

Visualization of Pain Severity Events in Clinical Records using Semantic Structures

Clare T. Grasso and Anupam Joshi

Department of Computer Science and Electrical Engineering
University of Maryland Baltimore County
Baltimore, MD USA

Eliot Siegel

Diagnostic Radiology and Nuclear Medicine
University of Maryland School of Medicine
Baltimore, MD USA

Abstract—Physicians are often required to make critical medical decisions that may be based on previous events in the patient’s health history. However, these events may be very difficult to locate in the patient record due to the large volume of unstructured textual data in the patient’s chart. Even when the chart is housed in an electronic health record (EHR) system, keyword search within the chart may produce many results that are not relevant or that may overlook related expressions and concepts entirely. In addition, some medical events, such as the occurrence of symptoms, are associated with important attributes such as location or severity, and require other elements such as the type of clinical note and its date and time in order to provide the proper context of the event. This paper describes a prototype system that performs ontology-based semantic search through clinical text to extract pain severity events, and then presents them in a visualization to monitor the progression of pain over time.

I. INTRODUCTION

The advent of electronic health record (EHR) systems has made it possible to mine vast amounts of clinical data for use in medical research, surveillance, quality of care, and clinical decision support. However, much of the data is contained in unstructured text that is not easily consumed by software systems. Medical and clinical terminologies and ontologies exist to aid in identifying relevant terms in the text and to provide some semantic context concerning the taxonomy of those terms. However, events that occur in a health history are often characterized by a set of related attributes. For example, pain is characterized by its location, severity, quality, onset and duration. In addition, other important aspects concerning the context of the event are the date and time at which it occurred as well as the type of clinical note the event occurs in. In order to present a more complete semantic representation of the event, all of these components need to be included in the search results.

Currently, the Veteran’s Administration (VA) hospitals that use the VistA EHR have no way to perform any kind of search through a patient chart. Instead, the clinician is presented with a list, ordered by date and time, of all notes for the patient that are contained in the system. The clinician must click on each note and read through it to find any events of interest. However, the amount of text involved is considerable. For a patient that experienced a three-day stay in the hospital that included routine appendectomy, a total of 55 clinical notes

were generated for a combined total of 180 pages of single-spaced textual data (10-point font). In order to monitor how well the patient’s post-surgical pain was being controlled, the clinician would have to click each note individually and then read through the entire note to find any text relating to the patient’s experience of pain.

Physicians at the VA hospital in Baltimore wanted to quickly assess and monitor a patient’s post-surgical pain using a visualization that contained as much pertinent information from the chart as possible. This paper describes a proof-of-concept software system that uses ontology-based semantic search to extract pain event mentions from a patient chart, store them in a structured semantic representation in a knowledge-base, and then use the knowledge-base to retrieve data necessary to visualize patient pain severity information.

II. METHODS

A. Corpus

The corpus consists of a portion of the record of an appendicitis patient that ranges over a period of three days from the time of the patient’s arrival at the emergency department until post-surgical discharge. There are 55 different notes of which there are 27 unique note types including triage, emergency department, surgery, radiology, laboratory, and nursing assessments. Each note begins with a header stating its type along with the date and time it was entered into the system. Unless otherwise specified in the text of the note, all events occurring in the note are assigned the date and time of the note itself. The patient data was downloaded from the VA’s VistA [1] EHR which stores and outputs the data as 80-character lines of ASCII text.

