BIOMEDICAL INFORMATICS

THE TARSQI TOOLKIT'S RECOGNITION OF TEMPORAL EXPRESSIONS WITHIN MEDICAL DOCUMENTS

Ferdo Renardi Ong

Thesis under the direction of Professor Steven Brown

To diagnose and treat patients, clinicians concern themselves with a complex set of events. They need to know when, how long, and in what sequence certain events occur. An automated means of extracting temporal meaning in electronic medical records has been particularly challenging in natural language processing. Currently the Tarsqi Toolkit (TTK) is the only complete software package (open source) freely available for the temporal ordering of events within narrative free text documents. This project focused on the TTK's ability to recognize temporal expressions within Veterans Affairs electronic medical documents. A baseline evaluation of TTK's performance on the Timebank v1.2 corpora of 183 news articles and a set of 100 VA hospital admission and discharge notes showed an F-measure of 0.53 and 0.15, respectively. Project development included the correction of missed and partial recognition of temporal expressions, and the expansion of its coverage of time expressions for medical documents. Post-modification, the TTK achieved an F-measure of 0.71 on a different set of 100 VA hospital admission and discharge notes. Future work will evaluate TTK's recognition of temporal expressions within additional sets of medical documents.

Approved ____ Date ______________

THE TARSQI TOOLKIT'S RECOGNITION OF TEMPORAL EXPRESSIONS WITHIN MEDICAL DOCUMENTS

By

FERDO RENARDI ONG

Thesis

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

MASTER OF SCIENCE

in

Biomedical Informatics

August, 2010

Nashville, Tennessee

Approved:

Professor Steven H. Brown

Professor Dominik Aronsky

Professor Joshua C. Denny

ACKNOWLEDGEMENTS

I would like to thank my thesis committee for their guidance and support of my work. I am grateful for my advisor, Dr. Steven Brown, for his mentorship, expertise, and friendship. He has directed my work over the past two years and has helped me develop my research interests in natural language processing (NLP) and time. I thank Dr. Dominik Aronsky for the pieces of advice he has given me over the years regarding academics and life in general. I also thank Dr. Joshua Denny for his genuine interest with my work and for sharing his own knowledge and experiences dealing with NLP and time

I am indebted to my colleagues in Veterans Affairs (VA). In particular, I thank Drs. Theodore Speroff and Ruth Reeves. Dr. Speroff is a master of research methods, and has helped me tremendously in organizing the steps needed for this project. Dr. Reeves is an expert linguist who shares the same interests about time. I have learned much from her and was very fortunate to be able to work side by side with her as co-annotators on this project. Additionally, I thank Dr. Reeves for proofreading and editing this manuscript. Without the funding from the VA Medical Informatics Fellowship, my work would not have been possible.

I would also like to thank the Vanderbilt University Department of Biomedical Informatics. I appreciate the support of the faculty, staff, and students. They are wonderful people to work with and to learn from.

Finally, I am grateful for my family: my wife Michelle; my children David, Sarah, Jonathan, and Laura; and my parents Eddie and Lianne. I am very fortunate to have all them in my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AQUAINT .... Advanced Question Answering for Intelligence

ARDA .... Advanced Research Development Agency

CLEF .... Clinical e-Science Framework

DC .... Discharge

H&P .... History & Physical

IAA .... Inter-annotator agreement

IARPA .... Intelligence Advanced Research Projects Activity

IC .... Intelligence community

ISO .... International Standards Organization

MCVS .... Multi-threaded Clinical Vocabulary Server

Mod-TTK .... Modified TTK

NLP .... Natural Language Processing

POEM .... Post-operative event monitoring

TARSQI .... Temporal Awareness and Reasoning Systems for Question Interpretation

TTK .... Tarsqi Toolkit

TimeML .... Time mark-up language

Timex .... Time expression

TVHS .... Tennessee Valley Healthcare System

VA .... Veterans Affairs

VISN .... Veterans Integrated Service Network

XML .... Extensible Markup Language

CHAPTER I

INTRODUCTION

With the ever increasing volume of electronic patient records, the need has grown for automated extraction of information. The community of health care workers and information engineers has shown great interest in accomplishing information extraction through various methods in natural language processing (NLP). A key challenge to NLP is the extraction of temporal meaning from narrative text. This project investigated the feasibility of using a promising tool for just such extraction. The Time Markup-Language (TimeML) is an XML-based markup language for encoding temporal and event time information for use in automatic text processing. [1] It was developed under the auspices of the Advanced Research Development Agency (ARDA) funded Advanced Question Answering for Intelligence (AQUAINT) program. [2] ARDA (later called the Disruptive Technology Office and now part of the Intelligence Advanced Research Projects Activity - IARPA) was a "U.S. Government entity that sponsored and promoted IT research and development for the intelligence community (IC), which included but was not limited to the Central Intelligence Agency, Defense Intelligence Agency, National Security Agency, National Imagery and Mapping Agency, and National Reconnaissance Office." [3] A team of computational linguists interested in temporal reasoning, (including James Pustejovsky and Marc Verhagen of Brandeis University, and Inderjeet Mani of Georgetown University), built the Temporal Awareness and Reasoning Systems for Question Interpretation (TARSQI) toolkit (TTK) software package. [4, 5] Building on research that grew out of TimeML specifications, this team developed TARSQI for the automated tagging and normalization of temporal expressions, the tagging of events, and the anchoring and ordering of events within general texts. Currently the TTK is the only complete software package (open source) freely available for the temporal ordering of events within narrative free text documents. This study evaluated the applicability of TimeML and TTK version 1.0 for the identification of time expressions within medical narratives.

TimeML developers had four primary goals: time stamping of events, ordering of events, contextual reasoning of “underspecified time expressions”, and reasoning about the duration of events. [1] This study evaluated the recognition capabilities of TTK with respect to temporal expressions in medical documents. Future work will explore the TTK’s ability to recognize medical concepts and events, and link them with TTK identified contextual temporal expressions.

Chapter II provides some background on the distribution and interpretation of temporal expressions in narrative documents and the role of natural language processing (NLP) in extracting temporal information from electronic documents. Chapter III introduces the reader to TimeML and the TTK. Chapter IV describes a pilot study evaluating the appropriateness of TimeML’s model of time for a set of Veteran Affairs (VA) hospital History & Physical (H&P) notes and Discharge (DC) summaries. Chapter V then describes a preliminary study and baseline evaluation of TTK’s ability to recognize temporal expressions within Timebank v1.2 (a corpus of 183 news articles), and a set of 100 random VA H&P and DC notes.

My colleagues and I determined that for medical documents, TTK version 1.0 automated extraction of time expressions compared with human reviewers, achieved a recall, precision, and F-measure of 0.11, 0.21, and 0.15 respectively. From this baseline evaluation, I developed methods for improving the recognition of temporal expressions found in medical records. A selection of shortfalls of the TTK motivated the methods of improvement I chose. The most notable shortfalls include the lack of or partial recognition of many specially formatted (absolute) temporal expression, the lack of recognition for many medical abbreviations for frequency/periodicity (sets), and the incorrect classification (date, time, duration, set) of numerous temporal expressions. Chapter VI discusses the modifications to the source code that correct missed or partially recognized common time expressions found within the VA medical documents. Chapter VII re-evaluates the modified TTK's ability to recognize temporal expressions for a subsequent set of 100 random VA H&P and DC notes. The modified TTK shows an improved recall, precision, and F-measure of 0.70, 0.71, and 0.71 respectively. The final chapter of this manuscript discusses the limitations of this study and suggests possible future work to be done on the TTK.

CHAPTER II

BACKGROUND

The importance of time in clinical medicine

As noted by Hurwitz, “Neither seen nor felt directly, nor heard, tasted or smelt, time is dimensional to being and inherent in the practice of medicine. Clinical encounters focus typically on temporal sequences, on relations of before and after, on discussions of beginnings and endings.” [6] Shahar and Combi remarked, “It is almost impossible to try to represent and analyze clinical data in the absence of a temporal dimension.” [7, 8] For the effective diagnosis and treatment of patients, clinicians must pay close attention to the onset, evolution, and duration of signs and symptoms; the timing of diagnostic tests and treatment modalities; and the progression of patients’ health and disease over time.

To make correct diagnoses, clinicians need to know when, how long, and in what sequence clinically relevant events occur. As an example,

Case 1: Mr. Smith <u>fell</u> then had <u>chest pain</u>.

Case 2: Mr. Smith had <u>chest pain</u> then <u>fell</u>.

Given the events alone (chest pain and fell), absent any knowledge of the timing or sequence of the events, or any additional information, the correct diagnosis is difficult or impossible to attain. In the first case, Mr. Smith may have tripped, fallen, and then fractured his ribs due to the fall. In the second case, Mr. Smith may have had a myocardial infarction (heart attack) that resulted in a dangerous arrhythmia causing him to collapse and fall. The two cases imply very different diagnoses, treatment modalities, and prognoses.

To make use of textual information, both the temporal data and the intent of the author must be uncovered. The sequence of events within the narrative text may give clues to cause and effect relationships between them. Often in narrative texts, the relationship between events is discursive, and not straightforwardly drawn.

The Increasing Use of Computers

Due to the ever increasing volume of electronic patient records available to physicians, computer information systems are increasingly important to patient care. The temporal properties relevant to the storage and retrieval of clinical data as well as temporal reasoning about time-related clinical data within databases have been the focus of a good deal of technological effort. Clearly stated by Shahar and Combi, “Representing, maintaining, querying, and reasoning about time-oriented data are a major theoretical and practical research area in medical informatics.” [8] Over two decades, more than 2000 papers on temporal databases have been published. [9] There is continual research concerned with processing temporal data within structured and coded data.

Informatics research often must achieve several goals, and the work presented here is no exception. As a natural language processing thesis, the goals include both explaining the nature of the linguistic data involved and a method for leveraging the knowledge about that data into a form that is computationally tractable. The goals I discuss here, therefore, are interpretive, operational and methodological. These goals inter-relate and it is often necessary to relate the subtasks of one goal to those of another.

The purpose of this background chapter is to lay out the terrain within the domain of each goal. To provide a useful background in the domain of temporal information, I define here a few terms that are specific to these goals.

The interpretive tasks of this research involve providing the human interpretation of the parts of natural language that pertain to temporal reasoning. The data I set forth includes temporal expressions, their values, and the communicative purposes involved in the use of these expressions. I describe here the expressions used to refer to various aspects of time. I provide a typology of temporal expressions and of their interpretations. Textual data that mention time are called *temporal expressions*. The referential assignments to these expressions are called *temporal values*. Temporal information can be expressed in degrees of granularity – from epochs to nanoseconds. The granularity expressed or required is a function of communicative purposes. Communicative purposes also determine whether the term a speaker deploys expresses temporal information that is dependent on contextual information or whether the term is a stand-alone reference to a particular time or date. An *absolute temporal expression* is straightforwardly assigned a temporal value based on the expression meaning itself, whereas a *relative temporal expression* is dependent on some other element for temporal value assignment. *Specification requirements* on temporal expressions are conditions on how much information is required in order to assign temporal values to them. These considerations involve both granularity and the richness of contextual information surrounding the temporal expressions in question.

The operational goals of this work require rendering the data into a form that is machine readable and machine actionable. The initial operational goal is to recognize the temporal expressions that occur in medical documents, and to classify them according to the appropriate category for their value type. Categorizing the expressions, in all their variability, aids in the recognition task, the baseline goal of data collection. Temporal expressions are both varied and ambiguous. Varied because there are many ways to express a certain value, and ambiguous because an expression may be assigned to more than one value. Humans cope well with ambiguity and variation; computers much less so. Therefore a goal is to reduce the variability and ambiguity of the values associated with the data. Operationally, one seeks a standard way to represent the values in one's data set. This is achieved through *normalization*. The temporal values assigned to the temporal expressions are normalized to the standards of International Standards Organization (ISO) 8601 format which I discuss below.

The relationship between sequences of temporal values, and between events that are referred to in medical texts is an inherent part of what temporal information conveys in that context. The aspect of an event that is germane to the discussion here is the set of temporal values that delimit its occurrence, duration or frequency. The communication purposes that a speaker has in using a given temporal expression are varied, but the salient two purposes in medical texts are providing a temporal value for a clinically relevant event, and providing a sequence of medically relevant events. The integrational piece of my interpretive goal is the temporal model. In discussing Allen's and Vilain's theories of temporal logic, I discuss a framework for how all of these elements fit together -- the overall organization of temporal values, the relations between them, and how they operate to satisfy the communicative needs of natural language speakers.

The role for Natural Language Processing

Clinicians, researchers, informaticists, and a variety of healthcare workers have fruitfully deployed natural language processing (NLP) to unlock valuable information within medical textual reports. By using natural language processing on medical free text, such healthcare information technologists can extract voluminous amounts of clinical information from textual reports and transform these data into more computable or human understandable formats. NLP systems parse textual data into appropriate sections, sentences, and words, then process keywords, concepts, and phrases by coding and indexing these terms for use in specified applications such as patient management, decision support, or clinical research. Examples of concept-based indexers in current use within the field of medicine include MEDLEE (developed by Carol Friedman) [10], KnowledgeMap (developed by Josh Denny) [11], and the Multi-threaded Clinical Vocabulary Server (MCVS) (developed by Peter Elkin) [12]. Fewer NLP applications have addressed the informational issues that arise from temporal data that inheres to medical research or practice. Development toward a robust temporal module for healthcare text has not achieved steady progress, and integrating one into a system that recognizes and properly structures medical concepts has remained a technologically high hurdle. [13]

TimeText is one example of an NLP system incorporating time that was developed specifically for medical documents. Recently developed by Zhou, Hripcsak, et al. [14-17], this temporal extraction and reasoning system modeled time as a “Simple Temporal Constraint Satisfaction Problem” where the start and endpoints of time interval and a set of constraints relationships between the time points define the time spans during which the pertinent medical events occur [15, 16]. The system's three-part architecture includes a temporal tagger and annotator; the MEDLEE system for extraction and encoding of events and temporal data; and additional post-processing to resolve temporal expressions and their relationships to one another. [14] The authors evaluated TimeText on a set of twenty hospital discharge summaries. TimeText correctly identified 96.5% of all temporal relationships and 79% of clinically important temporal relationships; and correctly answered 84% of temporal related questions. [17]

Temporal Expressions within clinical documents

Types of Temporal Expressions

According to a widely consulted textbook of NLP, *Speech and Language Processing* by Jurafsky and Martin,

> "Temporal expressions are those that refer to absolute points in time, relative times, durations, and sets of these. Absolute temporal expressions are those that can be mapped directly to calendar dates, times of day, or both. Relative temporal expressions map to particular times through some other reference point (as in "a week from last Tuesday"). Finally, durations denote spans of time at varying levels of granularity (seconds, minutes, days, weeks, centuries, etc.)." [18]

Absolute temporal expressions, relative temporal expressions, and durations singly or in sets are present within medical narratives. Sets are commonly seen in medical documents as frequency information (repetitions of duration expressions), such as "once a week", "every 30 seconds", and "twice daily"; as repetitions of dates or times, such as "every Friday" and "every night"; as lists of dates, such as "he had coronary artery bypass grafting in 1999, 2001, and 2005"; and as time regiments, such as "administer insulin at 8AM, noon, 4PM, and bedtime".

