and James Pustejovsky. 2005. Automating Temporal Annotation with TARSQI. In *Proceedings of the ACL 2005 on Interactive Poster and Demonstration Sessions*, ACLdemo '05, pages 81–84, Stroudsburg, PA, USA. Association for Computational Linguistics. Event-place: Ann Arbor, Michigan.

Marc Verhagen, Roser Saur, Tommaso Caselli, and James Pustejovsky. 2010. SemEval-2010 Task 13: TempEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, SemEval '10, pages 57–62, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yan Xu, Yining Wang, Tianren Liu, Junichi Tsujii, and Eric I.-Chao Chang. 2013. An end-to-end system to identify temporal relation in discharge summaries: 2012 i2b2 challenge. *Journal of the American Medical Informatics Association*, 20(5):849–858.

Xiaoshi Zhong, Aixin Sun, and Erik Cambria. 2017. Time Expression Analysis and Recognition Using Syntactic Token Types and General Heuristic Rules. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 420–429, Vancouver, Canada. Association for Computational Linguistics.