B. Special Challenges

1) *Most of the pain related text in the corpus is not amenable to natural language processing (NLP) techniques that are used by most clinical text processing systems.* Except for consultant reports and notes, most of the text of the clinical notes is entered at the bedside during the course of patient care. Narrative, free-form text (such as would be seen in history of present illness section of a note) is characterized by its telegraphic expressions and non-canonical use of capitalization, punctuation and white space. Fig. 1 contains an example of this style of text.

```
ABD DISCOMFORT @ OLD SURGICAL SITE, ONSET  
THIS AM, DENIES SOB, DENIES C/P, PT. THOUGHT  
PROCESS OFF/CONFUSING THIS THOUGHTS, DR.  
XXXXXX MADE AWARE PT IN ACUTE AREA BED  
#5
```

Fig. 1. Example of non-canonical text entered by the clinician at the point of care.

```
Pain:  
Patient reports pain during shift: Yes  
Type of Pain: surgical  
LOCATION:abdomen  
QUALITY:sharp  
SCALE:6  
PATIENT'S ACCEPTABLE LEVEL OF PAIN:0 no pain
```

Fig. 2. Example of non-canonical text entered by the clinician at the point of care.

Additionally, much of the textual data relating to pain assessment appears in the form of questionnaires that are filled out on the clinician's screen and are then copied into the clinical note by the EHR in the same form as it appears on the screen. Fig. 2 contains an example of this. In this case, the expression of interest appears outside of the sentence it relates to, or does not appear within a sentence at all.

2) *Because this system will be used interactively and integrated into the existing system environment, it must perform in near real-time on commodity hardware and software systems:* In order to achieve the speed necessary to be useful in clinical decision support, the approach seeks to minimize the architecture by only including the necessary knowledge and tools needed for the particular event being extracted. To accomplish this, the architecture is made up of composable components that are snapped together into a minimal set. In this application, only the components needed for pain and its attribute of severity will be used. Components for location, quality, onset, and duration can be excluded.

C. Ontology Based Semantic Search

The semantic search and extraction of pain events follows the general approach of ontology-based data access over unstructured data. It is made up of three components: two complementary ontologies and an extraction algorithm. The first ontology contains the semantic representation of the event type and any other domain knowledge that might be needed for inference in clinical decision support. The second ontology is used for mapping the lexical forms in the text to the semantic form contained in the domain ontology. It contains the knowledge needed to identify the concepts as well as any other knowledge needed to interpret and normalize lexical or numeric expressions. It also identifies the specific extraction algorithm needed for that component as well as any other external resources such as other terminological systems. Lexical forms are identified and mapped using regular expressions

and/or dictionary lookup in standard terminologies.

In this case, the mapping ontology normalizes synonymous terms such as *hurt*, or handles special terms such as *headache* which also contains an anatomical location. Expressions indicating severity such as "a little" are normalized to standardized concepts contained in the domain ontology such as *Mild*.

Both the mapping and semantic ontologies can be extended for further specialization of components. For example, while most symptoms are able to use generic severity components that incorporate terms such as *None*, *Mild*, *Moderate*, and *Severe*, the pain severity mapping ontology is a specialized component that extends it to handle the numeric pain scale (0 to 10) as well as lexical terms (e.g., *cramp*) that are specific to pain. Composition and specialization of ontologies is accomplished using imports. A more detailed discussion of this approach is contained in [2].

In this application the minimal set of components needed are those representing basic patient information (name, id), clinical notes (note type, timestamp), pain, and the specialized pain severity.

The ontologies were expressed in the OWL 2.0 representation language and engineered using the Protg 4.1 development environment.

D. Extraction Engines

The master extraction engine reads in the mapping/semantic ontology pairs for each component that is being extracted and stores them in the knowledge base. The extraction engines for each component are then loaded and executed in turn. The engine for each component queries the mapping information that it requires from the knowledge-base in order to identify the lexical expressions for that component in the text. When an expression is found, the semantic representation of the concept is inferred based on the relationship between the mapping and the semantic ontologies. In this way, the semantic representation of the event is created, one component at a time, and stored as individual instances of the event in the knowledge-base. Once the extraction is complete, all components of the mapping ontology are removed from the knowledge-base in order to reduce inferencing time over the domain knowledge and events.