Normalization of temporal expressions

All of these temporal expression types occur with wide variation in format, completeness and distribution over the text of medical documents. One of the subtasks of text recognition is to reduce this variety. This requires choosing a standard representation for each of the value types of all the expression types. This process is known as "normalization". The process of normalization is a base-line step necessary for making computational use of natural language expressions. Normalization reduces the variability in expression types that can correspond to a given value. Thus for the purposes of this study, once the TTK code identifies a temporal expression, the code proceeds to assign a date/time format normalized value to the expression. The TTK normalizes expressions to ISO 8601 format when possible, e.g. 19660202 – expresses a year (4 digits), month (2 digits), and day (2 digits) format for the date February 2, 1966.

Some circumstances require greater specification in temporal information than others. For example airlines often include time zone information along with year month, day, hour and minute values in flight information, but are unlikely to hazard second and millisecond specification. In other situations, such as reports on physiological reaction times, this level of granularity is appropriate. Thus the specification requirements for a temporal expression depend on its intended use. Generally, a time expression is fully specified for its particular usage if it overtly designates the year, month and day. There are however numerous situations of use that also require time of day. Unless a time expression is already fully qualified with known “year, month, and day in some conventional form, assigning a temporal value relies on a “temporal anchor” or “reference point” from which time can be calculated. [18] Even with known anchors, temporal expressions are often missing the information needed for full specification. This is generally true of natural language in that much of meaning assignment depends upon implicit or contextualized information. For any information extraction task, it is always the unexpressed, implicit information that is most computationally challenging. Absent a method for retrieving implied information in partially specified temporal expressions, their mapping properties remain underspecified.

For medical narratives, the pertinent temporal anchor is often the date and time of document creation (DCT). Even so, absolute temporal expressions in such documents frequently cannot be fully specified because they occur with insufficient granularity. Most often these are year dates without month or day units. Another typical data case occurs with a document containing a phrase such as “in June”. Here, the year can usually be inferred by the document creation date, but exact day or days in June cannot be determined. Relative temporal expressions, as this example shows, require a reference point or anchor to satisfy full specification. Within a clinical note, “yesterday”, “today” or “last Tuesday” have little temporal meaning unless the document’s creation date/time is also known. Times or events mentioned within the text, e.g. “one week after surgery”, can also serve as temporal anchors. In this example, to calculate the time under consideration, one needs to establish when the temporal anchor “surgery” occurred. The time specification for such expressions depends on the availability and completeness of the temporal information concerning the event anchor. Generally, relative temporal expressions possess a degree of uncertainty, particularly with phrases such as “this morning”, “last summer”, and “in the past” where one or both endpoints of such time intervals are unclear (fuzzy). Durations denote a span of time, and like relative temporal expressions, a duration, such as “for two weeks”, cannot be mapped to a timeline unless it is anchored at its start point, end point, or both, e.g. “for two weeks beginning now”. In this example, the expression "now" is itself reliant on the anchor provided by utterance time or document creation time.

Intervals or points in time

Time can be represented as an instant (point) in time or an interval of time with start and end points. There are circumstances in which it is more convenient to express the temporal information as a point and others in which an interval is more convenient. The two representations are, however, notational variants. Two points in time and an ordering relation between them define an interval. The reference value of the temporal points is by convention ordered from earliest to latest, so that a start point of an interval has a temporal reference that is smaller in value than that of the end point. The reader can verify that interval representation and point representation are notational variants by observing that an interval whose start point and endpoint are the same is indistinguishable from a single time point. Using a timeline as a basic type of temporal model, one can visualize an interval as a segment along the time line where length corresponds to the duration of the time segment. The utility of the timeline as a temporal model of sorts arises in particular when considering how to link an event to the time it occurs in. Once the interval is delimited, the place of the event along the timeline is set. A series of such events sequenced by the ordering of event intervals is the basic procedure of a clinical narrative. Mapping an interval onto a timeline is a time-theoretic procedure. The units of time making up the points along the line are a prerequisite; additionally, the mapping requires knowledge of the two endpoints, or knowledge of one endpoint and the duration (length) of the interval. The commonplace notion of a timeline is nonetheless a useful one in conceptualizing the relationship between temporal values and temporal sequences.

Laboratory databases often contain only lab results and the “time points” (specified date and time) at which they were completed. In contrast, medical textual documents commonly contain both basic element types: points and intervals. Within medical narratives, however, the two endpoints of an interval are rarely explicit, e.g. “yesterday morning”, “he will take antibiotics for two weeks”, and “she’s had heartburn since last year”.

Lexical Triggers

Temporal expressions can contain or accompany “lexical triggers”, which are pieces of language that orient temporal expressions to a timeline. Examples include nouns such as “the past”, “day”, “summer” and “New Year’s eve”; adjectives such as “daily”, “current”, and “long-term”; adverbs such as “ago”; and specialized time patterns such as “12/25/2008”, “1994”, and “1:30”. Figure 1 from Ferro et al. gives further examples. [19]

Noun	*minute, afternoon, midnight, day, night, weekend, month, summer, season, quarter, year, decade, century, millennium, era, semester, the future, future, the past, past, time, period, point*
Proper Name	*Monday, January,* *New Year's Eve, Washington's Birthday, Solstice* triggers that function as temporal modifiers within titles (such as "The Coach of the *Year*")
Specialized Time Patterns	*8:00, 12/2/00, 1994, 1960's*
Adjective	*recent, former, current, future, past, daily, monthly, biannual, daytime, daylong, onetime, ago, pre-season, short-term*
Adverb	*currently, lately, hourly, daily, monthly, age*
Time noun/adverb	*now, today, yesterday, tomorrow*
Number	*3* (as in *"He arrived at 3"), three, fifth* (as in *"the fifth of June"*), *Sixties* (as in referring to the decade *"the Sixties"*)

Figure 1 A sample list of Temporal Lexical Triggers and their Part of Speech from Ferro et. al. [19]

When an expression is fully specified with year, month, day, and possibly hour and minute, its position, and thus its order, within a timeline becomes obvious. More often, temporal expressions do not give sufficient detail to determine their exact positions within a timeline. Even so, it may still be possible to determine the temporal sequence referenced by such expressions and ultimately the events they modify.

Classifications of Temporal Sequences: the Allen and Vilain Time Models

James Allen modeled time according to intervals. In his model, two intervals are related to one another by one of 13 relationship categories (see Figure 2). [20-22] If a third interval is related to any one of the previous two, the relationship of this third interval to the remaining interval can be determined through transitivity (Figure 4 shows Allen's transitivity table for time intervals). Such reasoning permits the ordering of events.

Thus, given that interval A has relation r1 with interval B; and interval B has relation r2 with interval C; then the relation between A and C is constrained by the transitive closure for r1 and r2. In a simple example, given that A is before B and B is before C, then A is before C. New relational constraints can then be propagated to other intervals through transitivity. An advantage of Allen's constraint propagation algorithm is its execution within polynomial time. [20, 21]

Vilain extended Allen's notion of time to include points with intervals. In addition to the 13 interval relations identified by Allen, Vilain identified 3 point to point relationships and 10 point to interval relationships. (see Figure 3). Together, the Allen-Vilain model includes 26 categories for classification of temporal relationships. [22]

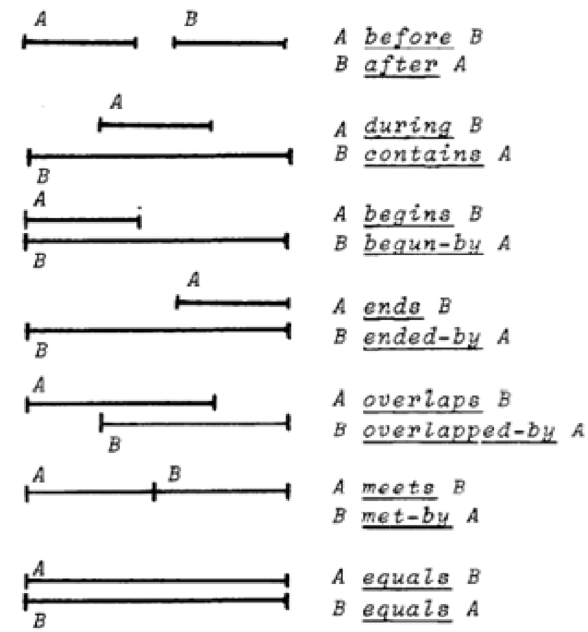


Figure 1: Primitive relations between intervals

Figure 2 – Allen's 13 possible temporal relations. (Modified From Vilain) [22]

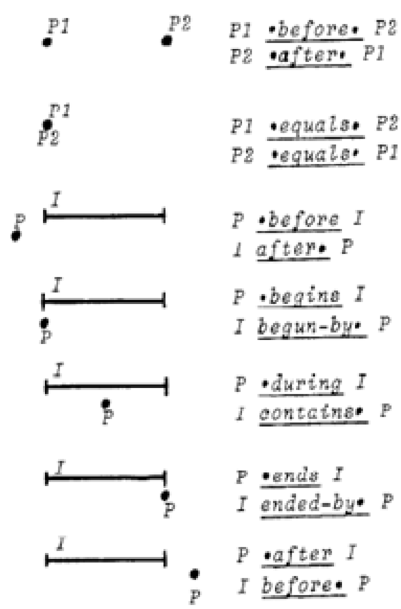


Figure 4: New primitive relations (involving points)

Figure 3 – Additional point-point and point-interval relations.
(From Vilain) [22]

B r2 C / A r1 B	<	>	d	di	o	oi	m	mi	s	si	f	fi
"before" <	<	no info	< o m d s	<	<	< o m d s	<	< o m d s	<	<	< o m d s	<
"after" >	no info	>	> oi mi d f	>	> oi mi d f	>	> oi mi d f	>	> oi mi d f	>	>	>
"during" d	<	>	d	no info	< o m d s	> oi mi d f	<	>	d	> oi mi d f	d	< o m d s
"contains" di	< o m di fi	> oi di mi si	o oi dur con =	di	o di fi	oi di si	o di fi	oi di si	di fi o	di	di si oi	di
"overlaps" o	<	> oi di mi si	o d s	< o m di fi	< o m	o oi dur con =	<	oi di si	o	di fi o	d s o	< o m
"over-lapped-by" oi	< o m di fi	>	oi d f	> oi mi di si	o oi dur con =	> oi mi	o di fi	>	oi d f	oi > mi	oi	oi di si
"meets" m	<	> oi mi di si	o d s	<	<	o d s	<	f fi =	m	m	d s o	<
"met-by" mi	< o m di fi	>	oi d f	>	oi d f	>	s si =	>	d f oi	>	mi	mi
"starts" s	<	>	d	< o m di fi	< o m	oi d f	<	mi	s	s si =	d	< m o
"started by" si	< o m di fi	>	oi d f	di	o di fi	oi	o di fi	mi	s si =	si	oi	di
"finishes" f	<	>	d	> oi mi di si	o d s	> oi mi	m	>	d	> oi mi	f	f fi =
"finished-by" fi	<	> oi mi di si	o d s	di	o	oi di si	m	si oi di	o	di	f fi =	fi

FIGURE 4. The Transitivity Table for the Twelve Temporal Relations (omitting "=").

Figure 4 Allen's Transitivity Table for his 12 Interval based Temporal Relations (omitting "=") [20]

Summary

Clinicians use time data extensively in the diagnosis and treatment of their patients. With the increasing use of electronic medical record systems and the fact that most patient data are in free text narratives, the importance of natural language processing has increased. Creating an automated means for extracting the temporal meaning surrounding the concepts and events in electronic records has proved thus far to be singularly challenging for natural language processing.

One could argue that all events within narrative texts are linked to temporal values. For the vast majority of events expressed in a text, the temporal meaning is implicit rather than explicit. By way of background, Chapter II provides a taxonomy of temporal expressions observed within general and medical texts. Once the data collection step of expression recognition is satisfactorily carried out, the goal is to normalize the expression. From this value if possible, the operational goal is to obtain the position this value determines within a timeline, or obtain its order (sequence) with respect to other time expressions and event temporal values. TimeML and the TTK create a framework whereby a computational system can provide the time-stamping and ordering of events in a structured, standardized fashion.

CHAPTER III

TIMEML AND THE TARSQI TOOLKIT

TimeML

TimeML, the time markup language, is an XML-based markup language for encoding temporal and event-to-time relations for use in automatic text processing. As mentioned previously, the Advanced Research Development Agency - ARDA, now part of the Intelligence Advanced Research Projects Activity - IARPA, funded its development.

Although used more extensively in annotating news articles, TimeML has also been used to extract time representations from e-mails, legal documents, and medical texts. [23] The Clinical e-Science Framework (CLEF) project proposed incorporating TimeML into their Patient Chronicle semi-automatic pipeline system. [24] Their principle challenge was in “improving event recognition”. [25] The United Kingdom Medical Research Council sponsored this work so as to “manage repositories of clinical patient data for the purpose of biomedical research and clinical care”.

TimeML developers had four primary goals: time stamping events, ordering events, contextual reasoning of “underspecified time expressions”, and reasoning about the duration of events. [1] Thus, the 3 building blocks for TimeML are times, events, and the temporal relationships between time and events. TimeML uses four different XML tag types: TIMEX3; EVENT; SIGNAL; and LINK:

TIMEX3 tags

TIMEX3 is based on the TIMEX2 time annotation schema.

TIMEX2 focuses on the recognition and normalization of temporal expressions. The TIMEX2 XML-based standard for annotation, tagging, and extraction of temporal expressions was developed as part of the Translingual Information Detection, Extraction, and Summarization (TIDES) research program and the Automatic Content Extraction (ACE) program supported by the Defense Advanced Research Projects Agency - DARPA. This annotation scheme identifies and then converts dates and times to International Standards Organization 8601 numerical representation of dates and times when possible. The "annotation scheme takes into account characteristic properties of temporal expressions in natural language, namely indexicality, granularity, context-dependence, fuzziness of boundaries, and ambiguity", extending ISO 8601 in various ways. [19]

Examples of TIMEX2 tags include:

1) <TIMEX2 VAL="1994-01-21T08:29:05">January 21, 1994 08:29:05 EST</TIMEX2>
2) <TIMEX2 VAL="1998-FA">Fall 1998</TIMEX2>
3) <TIMEX2 VAL="1999-07-14TNI"> last night</TIMEX2>, given a reference of July 15, 1999
4) <TIMEX2 VAL="1999-07-10">five days ago</TIMEX2>, given a reference of July 15, 1999
5) <TIMEX2 VAL="1999-W29">next week</TIMEX2>, given a reference of July 15, 1999 [19]

An automatic tagger for TIMEX2 expressions is downloadable free of charge from mitre.org at **http://timex2.mitre.org/taggers/timex2_taggers.html**. The computer algorithm uses a rule-based approach, whereby a series of Perl regular expressions identifies specific text string patterns for temporal expressions.

The following is an example of code from the TIMEX2 Perl module TempEx.pm version 1.05_03, written by George Wilson:

```
$string =~
    s/($OT+\'$CT+$OT+\d\d$CT+)/<TIMEX2 TYPE=\"DATE\">$1<\/TIMEX2>/gsio;
```

In this example, $OT and $CT represent opening and closing part of speech (POS) tags, respectively. This use of regular expression matching and substitution places TIMEX2 tags around any occurrence of an apostrophe followed by double digits. The tag is TIMEX2 of type date, e.g. <TIMEX2 TYPE=”DATE”>’99</TIMEX2>.

TimeML developers created TIMEX3 so as to accommodate both specified and under-specified temporal expressions into their annotation schema. TIMEX3 development arose from work on TIMEX2. According to TimeML guidelines, TIMEX3 captures four types of temporal expressions: DATE (for calendar date); TIME (for time of day); DURATION; and SET (for reoccurring times). [26]

Examples of TIMEX3 tags are:

1) <TIMEX3 tid="t1" type="DATE" value="2004-11-2">November 22, 2004</TIMEX3>

2) <TIMEX3 tid="t2" type="DURATION" value="P3D">three days</TIMEX3>

3) <TIMEX3 tid="t3" type="SET" value="P1W" quant="EACH" freq="3D">3 days each week </TIMEX3>

where tid is the time ID number of the expression, and value holds the normalized ISO 8601 format of the temporal expression if calculable. X placeholders are used for expressions which cannot be fully specified, e.g. "XXXX-02-02" for February 2 and no year.

TIMEX3 introduced the notion of "temporal anchor" to accommodate expressions whose temporal reference cannot be retrieved by evaluating the expressions themselves. The anchor then provides the temporal reference for such underspecified expressions via an anchor time ID. Other expressions, such as event or date/time expressions, may serve as anchors.

EVENT tags

EVENT tags are divided into seven classes: reporting, perception, aspectual, I_action, I_state, state, and occurrence. [26] These classes primarily represent different verbs seen within sentences. Multiple instances of events are tagged by MAKEINSTANCE tags.

SIGNAL tags

SIGNAL tags identify temporal prepositions and conjunctions such as *before, after, during, since, until, at, on, in, for, over, throughout, while*, and *when*. Such terms act to place temporal expressions and the events associated with them on a timeline.
An example of SIGNAL tagging for "on November 22, 2004" is:

<SIGNAL sid="s1">on</SIGNAL> <TIMEX3 tid="t1" type="DATE" value="1966-02-02"> February 2, 1966</TIMEX3>

LINK tags

TimeML has 3 LINK tags: TLINK; SLINK; AND ALINK, all of which provide linking or relational information. SLINK and ALINK tags convey subordinating and aspectual relationships, respectively, between events. "TLINK [Temporal LINK] tags are arguably the most important tag in all of TimeML". [26] The purpose of the tag is to provide a classifier which relates two intervals of time to one another. These relations determine the anchoring and ordering of events within a narrative's timeline. TLINK's rule set takes into account several syntactic rules based on intra-sentential presence of SIGNAL tags, event types, verbs, and verb tenses. The relation types between time intervals and events are similar to Allen's 13 interval relationships: *simultaneous; before; after; immediately before; immediately after; including; being included; during; beginning; begun by; ending; ended by;* and *identity.* The TTK uses constraint propagation to ensure the consistency of all relations. [4, 27, 28]

The Tarsqi Toolkit (TTK)

Members of the TimeML development team created the Tarsqi toolkit software package (also referred to as the Temporal Awareness and Reasoning Systems for Question Interpretation Toolkit or TTK) based on TimeML specifications.

The TTK contains a number of modules including Treetagger (the part of speech preprocessor), GUTime (the time-tagger), and Evita (the event recognition system). In addition, the scripts incorporated within TTK's temporal processing module (e.g. Slinket, Blinker, S2T, TLink Classifier, SputLink, and Link Merger) generate links between times and events, and then propagate links based on Allen's algorithm for transitive closure. Figure 5 shows a diagram of the TTK architecture from Verhagen and Pustejovsky [28]. Again, the time-tagger is based on TIMEX3, a variation of TIMEX2. [29] Hence, the GUTime algorithm is also rule-based, running on a 2001 version of TempEx.pm. "GUTime has been benchmarked on training data from the Time Expression Recognition and Normalization task (timex2.mitre.org/tern.html) at .85, .78, and .82 F-measure for timex2, text, and val fields respectively." [30]

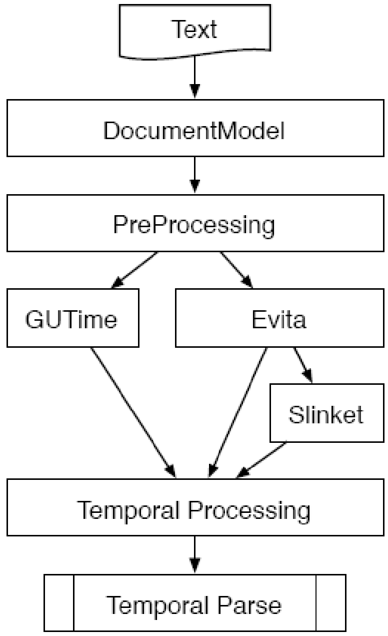


Figure 5. The TTK Architecture [27]

For temporal ordering, TTK is currently the only “complete toolkit that is freely available for use”. [31] TimeML’s TARSQI toolkit is downloadable free of charge from its developers at http://timeml.org/site/tarsqi/toolkit/download.html

Figure 6 shows an example of a sentence processed by TTK. Temporal expressions (t#) are highlighted in blue, events (e#) are highlighted in red, and links are listed on the right.

Mr . Smith had_e1 a cardiac catheterization today_t1 after being admitted_e2 for syncope Tuesday_t2 .	includes(t1,e1) includes(t2,e2) before(t2,t1)

Figure 6. Sample output from TTK showing time expressions, events, and links.

Discussion

The Time Markup-Language, TimeML, specifies the types of temporal expressions and events to annotate within documents and what XML tags to use. Based on TimeML specification, the TTK software package was created to perform the following steps:

1. Identify temporal expressions
2. Place numerical values to temporal expressions
3. Identify events
4. Link times and events together

This thesis project focuses on how well the TTK performs Step 1.

Step 2, valuation, depends heavily on the accuracy of Step 1.

The identification of temporal expressions (Step 1) entails:

1. Recognizing patterns of temporal expressions through the use of a series of perl regular expressions
2. Identifying the type of the temporal expression found. As per TimeML (TIMEX3) specifications, the four types are:
    - A. DATE (for calendar date)
    - B. TIME (for time of day)
    - C. DURATION (for spans of time)
    - D. SET (for recurring times)

CHAPTER IV

A PRELIMINARY EVALUATION OF TEMPORAL EXPRESSIONS IN VA MEDICAL DOCUMENTS

Aims

1. Manually identify and classify time expressions found in medical documents according to TimeML specifications and annotation guidelines [32,33]
2. Determine any obvious gaps with TimeML's coverage of time expressions.

Methods

Sample: This author manually reviewed and annotated 50 random History and Physical (H&P) notes and 50 random Discharge (DC) summaries for temporal expressions.

Eligible records included those of patients admitted to the Department of Veterans Affairs (VA) Tennessee Valley Healthcare System (TVHS) – Nashville, Tennessee Campus (Site 6 VISN 9) from 1999 to 2006. This author constructed Perl scripts to randomly extract History and Physical (H&P) notes and hospital Discharge (DC) summaries from the VA POEM (Post-operative event monitoring) research database. For H&P notes, the script looked specifically for the string "STANDARD TITLE: PHYSICIAN H & P NOTE" within the header of the note for inclusion. For DC summaries, the script looked specifically for the string "STANDARD TITLE: DISCHARGE SUMMARY" within the header of the note for inclusion.

Procedures: Starting with the list of sample lexical triggers identifying temporal expressions from Ferro et al. [19] and Mandel et al. [34] (Figure 1, p.9), this author adapted TimeML annotation guidelines from Sauri et.al. [33] to identify date, time, duration, and set expressions.

DATE (for calendar date):

1. Included specialized time patterns such as “12/1/2008”, “10-01”, “3 PM”, “0400”, and “02/2009”.
2. Included proper nouns such as days of the week (“Monday”, “Tuesday”, “Wed”); months of the year (“January”, “Feb”); holidays (“Christmas”, “Groundhog Day”), and seasons.

TIME (for time of day):

1. Included specialized time patterns such as “3 PM”, “12/1/ 2008 0400”, and “5 o’clock”.
2. Included nouns such as “morning”, “evening”, “afternoon”.

Duration denotes a span of time at varying levels of granularity:

1. Included expressions containing nouns such as “seconds”, “minutes”, “days”, “weeks”, and “years”, denoting a span of time, e.g. “three weeks”, “several days”

Set (for reoccurring times i.e. periodicity):

1. Included expressions containing adverbs and adjectives such as “daily”, “every day”, “per week”, and abbreviations such “qd”, “b.i.d”, and “t.i.d”.

Using a Microsoft Excel spreadsheet, I collected examples of temporal expressions. Expressions which were the same or showed the same pattern (e.g. "Feb 2, 2009" and "Dec 12, 1999") were grouped. The occurrences of the expressions were then counted and categorized according to the date, time, duration, and set criteria.

TimeML annotation guidelines do not include temporal prepositions and connectives (Figure 7) in the tagging of temporal expressions. SIGNAL tags mark this type of information. For this pilot study the author included surrounding temporal prepositions and conjunctions with marked temporal expressions, but did not include such information in the categorization of the expression to date, time, duration and set. The author also included for annotation what he thought were "relevant" temporal expressions, but not part of the annotation guidelines listed above.

Subordinating Conjunction	when, while, as, since, now [that], as long as, as soon as, sooner than, every time, any time
Preposition	at, on, in, for, over, throughout, during, before, after, since, until

Figure 7. A sample list of temporal prepositions and conjunctions from Ferro et al. [19]

Results

Metadata within the header and footer of each H&P and DC summary note supply temporal information such as the document creation date and time (DCT); transcription date and time; and date when the note was electronically signed by the authors. (Figures 8 and 9)

Header

LOCAL TITLE: HISTORY & PHYSICAL – INPATIENT STANDARD TITLE: PHYSICIAN H & P NOTE
DATE OF NOTE: MAR 21, 2007@06:53 ENTRY DATE: MAR 21, 2007@06:53:39
AUTHOR: WHITE, JAMES W. JONES EXP COSIGNER: SAM, HENRY
URGENCY: STATUS: COMPLETED

Footer

/es/ JAMES W. JONES
M.D., Surg. Resident, pager 123-4567
Signed: 03/21/2007 06:57

Figure 8. **An example of an H&P note Header and Footer (names are fictitious)**

Header

LOCAL TITLE: Discharge Summary STANDARD TITLE: DISCHARGE SUMMARY
DICT DATE: JUN 09, 2000@09:47 ENTRY DATE: JUN 10, 2000@09:31:34
DICTATED BY: RED, CHARLES H ATTENDING: JONES, JOHN L
URGENCY: routine STATUS: COMPLETED

Footer

Date Transcribed: 06-09-00
Date Dictated: 06-09-00@0947
J: 22222
V: JZ.
aks/ExecuMed

/es/ CHARLES H RED
M.D., PGY-1, pager 1234
Signed: 06/16/2000 06:11

/es/ JOHN L JONES
M.D., Associate Chief, Surgical Service

Figure 9. **An example of a DC note Header and Footer (names are fictitious)**

Including header and footer metadata information, the 100 documents contained 2509 temporal expressions with a mean of 25.09 expressions per document, standard deviation of 13.08, median of 21.5, and a 25-75 interquartile range of 18.

Table 1 shows the occurrences and classifications of the temporal expressions found within the 50 random H&P notes and 50 random DC summaries. There were 736 expressions classified as Date, 530 expressions classified as Time, 239 expressions classified as Duration, and 717 expressions classified as Set.

Within Date and Time, there were 1101 absolute temporal expressions and 165 relative temporal expressions. In addition, there were 287 expressions which did not fit the date, time, duration, and set classifications. 196 of these expressions were event-based expressions which refer to “landmark” clinical events such as hospital admission/discharge/stay (e.g. “prior to admission”, “upon discharge”, “while hospitalized”, etc.), operation/procedure (e.g. “postop day 1”, “until colonoscopy”, “pre-dialysis”, etc.), and clinical encounters (e.g. “prior to his visit”). 76 of the 287 expressions were stand-alone sequential expressions which refer to order and sequence of events (e.g. “prior”, “previously”, “initially”, “subsequently”, “next follow-up”, etc.). 15 remaining expressions included 7 expressions which were stand-alone expressions dealing with onset and frequency of disease and symptoms (e.g. “sudden onset”, “occasional”, “recurrent”, and “chronic”).

Table 1. Occurrences and categorization of the temporal expressions found within the 50 random H&P notes and 50 random DC summaries				
	Absolute Expressions	Relative Expressions	Total	Percent of Total
Date	614	122	736	29.3%
Time	487	43	530	21.1%
Sub-Total	1101	165		
Duration			239	9.5%
Set			717	28.6%
Event-based expressions			196	7.8%
Stand alone sequential expressions			76	3.0%
Misc			15	0.6%

The largest category of temporal expressions was absolute time expressions such as “Dec 05, 2008@19:30:27”; “12/05/2008 19:30”; “12/5/08”; “12-5”; “12-08”. These absolute time expressions comprised 1101 of the 2509 expressions (43.8 % of total expressions) found within the header, body, and footer of documents. Absolute time expressions appeared in more than 52 different formats (Table 2).

Table 2. Examples of absolute time expressions found within 50 random H&P notes and 50 random DC summaries

Absolute time Expressions (pattern)	Occurrences within 100 documents	Absolute time Expressions (pattern)	Occurrences within 100 documents
(12/05)	**7**	**12-5-2008**	**8**
(12/5/2008)	**5**	**1970s**	**2**
(12/05/2008 19:30)	**111**	**2008**	**68**
(2008)	**3**	**Dec 05, 2008@19:30**	**106**
(08)	**1**	**Dec 05, 2008@19:30:27**	**100**
05 December 08	**1**	**Dec 5, 08**	**1**
08	**15**	**Dec 5, 2008**	**23**
12/05 at 7:30pm	**2**	**Dec 5, 2008 19:30**	**1**
12/5/08	**111**	**Dec 08**	**1**
12/05/08@1930	**18**	**Dec 2008**	**2**
12/05/2008 @ 19:30	**3**	**December**	**2**
12/05/2008 19:30	**164**	**December, 2008**	**1**
12/05/2008 at 7:30pm	**1**	**December 2008**	**9**
12'08	**1**	**DEC.2008**	**4**
12/08	**26**	**December 5**	**3**
12/2008	**13**	**December 5, 2008**	**14**
12/5	**17**	**december 5/08**	**1**
12/5 at 7 am	**1**	**December 5th**	**5**
12/5 at 7:30	**1**	**December 5th at 7:30 p.m.**	**2**
12-05-08	**54**	**December 5th, 2008**	**3**
12-05-08@09851	**6**	**December 5, of this year**	**2**
12-05-08@1930	**24**	**December of 2008**	**14**
12-05-2008 at 7:30 p.m.	**1**	**December of this year (2008)**	**1**
12-08	**10**	**December of this year**	**1**
12-5	**10**	**Friday, December 5**	**1**
12-5- 08 at 7:30	**1**	**5th of December**	**1**

Another large category of temporal expressions pertained to sets (periodicity) - frequencies, rates, and repetitions of medications, tests, therapies, or disease processes. Words, phrases, and acronyms such as such as “daily”, “every day”, and “b.i.d,” occurred 717 times of the 2509 expressions (28.6% of total expressions). Set temporal expressions appeared in more than 110 different formats (Table 3).