Because the text in this corpus is not amenable to NLP techniques, it must rely on proximity and scoping which is the basis of the Context algorithm as described in [3]. Proximity is determined by the distance between target terms, which are generally the main concept (in this case the symptom of pain), and modifier terms. When extracting events from the clinical text, modifier terms establish important aspects such as whether the event is being negated ("denies pain") or how much certainty the text expresses about the occurrence of the target ("may have..." or "does have"). The attributes of the symptom such as severity also function as modifiers, for example, "Pain: 4". Occasionally, attributes also declare as target terms, such as *level* or *score*, in order to extract statements such as "Scale: 3".

Patient: Appendicitis - Events: Pain Severity

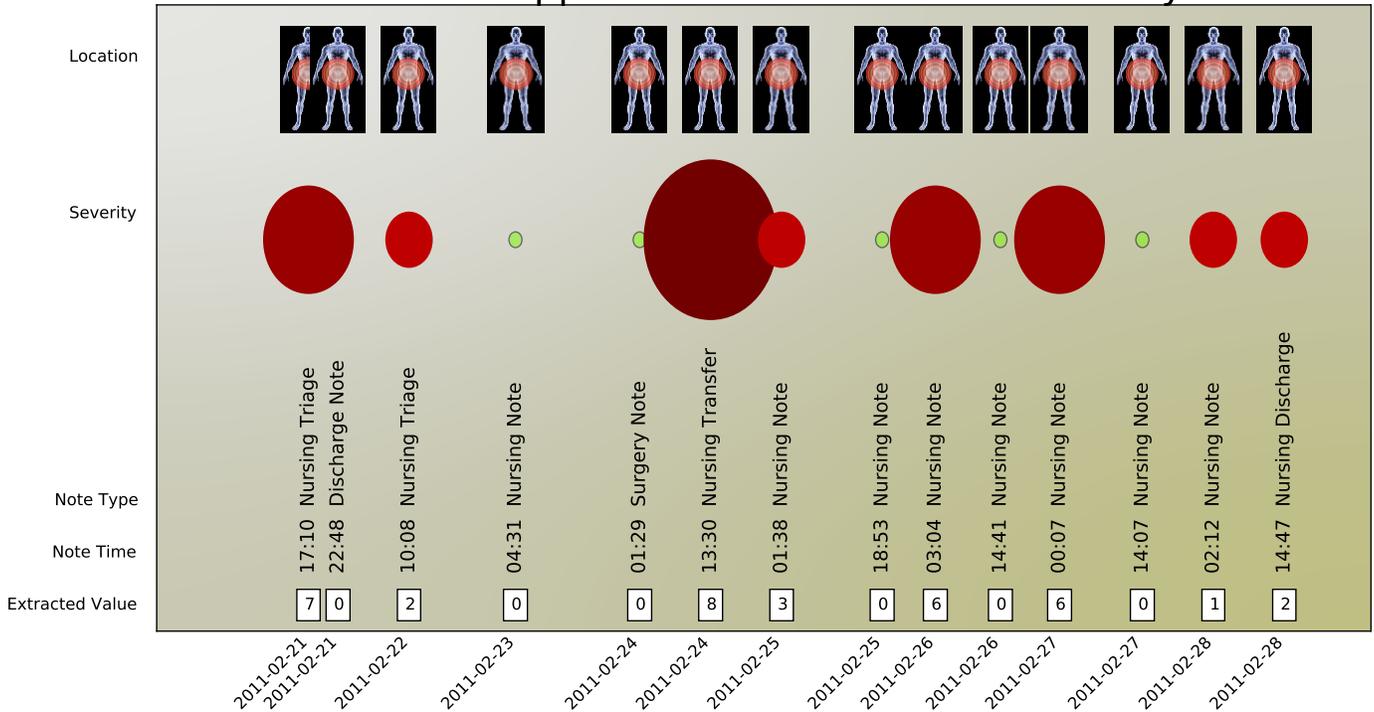


Fig. 3. Pain Severity Events Visualization Results

When working with well-formed text, the normal unit of scope is the sentence. However, this assumption does not hold on this corpus. This is partly because much of this text is not formed as sentences (as seen in Figure 2), and partly because the use of punctuation and white space in sentences in non-canonical text makes sentence segmentation extremely problematic (as seen in Figure 1). Therefore, scoping is determined by the end of the line, semi-colons, and some conjunctive terms such as *but*. The scope of target terms looks both forward and backward, while modifiers must indicate whether their scope proceeds from their location in the text to the beginning or end of the scope (i.e., either forward or backward). When a modifier term appears within the scope of a target term in the designated direction, a rule fires indicating that the modifier term is modifying the target term.

The extraction engines were developed in Python 3.5.

E. Knowledge Base

The mapping and semantic ontologies are loaded into an a knowledge-base. When events are identified in the text based on the mappings, individual instances of the semantic representation of the events are added to the store. A reasoner then uses an inference engine to apply all domain concept rules and relationships to the instances in order to generate any other inferred relationships. Inference is used to map the lexical and numerical severities to their normalized values. For instance, "minimal" would be inferred as *Mild*, while the numeric value of "7" would indicate *Moderate* pain. If a pain

event is negated ("no pain"), an inference is made that the pain severity is *None*.

As a prototype, proof of concept application, the owlready 0.2 [4] Python module was used to load the ontologies; manipulate classes, instances and annotations; and to re-classify instances automatically. Currently, the knowledge-base is stored in memory and the events are serialized to persistent storage as individual instances in OWL/XML representation. The Hermit version 1.3 OWL reasoner [6] was used for inferencing.

F. Visualization