Table 3. A subset of time expressions representing periodicity (sets) within 50 random H&P notes and 50 random DC summaries

Absolute time Expressions	Occurrences within 100 documents	Absolute time Expressions	Occurrences within 100 documents
q.d.	45	DAILY/daily	16
b.i.d	72	per day	12
t.i.d.	29	per week	2
q.i.d.	4	TWICE A DAY	11
Qd	9	TWICE A DAY BEFORE A MEAL	1
Bid	10	TWICE DAILY	57
Tid	4	TWO TIMES A DAY	21
Qid	3	THREE TIMES A DAY	21
Qhs	8	THREE TIMES A DAY BEFORE MEALS	1
Qac	1	FOUR TIMES A DAY	8
q4	1	AT BEDTIME	36
q8	1	AT NOON	1
EVERY HOUR	1	in the morning	3
EVERY THREE HOURS	2	in the evening	1
EVERY 2 to 4 HOURS	1	in the a.m.	1
EVERY 4 HOURS	5	in the p.m.	1
EVERY 6 HOURS	10	the a.m and the p.m.	1
EVERY SIX HOURS	2	Weekly	1
EVERY 8 HOURS	7	once a week	1
EVERY 12 HOURS	1	q Mondays	1
each day	2	75cc/hr	1
EVERY DAY	90	3-4 days/wk	1
EVERY DAY BEFORE BREAKFAST	7	ONE PER DAY EXCEPT TUESDAYS	1
EVERY MORNING	8	Nightly	1
EVERY EVENING	7	4-5 times/day	1
EVERY WEEK	3	twice a week	1
EVERY 2 WKS.	1	in the AM	2
EVERY MONTH	1	three times a week	1
MONDAY, WEDNESDAY, FRIDAY	1	BEFORE MEALS	1

Discussion

TimeML's model of how temporal terminology maps to time (date, time, duration, and set) is a simple and useful way to categorize the type of data included in this study, accounting for 88.5% of temporal expressions found within this set of medical documents. Date and Time expressions accounted for 50.4% of all expressions. Of these, absolute temporal expressions represented 43.8% of total expressions. They appeared in over 52 different formats. Duration expressions represented 9.5% of all the marked expressions. Time intervals where both endpoints were explicit, e.g. "from 12/03 to 1/04", occurred rarely, only 6 times (0.2% of the total). Sets were common and represented 28.6% of the temporal expressions. The high frequency of such expressions is unsurprising since H&P and DC notes commonly contain long lists of medications and dosage instructions, detailing frequency and duration of drug administration.

About 11% of the marked temporal expressions did not fit into TimeML's model of temporal expressions. A category not directly accounted for within TimeML was expressions which were event-based, referencing time according to important clinical landmarks (events), such as "s/p surgery" and "prior to admission". There were 196 occurrences of this type (7.8% of total expressions). To be assigned temporal reference points, and mapped to a time line, these event-based expressions require the availability of anchor points. Thus many of these expressions remained unspecified. Another data type that the TimeML annotation schema misses is a group of stand-alone expressions that refer to sequence/ordering and frequency/onset of disease/symptoms, such as "subsequently", "chronic diarrhea", "recurrent chest pain". There were 83 occurrences (3.3% of total) found. TimeML developers may have purposely excluded such stand-alone and event-based temporal expressions from their schema of time since these expressions are more difficult to normalize.

Although TimeML recognizes that signals in the form of temporal prepositions and subordinating connectives convey the relationships between temporal expressions and events, a number of adjectives, adverbs, and prefixes such as “ago”, “prior to”,” last”, “following”, “intra-“, “peri-“, “post-“, and “s/p” can also act as signals. Such signals can also help map temporal expressions and events onto an ordering or position within a timeline.

TTK’s GUTime time-tagger normalizes expressions to ISO 8601 formats. Dates, time, and durations are well defined by this standard, though sets as commonly seen in medical documents are less defined. ISO 8601 makes provisions for repeating intervals, e.g. “daily for 5 days” translates to R5P1D, i.e. five repetitions of one day periods (durations). Expressions such as “4 times daily” are not well defined. One can format this expression as R4P6H, though “4 times daily” in medical terminology does not necessarily mean every 6 hours repeated four times. The number of days (repetitions) in which “4 times daily” occurs is often undefined, unknown, or unbounded. The author proposes an additional parameter “Q” (quantity) for set notations to describe the number of times something occurs within a specified duration. Using this convention, the example “4times daily” normalizes to RQ4P1D. No integer follows “R” when the number of repetitions is undefined or unbounded. Alternately, “4 times daily for 5 days” normalizes to R5Q4P1D.

CHAPTER V

A BASELINE EVALUATION OF TTK'S RECOGNITION OF TEMPORAL EXPRESSIONS

Aims

I. Timebank v1.2

1. Compare Timebank v1.2 human annotations to TTK automated annotations of temporal expressions for this corpus of 183 news articles.

II. VA medical documents

1. Identify temporal expressions within a set of H&P and Discharge Summaries
2. Compare human annotations of temporal expressions to TTK automated annotations of temporal expressions.

Methods

Samples

I. Timebank v1.2

TimeML developers also collected and annotated the Timebank v1.2 Corpus. This corpus consisted of 183 news articles from a variety of sources, including Public Radio International (PRI), Voice of America (VOA), ABC, CNN, Associated Press (AP), New York Times (NYT), and the Wall Street Journal (WSJ). Each article was "annotated following the TimeML 1.2.1 specifications" [35]. According to Timebank v1.2 documentation, annotators had "various linguistics backgrounds … intimately familiar with the latest specifications" [36]. The Corpus was available free of charge from the Linguistic Data Consortium (catalog ID LDC2006T08 and ISBN 1-58563-386-0). [37]

The TTK v1.0 software package included 182 of the 183 Timebank news articles within its ttk-1.0/code/data/in/Timebank file directory. This collection was non-annotated, but was in appropriate XML format, ready for input and processing by TTK. The developers excluded document wsj0685 of the Timebank v1.2 Corpus likely because this document did not contain any temporal expressions.

II. VA medical documents

The universe of clinical documentation spanned the six VA healthcare systems across Veterans Integrated Service Network (VISN) 9. A database covering a veteran cohort of 33,000 VISN 9 patients and 12 million medical documents from 1999 through 2006 had been created for ongoing post-operative event monitoring (POEM) research on text processing. The POEM data base included various administrative, structured datasets and various clinical narrative documents, including physician notes and discharge summaries. A random sample of 50 Physician H&P and 50 Discharge Summary notes (different from the set described in Chapter IV) was extracted from the POEM data model using SQL query. Inclusion criteria included file (document) sizes between 2 and 10 kilobytes in length.

Measures: Temporal expressions identified from the documents.

DATE (for calendar date):

1. Included specialized time patterns such as “12/1/2008”, “10-01”, and “02/2009”.
2. Included proper nouns such as days of the week (“Monday”, “Tuesday”, “Wed”); months of the year (“January”, “Feb”); holidays (“Christmas”, “Groundhog Day”), and seasons.

TIME (for time of day):

1. Included specialized time patterns such as “3 PM”, “12/1/ 2008 0400”, and “5 o’clock”.
2. Included nouns such as “morning”, “evening”, “afternoon”.

Duration denotes a span of time at varying levels of granularity:

1. Included expressions containing nouns such as “seconds”, “minutes”, “days”, “weeks”, and “years”, denoting a span of time, e.g. “for three weeks”, “over several days”, “throughout the year”.
2. Did not include expressions such as “<u>in</u> three days”, “<u>in</u> four hours”, “2 weeks <u>ago</u>”, which were categorized as date or time instead.

Set (for reoccurring times i.e. periodicity):

1. Included expressions containing adverbs and adjectives such as “daily”, “every day”, “per week”, and abbreviations such “qd”, “b.i.d”, and “t.i.d".

Procedures

Figure 10 shows the steps taken in evaluating TTK's ability to recognize temporal expressions within Timebank v1.2 and VA medical documents. For each document set, I obtained TTK annotations and then compared and scored them against human annotations of the same documents.

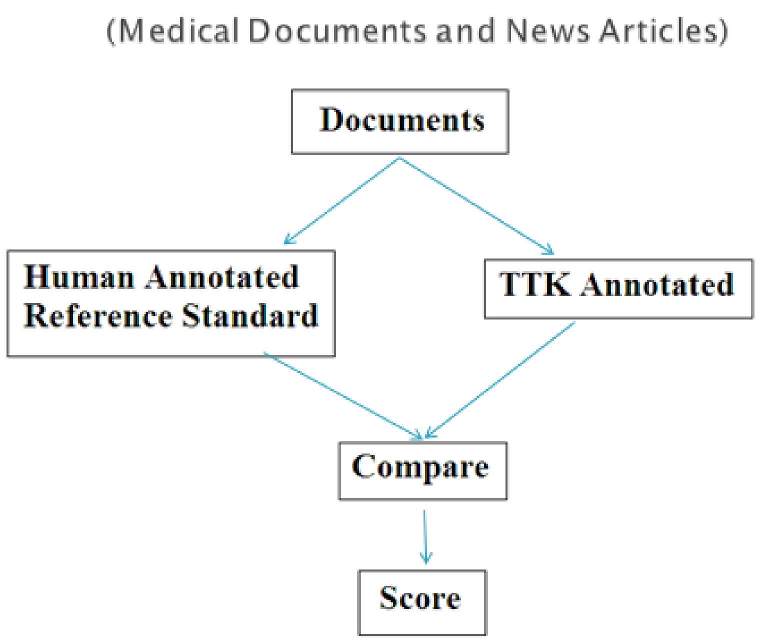


Figure 10. Steps to evaluating TTK's ability to recognize temporal expressions

Step1. Obtain human annotations for Timebank v1.2 and VA medical documents.

A human annotated reference standard consisting of XML tagged TIMEX3 expressions was available for the Timebank v1.2 corpora. In contrast, a reference standard had to be created for the VA medical documents collected for this study. For the VA medical documents, two human reviewers (an M.D. and a Ph.D in Linguistics) identified and annotated temporal expressions using Knowtator v1.8, a general-purpose text annotation tool integrated with the Protégé v3.3.1 knowledge representation system. Both software packages are free for download from http://protege.stanford.edu/ and http://knowtator.sourceforge.net/, respectively. Instructions for installation onto a Microsoft Window XP operating system environment are available at both websites.

This author replicated the TIMEML annotation scheme within Knowtator: the 4 types (DATE, TIME, DURATION, and SET) of temporal expressions specified by TimeML for TIMEX3. Figure 11 shows Knowtator and a document within the middle panel. A temporal expression is highlighted (left click and drag with the mouse), then, within the left panel, the type class it belongs to is chosen (right click then select from a pop-up dialogue box). For each document, reviewers' annotations were stored in XML format, recording string spans, string positions within the document, and string types.

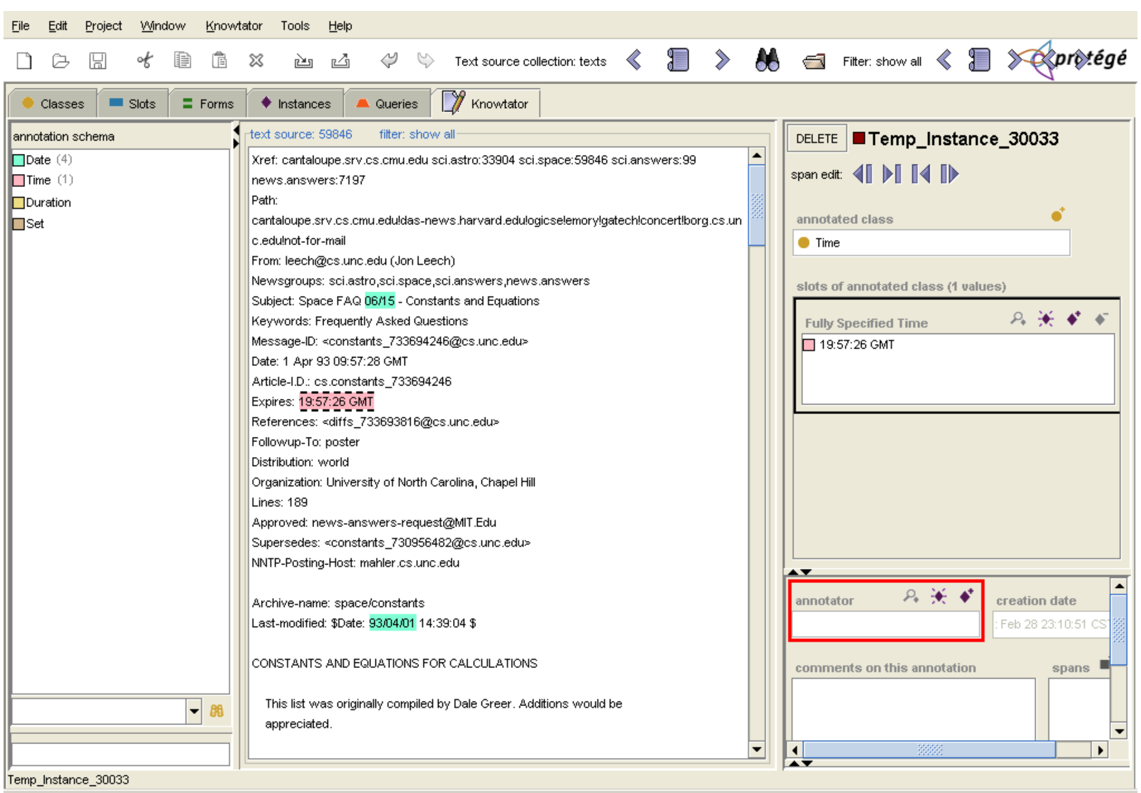


Figure 11. A screenshot of Knowtator v.1.8

Using Knowtator, this author merged document sets from both reviewers and, subsequently, calculated inter-annotator agreement (IAA) for each of the four types of temporal expressions. I performed IAA using "match" defined as an exact span match, and then again using "match" defined as only overlapping spans.

For each type:

Reviewer 2

		pos	neg	
Reviewer 1	pos	a	b	a+b
	neg	c	d	c+d
		a+c	b+d	a+b+c+d

a = number of matches
b + c = number of mismatches

IAA = a / (a+b+c) = number of matches / (number of matches + number of mismatches)
= number of matches / total number of annotations

This author constructed the a reference set for the VA medical notes using a Perl script that looped through the merged annotations from both reviewers and kept annotations that were unique. For non-matching but overlapping annotations, the Perl script selected the annotations with the longest strings. This reference set for the VA medical notes thus consisted of XML files, one for each VA document, representing annotations from both reviewers with string spans, string positions within the document, and string types.

Step2. Obtain TTK annotations for Timebank v1.2 and VA medical documents.

Non-annotated documents from both sets were processed by TTK for the automated recognition of temporal expressions. This author obtained the Tarsqi toolkit v1.0 from http://www.timeml.org/site/tarsqi/toolkit/download.html. TTK's developers developed and tested it for Red Hat Linux 5, with Python 2.4.3 and Perl 5.8.8; and for the Mac OS X, with Python 2.3.5 and Perl 5.8.8. This author installed VMware Workstation 6.5 on a Windows XP host operating system to run a virtual machine with Red Hat Enterprise Linux (RHEL) 5 Desktop as a guest operating system. I configured the virtual network adapter to use Network address translation (NAT) in order to share the internet protocol (ip) address and files of the host computer. I then installed TTK onto the RHEL 5 guest operating system following instructions given by the TimeML website manual http://www.timeml.org/site/tarsqi/toolkit/manual/

For the VA medical documents to be processed by the Tarsqi Control Panel, I converted each medical document to "simple-XML" format by adding "<Doc><Text>" and "</Doc></Text>" at the beginning and ending. Using a Perl script, I modified the documents by changing and removing XML special characters within documents, such as "<", ">", and "&". Using TTK's GUTime time-tagger this author set the automated tagging process into motion to test its performance on the non-annotated Timebank documents (already in appropriate "timebank" XML format) and VA medical documents (in "simple-XML" format).

Figure 12 shows the Tarsqi toolkit control panel which loaded and processed documents according to specific module components desired (modules listed on the left of the screen). Once processed, a list of TIMEXes (time expressions) from each document can be viewed (Figure 13).

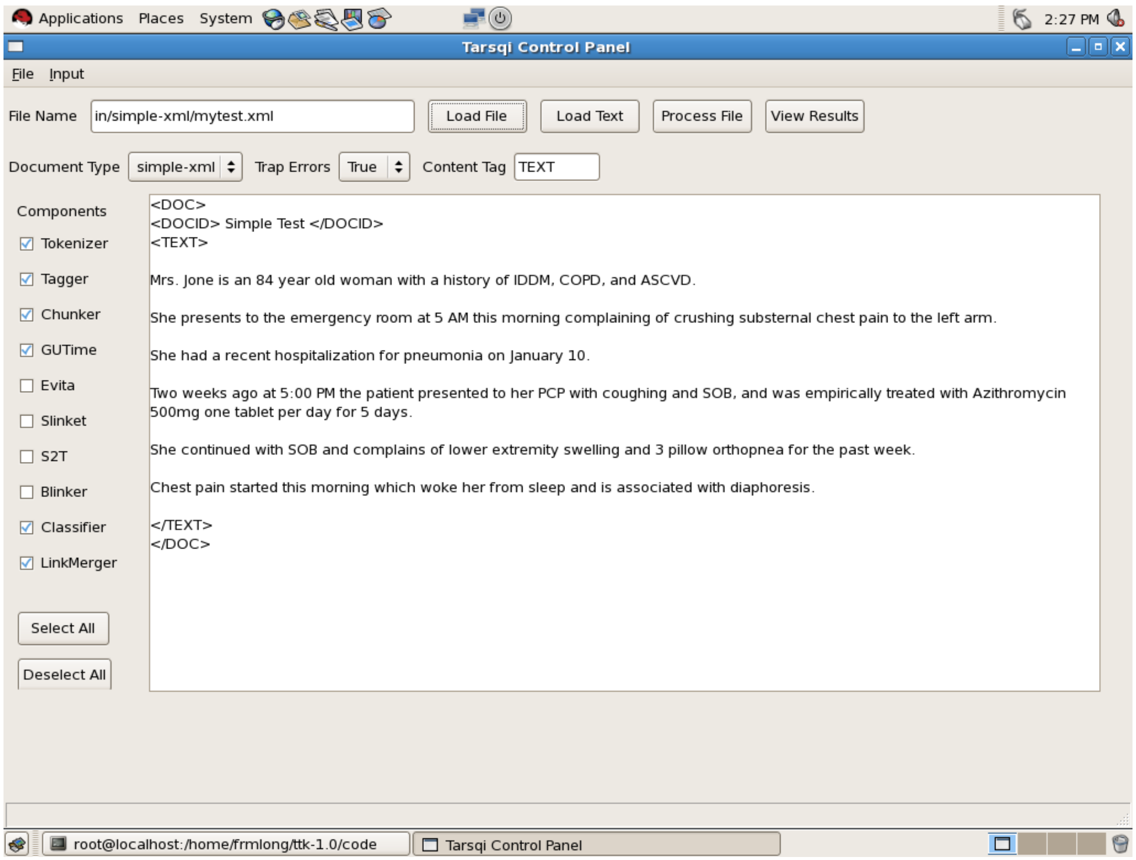
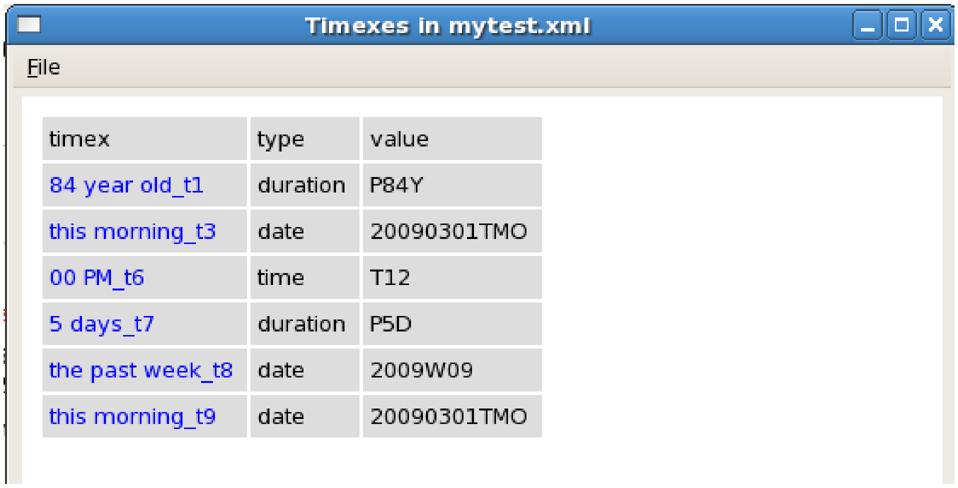


Figure 12. A screenshot of the Tarsqi Toolkit (TTK) control panel



timex	type	value
84 year old_t1	duration	P84Y
this morning_t3	date	20090301TMO
00 PM_t6	time	T12
5 days_t7	duration	P5D
the past week_t8	date	2009W09
this morning_t9	date	20090301TMO

Figure 13. A screenshot viewing a list of the Timexes found within a document

Step3. Match TTK annotations to human annotations.

As can be seen in figure 13, output from TTK included a list of temporal expressions with types, ordered by their appearance within the document. Similar lists were extracted from Timebank's reference set and the VA documents reference set that was created using Knowtator. Unfortunately, string positions identified by Knowtator for the VA reference set and those calculated from Timebank v1.2 did not match string positions calculated from TTK output. This discrepancy resulted from the removal and addition of white spaces during preprocessing tokenizing, part of speech tagging, and chunking performed by TTK. This author thus obtained an ordered list of human annotated time expressions for Timebank v1.2 and VA medical notes for each document belonging in the reference sets and then compared these with a corresponding ordered list of expressions identified by the TTK. I did not include the document creation time (DCT) in the evaluation of Timebank v1.2 articles since TTK required the DCT as part of its input, and consequently did not tag it as part of its output. Table 4 shows an example of actual human and TTK annotation lists found for a single document.

<u>Table 4</u>. VA medical documents: Ordered lists of human and TTK annotations for a single document			
Human annotations	Types	TTK annotations	Types
DEC 25, 1996@14:04	Time	1996	Date
DEC 26, 1996	Date	1996	Date
12-24-96	Date		
57-year-old	Duration	57-year-old	Duration
		the past	Date
8/95	Date		
month	Date	about a month ago	Date
b.i.d	Set		
q.12h	Set		
q. h.s	Set		
b.i.d	Set		
q.a.m	Set		
q.a.m	Set		
at night	Set		
b.i.d	Set		
b.i.d	Set		
q.d	Set		
q.d	Set		
q.d	Set		
t.i.d	Set		
week	Date		
Christmas	Date	Christmas	Date
24-hour	Duration	24-hour	Duration
12-24-96	Date		
Thursday	Date	Thursday	Date
		1404	Date
		351	Date
		2942	Date
that time	date		
12-25-96	date		
12-25-96@1404	time		
12/29/1996	date	1996	Date
02/27/1997	date	1997	Date

I created a Perl script to score TTK performance. This script looped through the lists of expressions found in the reference standard and those found by TTK to match corresponding annotations. For a given human annotated expression, a corresponding TTK annotated expression matched in the following manner:

Category	String (span) match	Type (date, time, duration, set) match
1	exact match	Match
2	exact match	not matched
3	overlap (partial match)	Match
4	overlap (partial match)	not matched
Miss	no match	

I categorized as Spurious, TTK annotated expressions with no matching or overlapping spans in the reference standard. I then counted and saved to a file the expressions fitting each category.

For each type, I followed the scoring used by the Time Expression Recognition and Normalization (TERN) 2004 Plan, available at http://timex2.mitre.org/tern.html and http://timex2.mitre.org/tern_2004/tern_evalplan-2004.29apr04.pdf

1) CORR (Correct) = The two items under comparison were considered identical. (The # of expressions in Category 1)

2) INCO (Incorrect) = The two items under comparison were not identical. (The # of expressions in Category 2 + Category 3 + Category 4)

3) MISS (Missing) = A reference had no TTK output aligned with it. (The # of Missed expressions)

4) SPUR (Spurious) = A TTK output had no reference aligned with it.

(The # of Spurious expressions)

5) POSS (Possible) = The number of items in the reference that contributed to the final score

= CORR + INCO + MISS

6) ACT (Actual) = The number of items in the TTK output:

= CORR + INCO + SPUR

7) REC (Recall) = The # of reference items that was present in the TTK output

= CORR/POSS

8) PREC (Precision) = The # of TTK output items that was actually in the reference

= CORR / ACT

9) F- measure combines recall and precision into one measure

$$\textbf{F-measure} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{(\text{precision} + \text{recall})}$$

10) UND (Undergeneration) = MISS / POSS

11) OVG (Overgeneration) = SPUR / ACT

12) SUB (Substitution) = INCO / CORR + INCO

13) ERR (Error) = MISS / CORR + INCO + SPUR + MISS

Results

I. Timebank v1.2

Table 5 shows the distribution of expressions found for each temporal type in the human annotated reference standard and TTK output for 182 Timebank news articles. Table 6 shows Timebank v1.2 TTK scores against the reference standard.

Table 5. Timebank v1.2 - Occurrences and distribution of each temporal type within the human annotated Timebank v1.2 Corpus and TTK output for the 182 news documents.

	Human		TTK	
Date	1003	81.48%	914	86.14%
Time	41	3.33%	7	0.66%
Duration	175	14.22%	140	13.20%
Set	12	0.97%	0	0%
	1231	100%	1061	100%

Table 6. Timebank v1.2 - TTK scores against human annotations

	Date	Time	Duration	Set	Total
CORR	549	0	57	0	606
INCO	236	27	43	7	313
MISS	238	14	83	5	340
SPUR	94		24	0	118
POSS	1023	41	183	12	1259
ACT	879	27	124	7	1037
REC	0.54	0.00	0.31	0.00	0.48
PREC	0.62	0.00	0.46	0.00	0.58
F-measure	0.58		0.37		0.53
UND	0.23	0.34	0.45	0.42	0.27
OVG	0.11	0.00	0.19	0.00	0.11
SUB	0.30	1.00	0.43	1.00	0.34
ERR	0.21	0.34	0.40	0.42	0.25

Table 7 shows the distribution of incorrect TTK annotations for each temporal type.

Table 8 shows the distribution of incorrect TTK annotations for all types.

Table 7. Timebank v1.2 - Distribution of Incorrect TTK annotations for each temporal type

	Date		Time		Duration		Set	
	# Incorrect	% Incorrect	# Incorrect	% Incorrect	# Incorrect	% Incorrect	# Incorrect	% Incorrect
Exact Span Match Incorrect type	5	2.12%	16	59.26%	4	9.30%	7	100%
Partial span match Correct type	206	87.29%	1	3.70%	26	60.47%	0	0%
Partial span match Incorrect type	25	10.59%	10	37.04%	13	30.23%	0	0%
Total Incorrect	236	100%	27	100%	43	100%	7	100%

Table 8. Timebank v1.2 - Distribution of Incorrect TTK annotations for all temporal types

	All Types	
	# Incorrect	% Incorrect
Exact Span Match Incorrect type	32	10.22%
Partial span match Correct type	233	74.44%
Partial span Match Incorrect type	48	15.34%
Total Incorrect	313	100%

II. VA medical documents

Three Discharge Summaries and two H&P notes could not be processed by TTK due to errors occurring while processing. Results below reflect the remaining 95 documents.

IAA was calculated 2 ways, according to:

1) Type match + partial overlap in marked texts (Table 9)
2) Type match + exact match in marked texts (Table 10)

Table 9. VA medical documents - Inter-annotator agreements (IAA) when match was defined as having the same type assignment and span overlap.

Type	IAA	matches	non-matches	Total
Date	78.85%	1182	317	1499
Time	93.28%	1208	87	1295
Duration	83.55%	640	126	766
Set	90.71%	1660	170	1830
Total	87.01%	4690	700	5390

Table 10. VA medical documents - Inter-annotator agreements (IAA) when match was defined as having the same type assignment and exact span match.

Type	IAA	matches	non-matches	Total
Date	73.52%	1102	397	1499
Time	90.97%	1178	117	1295
Duration	69.71%	534	232	766
Set	81.53%	1492	338	1830
Total	79.89%	4306	1084	5390

Table 11 shows the distribution of expressions found for each temporal type in the human annotated reference standard and TTK output.

Table 12 shows TTK scores against the reference standard.

Table 11. VA medical documents - Occurrences and distribution of each temporal type within the human annotated Reference Standard and TTK output for 95 VA medical documents.				
	Human		TTK	
Date	797	29.31%	1222	81.03%
Time	606	22.29%	19	1.26%
Duration	394	14.49%	267	17.71%
Set	922	33.91%	0	0%
	2719	100%	1508	100%

Table 12. VA medical documents - TTK scores against human annotations for the 95 VA medical documents					
	Date	Time	Duration	Set	Total
CORR	169	9	135	0	313
INCO	180	506	64	235	985
MISS	476	145	206	703	1530
SPUR	167	1	3	0	171
POSS	825	660	405	938	2828
ACT	516	516	202	235	1469
REC	0.20	0.01	0.33	0.00	0.11
PREC	0.33	0.02	0.67	0.00	0.21
F-measure	0.25	0.02	0.44		0.15
UND	0.58	0.22	0.51	0.75	0.54
OVG	0.32	0.00	0.01	0.00	0.12
SUB	0.52	0.98	0.32	1.00	0.76
ERR	0.48	0.22	0.50	0.75	0.51

Table 13 shows expression patterns missed by TTK.

Table 14 shows expression patterns incorrectly identified by TTK.

Table 13. VA medical documents - Notable expressions missed by TTK

Types	Expression patterns	Examples	% of total Misses
DATE	slash notation	02/02/2009	13.6
	dash notation	02-02-2009	3.3
	Containing month or month abbreviations	February 2, 2009	3.7
	units of time (days, weeks, months, years)	three days	6.2
TIME	colon notation	09/12/2003 15:05	6.6
	portions of the day (am, pm, morning, afternoon, night)	morning of 5/12	2.0
	units of time (minutes, hours,)	two hours	.6
DURATION	unit abbreviations (min, hr, yr, yo, y/o)	15 y/o	5. 2
	range durations	6-8 weeks	1.7
SET	All were missed		
	Medical abbreviations (q___, bid, tid)	qid, q8hrs, t.i.d.	24.1
	frequency expressions (per, every, a, /, at)	Twice a day, at bedtime, /hr	17.0

Table 14. VA medical documents - Notable expressions identified incorrectly by TTK

Types	Expression patterns	Examples	% of total Incorrect
DATE	Matching string, incorrect type (typed by TTK as duration instead of date)	two weeks	2.7
	Partial string match (absolute expressions)	Reference (FEB 13,1997) TTK (1997)	13.6
		Reference (02/02/2007) TTK (2007)	
TIME	Matching string, incorrect type (typed by TTK as date instead of time)	tonight, morning	0.9
	Partial string match (absolute expressions), correct type	Reference (01:00 p.m) TTK (00 p.m)	0.6
	Partial string match (absolute expressions), incorrect type (typed by TTK as date instead of time)	Reference (03/27/2007 09:03) TTK (2007)	49.6
		Reference (JAN 4,2006@10:00) TTK (2006)	
SET	Matching string, incorrect type (typed by TTK as date or duration instead of set)	every day, every other day, daily, monthly	11.0
		a day, a month	1.2
	Partial string match, incorrect type (typed by TTK as date or duration instead of set)	Reference (TWICE DAILY) TTK (DAILY)	8.1
		Reference (every six hours) TTK (six hours)	3.6

Discussion

I. Timebank v1.2

This author elected to use a rather strict criteria for comparing human and TTK markups. A correct match was one where human and machine annotations were identical. TTK overall recall (sensitivity), precision (positive predictive value), and F-measure on this corpus of documents for date, time, duration, and set expressions were 0.48, 0.58, and 0.53 respectively as compared to a reference standard created by human annotations. Date expressions accounted for 81.5% of all annotations in Timebank v1.2, and 87.3% of incorrect TTK date annotations were the result of partial span matching, but correct type matching. 94.4% of all missed annotations occurred in expressions of type date (70%) and type duration (24.4%). Time and Set expressions comprised 3.3% and 0.97%, respectively, of all expressions found in the Timebank v1.2 Corpus; much smaller compared with 22.3% and 33.9%, respectively, of all expressions found in the set of VA medical documents.

II. VA medical documents

IAA (Inter-Annotator Agreement) scores between the reviewers were good. From these annotations, I subsequently developed the reference standard. Depending on the strictness of the text span matching criteria, IAA was 87% (overlapping spans match criteria), and 80% (exact string match criteria). Using the strict criteria, TTK overall recall (sensitivity), precision (positive predictive value), and F-measure for date, time, duration, and set expressions were 0.11, 0.21, 0.15, as compared to a reference standard created by human annotations.

SET type classification was missing from the TTK output, which is contrary to the latest TimeML specifications and annotation guidelines [32, 38] found at http://www.timeml.org/site/publications/specs.html. SET type expressions represented a large proportion (33.9%) of all expressions in the reference standard, but 0% of expressions in the TTK output. TTK missed or only partially matched many absolute expressions in forms such as date slash notation (e.g. 02/02/1966), date dash notation (e.g. 02-02-1966), date month/month abbreviation format (e.g. February 2, 1966), and time colon notation (e.g. 01:00 p.m). Incorrect type classification occurred in 77.1% of incorrectly identified expression by TTK. This included typing expressions as date instead of time (50.5%), date or duration instead of set (23.9%), and duration instead of date (2.7%).

CHAPTER VI

MODIFICATION OF THE TTK SOURCE CODE

TTK scripts used for temporal expressions

I received the ttk-1.0 software package from Marc Verhagen (project manager for TARSQI). The GUTime directory (ttk-1.0/code/components/gutime) contained two python scripts (wrapper.py and btime.py), three Perl scripts (gutime.pl, TimeTag.pl, and postTempex.pl), and one Perl module (TempEx.pm v.1.05, Copyright 2001 by The MITRE Corporation).

1. The python script, wrapper.py, integrates GUTime with other components within the TTK.
    a. It accepts text string fragments after it has been preprocessed for part of speech (POS) tags by Treetagger, and then sends fragments along with the document creation times as inputs to the Perl script, gutime.pl.
    b. It stores the input to gutime.pl as a .gut.i.xml extension file and the output from gutime.pl as a .gut.o.xml extension file in the ttk-1.0/code/data/tmp directory.
    c. It then strips text fragments of non-lexical tags (i.e. tags not <lex> nor </lex>) for processing by gutime.pl. The output from gutime.pl contains lex and TIMEX3 tagging only. wrapper.py then sends the input and output from gutime.pl to a merge_tag routine defined in

ttk-1.0/code/utilities/xml_utils.py to reconstitute the non-lexical tags previously removed.

   d. After merging, btime.py "provides patches to GUTime".

2. The Perl script, gutime.pl, receives a text string fragment and document creation time as input from wrapper.py. The purpose of gutime.pl is to apply TIMEX3 tagging and values to temporal expressions found within the text fragment.

   a. It removes non-lexical tags from the body of the text fragment using its Cleanup subroutine.

   b. Subsequently, gutime.pl saves the document creation time and "cleaned up" fragment to a temporary file which is received as input to TimeTag.pl.

   c. The output from TimeTag.pl is then piped to postTempEx.pl.

   d. Further "cleaning" occurs to the output from postTempEx.pl. gutime.pl then saves the fragment into the ttk-1.0/code/data/tmp directory as a .gut.t1.xml extension file.

   e. Finally, it removes the temporary file made in step b.

3. The Perl script, TimeTag.pl

   a. Sends sentence fragments to subroutine TE_TagTIMEX (defined within TempEx.pm) for TIMEX3 tagging.

   b. Sends the output from TE_TagTIMEX and the document creation time to subroutine TE_AddAttributes (also defined within TempEx.pm) to normalize expressions found by TE_TagTIMEX .

4. The Perl module, TempEx.pm, is GUTime's principal engine. It contains the two major subroutines, TE_TagTIMEX (to identify and tag temporal expressions) and TE_AddAttributes (to normalize expressions and add attributes to TIMEX3 tags)

Debugging

In order to keep track of changes that occur to a text string fragment during processing by GUTime, I made source code changes to TimeTag.pl to preserve the temporary file in 2b, and to save the input and output for 3a (TE_TagTIMEX) and 3b (TE_AddAttributes) to a file, ttk-1.0/code/components/gutime/sentfile, for viewing. Thus, a text string could be loaded into the Tarsqi Control Panel for processing, and changes occurring to the text string at key steps during processing could be followed. Figure 14 shows an example of a sentence entry from sentfile which indicates the reference date, the sentence without tags, the sentence after part of speech tagging, the sentence after TE_TagTIMEX, and the sentence after TE_AddAttributes.

Reference date:
20091130

Sentence without tags:
He will come Christmas.
.
Sentence after POS tagging:
<lex pos="PP">He</lex> <lex pos="MD">will</lex> <lex pos="VB">come</lex> <lex pos="NNP">Christmas</lex> <lex pos=".">.</lex>

Sentence after TE_TagTIMEX:
<lex pos="PP">He</lex> <lex pos="MD">will</lex> <lex pos="VB">come</lex>
<TIMEX3 tid="t1" TYPE="DATE"><lex pos="NNP">Christmas</lex></TIMEX3>
<lex pos=".">.</lex>

Sentence after TE_AddAttributes:
<lex pos="PP">He</lex> <lex pos="MD">will</lex> <lex pos="VB">come</lex>
<TIMEX3 tid="t1" TYPE="DATE" VAL="20091225"><lex pos="NNP">Christmas</lex></TIMEX3> <lex pos=".">.</lex>

Figure 14. An example of a sentence after major steps in TTK processing

Errors occurring in the TTK program are recorded to a TTK log file, ttk-1.0/code/data/logs/ttk_log.html. This file is useful for debugging syntax errors and for printing values of string variables at specific locations within Perl scripts. To obtain the value of the variable $string, one can add a print to standard error command where desired within the Perl code, e.g. print STDERR “The string is $string”. The ttk_log.html file will then contain a message such as:

The string is <lex pos="PP">He</lex> <lex pos="MD">will</lex> <lex pos="VB">come</lex> <TIMEX3 tid="t1" TYPE="DATE"><lex pos="NNP">Christmas</lex></TIMEX3> <lex pos=".">.</lex>

Changes made

Pre-TempEx.pm

In addition to the changes for debugging mentioned above, I also made source code changes to ttk-1.0/code/docmodel/model.py to accept "simple-xml" files with imbedded reference document creation date and time (DCT). This allows valuation to occur in TE_AddAttributes based on the DCT. Further, in order to fix a recognition problem for expressions such as "Dec. 25", where the period and space caused TTK to process Dec, then 25 as two separate sentence fragments, I also altered TimeTag.pl.

TempEx.pm

Most of the changes and modification made to TTK for time recognition were ones that I made to TempEx.pm. TE_TagTIMEX recognizes and tags temporal expressions through a long series of perl regular expressions and text substitutions. Each regular expression can be validated by loading representative text into the Tarsqi Control Panel and checking (viewing) for the accuracy of the results using the TIMEX viewer (Figure 12, p. 45).

The principal problem with missed and partial recognition of expressions involved unaccounted for white spaces that were added during part of speech tagging. Although TTK installation instructions located in ttk-1.0/doc/manual/index.html claimed "The Tarsqi toolkit is designed to work seamlessly with the IMS Treetagger", with GUTime's poor baseline performance on Timebank v1.2 articles and VA medical documents (described in Chapter V), one wonders whether GUTime was actually developed using a different part of speech tagger or version of the TreeTagger.

After TreeTagger, each token (e.g. word) is surrounded with an opening and closing tag, e.g. “<lex pos="UH">Hello</lex>”. Between encapsulated tokens is a space, e.g. “<lex pos="UH">Hello</lex> <lex pos="NN">world</lex>”. It is this space that was often not accounted for in TE_TagTIMEX regular expressions, resulting in recognition errors.

Also, to avoid system errors, TIMEX3 tags should be placed contiguous (absolutely no intervening white space) with opening and closing token tags (please see the example in Figure 15).

<TIMEX3 TYPE="DATE""><lex pos="NNP">Christmas</lex> <lex pos="NNP">Day</lex></TIMEX3>

Figure 15. An example of TIMEX3 tags contiguous with token opening and closing tags

Given that the regular expression for an opening tag ($OT) and closing tag ($CT) are $OT = "(<[^\/][^>]*>)" and $CT = "(<\/[^>]*>)", the following Perl substitution command example captures the substring *Christmas morning*, *Christmas day*, and *Christmas night* within the string $string with TIMEX3 tags of type time:

```
$string =~ s/($OT+Christmas$CT+\s+$OT(morning|day|night)$CT+)/<TIMEX3
TYPE=”TIME”>$1</TIMEX3>/gi;
```

Special characters, such as “/”, “:”, “@”, and “,” are also tokenized, and thus are encapsulated by opening and closing tags, e.g. 12/25 appears as

<lex pos="CD">12</lex> <lex pos="SYM">/</lex> <lex pos="CD">25</lex>.

Dashes (“-“), are not tokenized, e.g. 12-25-2009 appears as

<lex pos="CD">12-25-2009</lex>.

Periods (“.”) within a phrase are not tokenized, e.g. 12.25.2009 appears as <lex pos="CD">12.25.2009</lex>. A period followed by white space, however, is used for chunking sentences. Thus, a correction was made to TimeTag.pl to fix a recognition problem for expressions such as “Dec. 25”, where the period and space caused TTK to process Dec, then 25 as two separate sentence fragments.

With these observations, I made source code corrections, particularly to regular expressions dealing with slash notations (e.g. 12/25, 12/2009, 12/25/2009), dash notations, (e.g. 12-25, 12-2009, 12-25-2009), dot notations, (e.g. 12.25.2009), month word/abbreviations patterns (e.g. December 25, 2009; December, 2009; Dec 2009; Dec 25th), colon notations (e.g. 10:00 P.M.), and compound time expressions with date and time (e.g. 12/25/2009@10:00pm).

Correct type assignments (TIME, DATE, DURATION or SET) to expressions required recognition of key words or characters within expressions. The TTK tended to over-generalize expressions as type date, but using key words and characters allowed for separating the different types accordingly. An expression such as “12/25/2009@10:00pm” is typed as time because of characters “@”, “:”, and pm. Other words useful for typing expressions as time instead of date include “am”, “morning”, “noon”, “afternoon”, “evening”, and “night”; e.g. “Christmas morning” and “yesterday afternoon”. Words and characters useful for typing expressions as type set instead of date include “alternate”, “each”, and “every”; e.g. “every Christmas morning”, “each afternoon”, and “alternate Wednesdays”.

Expressions with units of time (e.g. seconds, minutes, hours, days, weeks, months, years) may be of type date, time, duration, or set. Expression were designated as type date or time if the unit of time was preceded by "in", "within", and "after"; and/or followed by "ago", "before", "previous", "previously", "previous to", “prior”, “prior to”, “hence", "from", "following", "after", and "later”; e.g. “follow-up in 2 weeks”, “2 hours ago” and “2 months hence”. Expression were designated as type set if the unit of time was preceded by “per”, “/”, “alternate”, “each”, “every”, and “q”; e.g. “every 2 weeks”, “per second”, and “q2h”. Otherwise, expressions that contain a unit of time were designated as type duration, particularly if the unit of time was preceded by "the", "for", "for over", "over", "throughout", "through", and "x”; and/or followed by “course", "duration", "long", and "regimen"; e.g. “x 2weeks”, “for 2 weeks” and “2 weeks long”.

TE_TagTIMEX, identifies durations first, dates second, and times third. The author added the identification of sets as its fourth step. A disadvantage to the Perl rule-based approach is that strings may follow more than one rule. Thus, for example, a duration tag may be upended downstream by a set tag. This happened when the author tried adding to TE_TagTIMEX recognition of expressions containing a unit of time preceded by “a” or “an”, e.g. “an hour”, “a week”, and “a year”. Such expressions can be of type date (e.g. “a week later”), time (e.g. “an hour ago”), duration (e.g. “an hour long””), or set (e.g. “100 cc an hour”). Figure 16 shows what happened to the string “an hour long” as it was processed by TE_TagTIMEX after recognition of sets was added by the author. First, duration tags were inserted around “an hour” and, later, these duration tags were upended by set tags.

<TIMEX3 TYPE="DURATION" VAL="PT1H"><lex pos="DT">an</lex> <lex pos="NN">hour</lex></TIMEX3> <lex pos="RB">long</lex>

<TIMEX3 TYPE="SET"><TIMEX3 TYPE="DURATION" VAL="PT1H"><lex pos="DT">an</lex> <lex pos="NN">hour</lex></TIMEX3></TIMEX3> <lex pos="RB">long</lex>

Figure 16. A duration tag gets upended by a set tag

A routine was thus created to “protect” expressions containing a unit of time preceded by “a” or “an” that had previously been tagged as duration, date, or time from being tagged as set.

TTK v1.0 did not tag expressions as type set. It actually typed set expressions as type date with a set attribute specified as “yes” (please see Figure 17).

<TIMEX3 tid="t1" TYPE="DATE" SET="YES" PERIODICITY="F1D"><lex pos="DT">every</lex> <lex pos="NN">day</lex></TIMEX3>

Figure 17. An example of TTK’s current treatment of SET expressions

In accordance with TimeML guidelines, the author added set as a temporal type and separated set expressions from date expressions by identifying key words and characters described earlier, i.e. expressions preceded by “each”, “every”, “every other”, and “alternate”. Additional recognition for medically related set expressions were added, such as those in Table 3, p. 32, including “per hr”, “/hr”, “q___”, “bid”, “tid”, “qoweek”, etc.

Practically all of over fifty regular expressions within the TE_TagTIMEX subroutine were modified to expand its coverage of temporal expressions. Additionally, improvements in the recognition of intervals were made, e.g. for expressions such as “2-3 weeks ago”, “Feb 2-5, 1966”, “’95-‘99”, “2/2 to 2/5”, etc. Although it was tempting to add an “interval” or “daterange” type to the time schema, these interval expressions were primarily added as type date.

A byproduct of increasing the sensitivity for temporal expression recognition is increasing false-positive results. For example, not all four digit numbers represent years, though the occurrence of false positives can be decreased by narrowing accepted years to those between 1900 and 2025. A number of subroutines were created to remove some obvious non-temporal expressions that were captured by TE_TagTIMEX. Examples include non-temporal slash expressions such as “60/30” and “98/58”, non-temporal dash notations such as “88-90”, and non-temporal use of abbreviations such as “320 H” (high glucose level, not 320 hours). Still, many expressions are ambiguous or difficult to sort out without taking account contextual information, e.g. “5/5” may represent May 5th or 5 out of 5 motor strength.

Post-TempEx.pm

After a sentence fragment is processed by subroutines TE_TagTIMEX and TE_AddAttributes in TempEx.pm, non-lexical tags that were previously removed are reinserted by the merge_tag routine defined in ttk-1.0/code/utilities/xml_utils.py. Unfortunately the reconstitution of these non-lexical tags occasionally resulted in poorly formatted XML, causing system error and subsequent removal of TIMEX3 tags. To rectify these errors, I created a Perl script, ttk-1.0/code/components/gutime/merge.pl to replace the merge_tag routine in xml_utils.py. Additionally, the need for btime.py was eliminated because of the changes I made to TE_TagTIMEX for recognition of four digit years.

Discussion

TTK incorporates part of speech (POS) tags before it adds time and event tags. These POS tags are actually not used at all by TE_TagTIMEX for the recognition of temporal expressions, but truly complicate the regular expressions needed for identifying temporal information from documents. These POS tags play a very small role in the valuation of time expressions performed by subroutine TE_AddAttribute. They are, however, needed for calculating the rules that link events to temporal expressions, although that module of the TTK system is not evaluated in this study.

Modifications to a number of TTK scripts were made to improve and expand its recognition of temporal expressions for medical documents. The rule-based approach used by GUTime (the use of a series of Perl regular expressions) is susceptible to errors when white space is not accounted for, and when strings can follow more than one rule.

The accurate recognition and typing of time expressions is crucial for the proper valuation (normalization) of these expressions. Though beyond the scope of this project, this author also modified TE_AddAttribute, to correctly place values (primarily following ISO 8601 conventions) on expressions found by TE_TagTIMEX

CHAPTER VII

A LOOK AT THE MODIFIED TTK'S RECOGNITION OF TEMPORAL EXPRESSIONS

Aims

1. Identify temporal expressions within a set of H&P and Discharge Summaries
2. Compare human annotations of temporal expressions to the modified TTK (mod-TTK) automated annotations of temporal expressions.

Methods

Sample

From the POEM database, a random sample of 50 Physician H&P and 50 Discharge Summary notes, different from those used in Chapter V, was extracted using SQL query. Inclusion criteria included file (document) sizes between 2 and 10 kilobytes in length.

Measures: Temporal expressions identified from the documents (same as Chapter V).

DATE (for calendar date):

1. Included specialized time patterns such as “12/1/2008”, “10-01”, and “02/2009”.
2. Included proper nouns such as days of the week (“Monday”, “Tuesday”, “Wed”); months of the year (“January”, “Feb”); holidays (“Christmas”, “Groundhog Day”), and seasons.

TIME (for time of day):

1. Included specialized time patterns such as “3 PM”, “12/1/ 2008 0400”, and “5 o’clock”.
2. Included nouns such as “morning”, “evening”, “afternoon”.

Duration denotes a span of time at varying levels of granularity:

1. Included expressions containing nouns such as “seconds”, “minutes”, “days”, “weeks”, and “years”, denoting a span of time, e.g. “for three weeks”, “over several days”, “throughout the year”.
2. Did not include expressions such as “<u>in</u> three days”, “<u>in</u> four hours”, “2 weeks <u>ago</u>”, which were categorized as date or time instead.

Set (for reoccurring times i.e. periodicity):

1. Included expressions containing adverbs and adjectives such as “daily”, “every day”, “per week”, and abbreviations such “qd”, “b.i.d”, and “t.i.d”.

Procedures

Step1. Obtain human annotations for the VA medical documents.

The same two human reviewers as in Chapter V identified and annotated temporal expressions using Knowtator v1.8, a general-purpose text annotation tool that is integrated with the Protégé v3.3.1 knowledge representation system.

The annotation scheme created within Knowtator held the 4 types (DATE, TIME, DURATION, and SET) of temporal expressions specified by TimeML for TIMEX3. A temporal expression is highlighted (left click and drag with the mouse), then, within the left panel, the type class is chosen (right click then select from a pop-up dialogue box). For each document, reviewers' annotations were stored in XML format, recording string spans, string positions within the document, and string types.

Using Knowtator, I merged document sets from both reviewers and, subsequently, calculated inter-annotator agreement (IAA) for each of the four types of temporal expressions. IAA was performed using "match" defined as an exact span match, and then again using "match" defined as only overlapping spans. A Perl script then looped through the merged annotations from both reviewers and kept annotations that were unique, kept only one if there was an exact match, and kept the longest string if there were overlapping spans. A list of temporal expressions with their types (date, time, duration, or set) ordered according to their position within each document was then extracted from each document and stored.

Step2. Obtain mod-TTK annotations for the VA medical documents.

In the prior evaluation presented in Chapter V, I processed each document individually using the Tarsqi Control Panel. For the mod-TTK evaluation, I created a Perl script to batch process this set of medical documents. Given the directory that contained a set of documents, the script:

1. Extracted each document creation time (DCT).
2. Converted each DCT to ISO 8601 format.
3. Converted each document to "simple-xml" format
   a. Added "<Doc><Text>" and "</Doc></Text>" at the beginning and ending.
   b. Changed and removed XML special characters within documents, such as "<", ">", and "&".
   c. Added the DCT within Date_time tags before each text section e.g. <DATE_TIME>20090511T08:30</DATE_TIME>
4. Processed each document (in "simple-xml" format) through mod-TTK using the following Linux command within the script,

   `python tarsqi.py $doctype pipeline=PREPROCESSOR,GUTIME $ttk_source "$dir/ttk_tagged_xml"`;

   Where $doctype = "simple-xml",

   $ttk_source = the directory where documents was located,

   $dir/ttk_tagged_xml = the directory where output was placed
5. Listed temporal expressions and their types according to their position within each document.
6. Stored each list for use in Step 3.

Step3. Match mod-TTK annotations to human annotations.

For each document the ordered list of human annotated time expressions was compared with the corresponding ordered list of mod-TTK identified expressions. Table 15 shows an example of actual human and mod-TTK annotation lists found for a single document.

Table 15. VA medical documents with mod-TTK: An example of ordered lists of human and mod-TTK annotations for a single document

Human annotations	types	Mod-TTK annotations	types
JUL 24, 2001@12:56	Time	JUL 24, 2001@12:56	Time
JUL 25, 2001@12:21:24	Time	JUL 25, 2001@12:21:24	Time
77-year-old	Duration	77-year-old	Duration
recently	Date		
eight years	Duration	last eight years	Duration
40 years	Duration	past 40 years	Duration
several days	Duration	several days	Duration
Monday, July 23rd, 2001	Date	Monday, July 23rd, 2001	Date
July 17th, 2001	Date	July 17th, 2001	Date
q.12 hours	Set	q.12 hours	Set
morning of 07/18/01	Time	morning	Time
		7/18/2001	Date
Sunday, July 22nd, 2001, in the a.m.	Time	Sunday, July 22nd, 2001	Date
		a.m.	Time
Monday, July 23rd, 2001	Date	Monday, July 23rd, 2001	Date
		Fourth year	Duration
7/25/2001	Date	7/25/2001	Date
07/24/01@1256	Time	07/24/01@1256	Time
7/30/2001 10:55	Time	7/30/2001 10:55	Time
7/30/2001 13:55	Time	7/30/2001 13:55	Time

I created a Perl script to loop through the list of expressions found in the reference standard and those found by mod-TTK. The same scoring system described in Chapter V was used and copied below for the reader's convenience. For a given human annotated expression, a corresponding TTK annotated expression matched in the following manner:

Category	String (span) match	Type (date, time, duration, set) match
1	exact match	Match
2	exact match	not matched
3	overlap (partial match)	Match
4	overlap (partial match)	not matched
Miss	no match	

I categorized as Spurious, TTK annotated expressions with no matching or overlapping spans in the reference standard. I then counted and saved to a file the expressions fitting each category.

For each type, the scoring used followed the Time Expression Recognition and Normalization (TERN) 2004 Plan, available at http://timex2.mitre.org/tern.html and http://timex2.mitre.org/tern_2004/tern_evalplan-2004.29apr04.pdf

1. CORR (Correct) = The two items under comparison were considered identical. (The # of expressions in Category 1)

2. INCO (Incorrect) = The two items under comparison were not identical. (The # of expressions in Category 2 + Category 3 + Category 4)

3. MISS (Missing) = A reference had no TTK output aligned with it. (The # of Missed expressions)

4. SPUR (Spurious) = A TTK output had no reference aligned with it.

(The # of Spurious expressions)

5. POSS (Possible) = The number of items in the reference that contributed to the final score

= CORR + INCO + MISS

6. ACT (Actual) = The number of items in the TTK output:

= CORR + INCO + SPUR

7. REC (Recall) = The # of reference items that was present in the TTK output

= CORR/POSS

8. PREC (Precision) = The # of TTK output items that was actually in the reference

= CORR / ACT

9. F- measure combined recall and precision into one measure

$$\text{F-measure} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{(\text{precision} + \text{recall})}$$

10. UND (Undergeneration) = MISS / POSS

11. OVG (Overgeneration) = SPUR / ACT

12. SUB (Substitution) = INCO / CORR + INCO

13. ERR (Error) = MISS / CORR + INCO + SPUR + MISS

Results

One H&P note could not be scored because it included the human annotated expression "??/??/98" that resulted in a processing error during scoring.

IAA was calculated 2 ways, according to:

1) Type match + partial overlap in marked texts (Table 16)

2) Type match + exact match in marked texts (Table 17)

Table 16. VA medical documents - Inter-annotator agreements (IAA) when match was defined as having the same type assignment and span overlap.

Type	IAA	matches	non-matches	Total
Date	88.07%	1794	243	2037
Time	91.05%	1710	168	1878
Duration	84.41%	742	137	879
Set	92.22%	1708	144	1852
Total	89.58%	5954	692	6646

. Table 17. VA medical documents - Inter-annotator agreements (IAA) when match was defined as having the same type assignment and exact span match

Type	IAA	matches	non-matches	Total
Date	79.92%	1628	409	2037
Time	85.84%	1612	266	1878
Duration	77.59%	682	197	879
Set	80.24%	1486	366	1852
Total	81.14%	5408	1238	6646

Table 18 shows the distribution of expressions found for each temporal type in the human annotated reference standard and mod-TTK output.

Table 18. VA medical documents with mod-TTK: Occurrences and distribution of each temporal type within the human annotated Reference Standard and mod-TTK output for VA medical documents.

	Human		Mod-TTK	
Date	1043	30.25%	1254	35.25%
Time	959	27.81%	924	25.98%
Duration	472	13.69%	459	12.90%
Set	974	28.25%	920	25.86%
	3448	100%	3557	100%

Table 19 shows mod-TTK scores against the reference standard.

Table 19. VA medical documents with mod-TTK: mod-TTK scores against human annotations for the VA medical documents

	Date	Time	Duration	Set	Total
CORR	831	801	268	616	2516
INCO	129	135	126	255	645
MISS	121	67	78	148	414
SPUR	199	32	59	77	367
POSS	1081	1003	472	1019	3575
ACT	1159	968	453	948	3528
REC	0.77	0.80	0.56	0.60	0.70
PREC	0.72	0.83	0.59	0.65	0.71
F-measure	0.74	0.81	0.58	0.63	0.71
UND	0.11	0.07	0.17	0.15	0.12
OVG	0.17	0.03	0.13	0.08	0.10
SUB	0.13	0.14	0.32	0.29	0.20
ERR	0.09	0.06	0.15	0.14	0.11

Table 20 shows the distribution of incorrect TTK annotations for each temporal type. Table 21 shows the distribution of incorrect TTK annotations for all types.

Table 20. VA medical documents with mod-TTK: Distribution of Incorrect mod-TTK annotations for each temporal type

	Date		Time		Duration		Set	
	# Incorrect	% Incorrect	# Incorrect	% Incorrect	# Incorrect	% Incorrect	# Incorrect	% Incorrect
Exact Span Match Incorrect type	14	10.85%	23	17.04%	18	14.28%	18	7.05%
Partial span match Correct type	83	64.34%	45	33.33%	81	64.28%	186	72.94%
Partial span match Incorrect type	32	24.81%	67	49.63%	27	21.43%	51	20%
Total Incorrect	129	100%	135	100%	126	100%	255	100%

Table 21. VA medical documents with mod-TTK: Distribution of Incorrect mod-TTK annotations for all temporal types

	All Types	
	# Incorrect	% Incorrect
Exact Span Match Incorrect type	73	11.32%
Partial span match Correct type	395	61.24%
Partial span Match Incorrect type	177	27.44%
Total Incorrect	645	100%

Figure 18 shows an example of a medication list in a format that frequently caused mod-TTK to miss or partially recognize time expressions. For drug #1 (glyburide), human annotation would result in a reference expression of “TWICE A ACTIVE DAY” instead of the true expression “TWICE A DAY”.

```
MEDICATIONS ON DISCHARGE:
1)   GLYBURIDE 5MG TAB TAKE TWO TABLETS BY MOUTH TWICE A     ACTIVE
       DAY FOR DIABETES
2)   FOLIC ACID 1MG TAB TAKE ONE TABLET BY MOUTH EVERY       ACTIVE
       MORNING
3)   INSULIN REG HUMAN 100 UNIT/ML NOVOLIN R INJECT 7        ACTIVE
       UNITS SUBCUTANEOUSLY TWICE DAILY FOR DIABETES
4)   CYANOCOBALAMIN 500MCG TAB TAKE ONE TABLET BY MOUTH      ACTIVE
       EVERY DAY FOR VITAMIN DEFICIENCY
5)   OMEPRAZOLE 20MG SA CAP TAKE ONE CAPSULE BY MOUTH        ACTIVE
       TWICE A DAY FOR STOMACH - DO NOT CRUSH TAB/CAP
6)   MULTIVITAMIN/MINERALS CAP/TAB TAKE 1 TABLET BY MOUTH    ACTIVE
       EVERY DAY
7)   FERROUS SULFATE 325MG TAB TAKE ONE TABLET BY MOUTH      ACTIVE
       THREE TIMES A DAY FOR IRON
8)   FINASTERIDE 5MG TAB TAKE ONE TABLET BY MOUTH EVERY      ACTIVE
       DAY FOR PROSTATE
```

Figure 18. An example of a medication list in tabular format

Table 22 shows examples of expressions missed by mod-TTK.

Table 23 gives examples of some notable expressions identified incorrectly by mod-TTK

Table 22. VA medical documents with mod-TTK: Some notable expressions missed by mod-TTK			
Types	Expression patterns	Examples	% of total Misses
DATE	Key clinical events (19 of 121)	Post-op, discharge, admission	4.5
SET	42 of 148 missed expressions due to formatting issues of medication tables causing inclusion of “ACTIVE” and spaces	“EVERY ACTIVE MORNING”	10.1
	13 of 148 missed expressions due to formatting issues of medication tables causing inclusion of spaces	“every few days”	3.1
	23 of 148 missed expressions due to / notation	/HR	5.6

Table 23. VA medical documents with mod-TTK: Some notable expressions identified incorrectly by mod-TTK

Types	Expression patterns	Examples	% of total Incorrect
DATE	Partial string match, correct type (25 of 83 incorrect expressions due to mod-TTK including adjectives, adverbs, prepositions, and conjunctives)	Reference (2 weeks) Mod-TTK (for 2 weeks, approximately 2 weeks, over 2 weeks, 2 weeks long, 2 weeks after, etc.)	3.8
TIME	Partial string match, correct type (15 of 45 incorrect expressions due to inclusion of adjectives, adverbs, prepositions, and conjunctives)	The, a, last, this, in, etc.	2.3
	Partial string match, correct type (9 of 45 incorrect expressions due to dropped period at end of string-Knowtator artifact)	Reference (6:00 p.m) Mod-TTK (6:00 p.m.)	1.4
	Incorrect type (90 total), eg. Date instead of time (58 of 90)	Now, current, tonight	14.0
	Duration instead of time (23 of 90)	Reference (24 hours) mod- TTK (last 24 hours	
DURATION	Partial string match, correct type (70 of 81 incorrect expressions due to mod-TTK including adjectives, adverbs, prepositions, and conjunctives)	Reference (2 weeks) Mod-TTK (for 2 weeks, approximately 2 weeks, over 2 weeks, 2 weeks long, 2 weeks after, etc.)	10.8
SET	Partial string match, correct type (110 of 186 incorrect expressions due to dropped period at end of string-Knowtator artifact)	Reference (b.i.d, q.d) mod-TTK(b.i.d., q.d.)	17.1

Discussion

IAA scores between the reviewers were good, and a reference standard was subsequently developed. IAA was 90% when match was defined as overlapping spans, and 81% when match was defined as exact string match. Using strict criteria that a correct match was one where machine and human annotations were identical, mod-TTK overall recall (sensitivity), precision (positive predictive value), and F-measure for date, time, duration, and set expressions were 0.70, 0.71, 0.71, respectively, as compared to a reference standard created by human annotations.

88% of incorrectly identified expressions involved partial string matching. For sets, 110 of 186 set expressions showing partial string matching and correct type matching (17.1% of all incorrect expressions) were due to a behavior of Knowtator dropping periods at the end of string annotations. Thus, many set expressions identified correctly by mod-TTK, such as “b.i.d.”, were counted as incorrect because the reference strings were missing end periods, i.e. “b.i.d”. For dates and duration showing partial string matching and correct type matching, inclusion of extra words such as prepositions, adjectives, adverbs, and conjunctives were the only difference between mod-TTK and human annotations, accounting for 14.6% of all incorrect expressions. With the current strict scoring system, human annotations such as “for 3 weeks” (type duration) and corresponding machine annotations such as “3 weeks” (type duration) are marked as incorrect because the two spans are not identical, even though the two strings are in fact identical in meaning. Compare this with human annotation such as “3 weeks ago” (type date) and corresponding machine annotation such as “3 weeks” (type duration). This pair of strings should continue to be marked as incorrect since the adjective “ago” significantly changes the type of the expression.

Although not part of the annotation guidelines, reviewers did annotate several expressions which were event based such as “after admission” and “post-op day 1”. Predictably all of these were missed by mod-TTK.

CHAPTER VIII

CONCLUSION

The author commends the developers of TimeML and the TTK for creating a framework whereby the automated identification and linkage of time and events can be performed. The recognition of temporal expressions is the first of two steps performed by TTK's GUTime time-tagger. In VA admission and discharge notes, about 30% of temporal expressions are of type set, most often describing when medications and other treatment modalities are to be taken. This is the author's reason for installing the recognition of temporal set expressions into TTK, previously missing in TTK v1.0 but clearly specified by TimeML. In this study, looking solely at TTK's recognition of temporal expressions in VA medical documents, TTK performance was markedly improved from an F-measure of 0.15 to an F-measure of 0.71 by correcting white space discrepancies between GUTime and Treetagger, and by expanding its recognition of sets and other general and medical specific temporal expressions.

Other contributions of this work are proposed extensions to TimeML's model of time and ISO 8601's coverage for sets.

1. TimeML's model of time is simple and useful, consisting of four temporal types - Date, Time, Duration and Sets. This model covered 88.5% of temporal expressions found in a set of 50VA H&P and 50 VA DC notes described in Chapter 4, but did not cover approximately 11% of temporal expressions found, those related to key clinical

events such as “post discharge” and “pre-admission”; and stand-alone expressions that refer to sequence/ordering and frequency/onset of disease/symptoms such as “subsequently”, “chronic diarrhea”, “recurrent chest pain”. We propose the addition of these missing concepts to the TimeML model to increase its coverage of VA utterances. Such additions may be incorporated with modifications to TimeML’s recognition of events, particularly the recognition of medically related nouns as events for the former expressions and the recognition as aspectual events for the latter stand-alone expressions.

2. ISO 8601 defines dates, times, and durations quite well, but defines sets as commonly seen in medical documents less well. The author proposes a simple extension of ISO 8601’s representation of repeating intervals to include a parameter “Q” (quantity) to describe the number of times something occurs in a duration of time.

Limitations of the Study

The author recognizes a number of limitations to this study.

1. The principal limitation of the study is that the author played the dual role of annotator and programmer.

2. The study looked only at VA H&P and DC documents.

3. The scoring system, though strict, was not optimal since the computer algorithm created for scoring compared expressions based on their order of appearance within documents rather than their specific positions within documents. In addition, the strictness of the scoring system counted all non-identical string pairs as incorrect.

Future Work

The author recognizes that much work still needs to be done to make the TTK applicable for medical use.

1. Further improvements in temporal recognition are required. Further TTK evaluations and code modifications based on a greater number and variation of medical documents will make it more applicable for general use. This work will require improving the scoring system and recruiting additional reviewers to annotate additional document sets from the VA and other healthcare sites.

2. Evaluations of the TE_AddAttributes subroutine in TempEx.pm for the normalization of temporal expressions found are needed. The normalization of temporal expressions is the second step performed by TTK's GUTime. GUTime normalizes expressions to ISO 8601 formats, and currently uses the document creation time (DCT) as its main anchor for the normalization of time expressions. While working on this temporal recognition project, the author has also modified a number of routines in the TE_AddAttributes subroutine. Evaluations for the accuracy of the values placed on all types of temporal expressions (not only Sets, but Date, Time, and Duration as well) are needed. Additionally, studies are needed to explore the identification and use of anchor (reference) points within medical documents in addition to the DCT.

3. Once done with the time-tagger, evaluations and modifications of the remaining components of TTK (i.e. Evita and linkers) as it pertains with medical documents are needed.

a) The recognition of events is performed by TTK's Evita. Evita tags verbs as defining the time of an action. Relevant medical events and concepts are chiefly nouns representing diseases, symptoms, diagnostic tests, and treatment modalities. Additional studies are needed to explore whether medical events and concepts can be linked ("piggy-backed") with Evita recognized verbs.

b) The generation of links between times and events and the merging of these links based on Allen's algorithm for transitive closure are performed by a number of scripts incorporated within TTK's temporal processing module (e.g. Slinket, Blinker, S2T, TLink Classifier, SputLink, and Link Merger). Additional studies are needed to explore how well these linkers perform with medical documents and whether they can be modified for medical use.

REFERENCES

1. TimeML. Markup Language for Temporal and Event Expressions. http://www.timeml.org/site/index.html

2. Pustejovsky J, Ingria R, Sauri R, Castano J, Littman J, Gaizauskas R, Setzer A, Katz G, Mani I. The Specification Language TimeML. January 23, 2004. http://xml.coverpages.org/TimeML-SpecLang200401.pdf

3. New Research Center Focuses on IT and the Intelligence Community. http://www.mitre.org/news/digest/defense_intelligence/09_02/di_research_nnrc.html

4. TARSQI - Temporal Awareness and Reasoning Systems for Question Interpretation. http://www.timeml.org/site/tarsqi/index.html and http://www.timeml.org/site/tarsqi/toolkit/index.html

5. Verhagen M, Mani I, Sauri R, Knippen R, Jang SB, Littman J, Rumshisky A, Phillips J, Pustejovsky J. Automating Temporal Annotation with TARSQI. Proceedings of the ACL. Interactive Poster and Demonstration Sessions, pages 81–84, Ann Arbor, June 2005. http://www.aclweb.org/anthology-new/P/P05/P05-3021.pdf

6. Hurwitz B. Time, a narrative organizer of events and experience. The British Journal of General Practice; April 2002:344-347.

7. Shahar Y. Dimension of time in Illness: An objective view. Ann Intern Med 2000; 132:45-53.

8. Shahar Y, Combi C. Timing is everything. Time-Oriented Clinical Information Systems. West J Med 1998; 168:105-113.

9. Terenziani P, Snodgrass RT, Bottrighi A, Torchio M, Molino G. Extending temporal databases to deal with telic/atelic medical data. Artificial Intelligence in Medicine. 2007;39:113-126.

10. Friedman C. Towards a Comprehensive Medical Language Processing System: Methods and Issues. Proc AMIA Annual Fall Symp. 1997:595-599.

11. Denny JC, Smithers JD, Miller RA, Spickard A 3rd. Understanding medical school curriculum content using KnowledgeMap. J Am Med Inform Assoc.Jul-Aug 2003;10(4):351-362.

12. Elkin PL, Mohr DN, et al. Standardized Problem List Generation, Utilizing the Mayo Canonical Vocabulary Embedded within the Unified Medical Language System. Proc AMIA Annu Fall Symp. 1997:500-4.

13. Zhou L, Hripcsak G. Temporal reasoning with medical data – A review with emphasis on medical natural language processing. J Biomed Inf 2007; 40: 183-202.

14. Zhou L, Friedman C, Parsons S, Hripcsak G. System Architecture for Temporal Information Extraction, Representation and Reasoning in Clinical Narrative Reports. AMIA Annu Symp Proc. 2005; 2005: 869–873.

15. Zhou L, Melton GB, Parsons S, Hripcsak G. A temporal constraint structure for extracting temporal information from clinical narrative. J Biomed Inf 2006; 39:424-439.

16. Hripcsak G, Zhou L, Parsons S, Das A, Johnson SB. Modeling Electronic Discharge Summaries as a Simple Temporal Constraint Satisfaction Problem. J Am Med Inform Assoc 2005; 12:55-63.

17. Zhou L, Parsons S, Hripcsak G. The Evaluation of a Temporal Reasoning System in Processing Clinical Discharge Summaries. J Am Med Inform Assoc 2008; 15:99-106.

18. Jurafsky D, Martin J. Speech and Language Processing. Second Edition. Pearson Education, Inc., Upper Saddle River, New Jersey. 2009:743-752.

19. Ferro L, Gerber L, Mani I, Sundheim B, Wilson George. TIDES. 2005 Standard for the Annotation of Temporal Expressions. http://projects.ldc.upenn.edu/ace/docs/English-TIMEX2-Guidelines_v0.1.pdf April 2005.

20. Allen, JF. Maintaining knowledge about temporal intervals. Communications of the ACM 26(11), 832-843, November 1983.

21. Allen, JF. Time and time again: The many ways to represent time. International Journal of Intelligent Systems 6(4), 341-356, July 1991.

22. Villain MB. A system for reasoning about time. AAAI-82 Proceedings. 1982.

23. Schloss Dagstuhl Seminar Homepage. Annotating, Extracting and Reasoning about Time and Events. http://www.dagstuhl.de/en/program/calendar/semhp/?semnr=05151

24. Harkema H, Setzer A, Gaizauskas R, Hepple M, Power R, Rogers J. Mining and Modeling Temporal Clinical Data. http://www.allhands.org.uk/2005/proceedings/papers/507.pdf

25. TimeML in a Medical Application. http://kathrin.dagstuhl.de/files/Materials/05/05151/05151.SetzerAndrea.Slides.ppt

26. Pustejovsky J, Knippen R, Littman J, Sauri R. Temporal and Event Information in Natural Language Text. 2005. http://www.timeml.org/site/publications/timeMLpubs/PustejovskyEtAl2.pdf.

27. Verhagen M, Mani I, Sauri R, Knippen R, Jang SB, Littman J, Rumshisky A, Phillips J, Pustejovsky J. Automating Temporal Annotation with TARSQI. http://www.timeml.org/site/publications/timeMLpubs/tarsqi-acl05.pdf

28. Verhagen M, Pustejovsky J. Temporal Processing with the TARSQI Toolkit. http://www.aclweb.org/anthology-new/C/C08/C08-3012.pdf

29. Sundheim B. Association of Absolute Times with Events: Current Status and Future Challenges. July 21, 2002. http://www.timeml.org/site/terqas/documentation/timestamping-events.pdf

30. GUTime- Adding TIMEX3 tags. http://www.timeml.org/site/tarsqi/modules/gutime/index.html

31. Mulkar-Mehta R, Hobbs JR, Liu C, Zhou XJ. Discovering Causal and Temporal Relations in Biomedical Texts. Association for the Advancement of Artificial Intelligence. 2009. http://www.aaai.org/Papers/Symposia/Spring/2009/SS-09-07/SS09-07-013.pdf

32. TimeML specifications. http://www.timeml.org/site/publications/specs.html

33. Sauri R, Littman J, Knippen B, Gaizauskas R, Setzer A, Pustejovsky J. TimeML Annotation Guidelines Version 1.2.1. January 31, 2006. http://www.timeml.org/site/publications/timeMLdocs/annguide_1.2.1.pdf

34. Mandel M, Walker, C. Linguistic Data Consortium. Time Annotation Guidelines for Less Commonly Taught Languages (Based upon the TIMEX2 Standard) Version 1.0, March 28, 2006. http://crl.nmsu.edu/say/TimeAnnotationGuidelinesV1.0.pdf

35. TimeML Corpora. http://www.timeml.org/site/timebank/timebank.html

36. Pustejovsky J, Littman L, Saurí R, Verhagen M. TimeBank 1.2 Documentation. V2. Brandeis University. April 2006. http://www.timeml.org/site/timebank/documentation-1.2.html

37. TimeBank 1.2.
http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2006T08

38. TimeML 1.2.1 Specifications. October 2005.
http://www.timeml.org/site/publications/timeMLdocs/timeml_1.2.1.html