Once the extraction is complete, the visualization software queries the knowledge-base for the pain severity events that were found in the text. The note title, date and time, and the normalized pain severity is retrieved for each event and ordered by date and time. The structured representation can then be used by the graphics library to create the visual representation.

The visualization presents pain severity over time. The horizontal dimension shows time as the number of hours that have passed since the date and time associated with the first note. The vertical dimension is made up of four stacked layers. The top layer shows the anatomical location of the pain and is represented by a bulls-eye figure overlaid on top of the image of a human body at the location of the pain. In the second layer, pain severity is represented as a red circle which is centered horizontally over the time of the event occurrence.

In order for the clinicians to ascertain the progression of pain as quickly as possible, the pain severity is represented in two different visual dimensions: 1) the size of the circle; and 2) the color of the circle. A small green circle indicates that the note explicitly indicated the absence of pain. If the existence of the pain event was indicated as being *Definite*, but the severity was not stated ("Patient states he does have pain at this time"), a question mark is displayed instead of the circle. The third layer contains text which gives more specific information about the context. This includes the actual date and time of the event, and the type of note it occurred in. The bottom layer is the actual pain value extracted from the clinical text.

III. RESULTS

The results in Figure 3 show the final visualization of the progression of pain for the appendicitis patient. In this case, the anatomical location of the pain was known in advance to be abdominal and was sent in to the visualization as a parameter. The visualization shows the patient arriving at the emergency department with moderate pain. Initially, the pain resolves and the patient is discharged. However, the patient is called back after a positive diagnosis of appendicitis for which he receives surgery. The immediate post-surgical pain is severe, and oscillates between none and moderate due to treatment with pain medication. The patient is finally discharged with mild pain.

As stated before, the patient history visualized here consisted of 55 clinical notes for a total 180 pages of single-spaced textual data. The application was run (single-threaded) on a commodity Intel i7 1.8 GHz processor. Time to perform the extraction over this corpus was 1.926 (95%C: +/-0.035) seconds. Visualization time to read and display the results was 0.808 (95%C: +/-0.007) seconds. Total wall-clock processing time, therefore, was under three seconds which meets the near real-time requirements necessary to be useful in clinical decision support.

Initial results for the accuracy of the extraction were reported in [2] in which they are described in detail. In that work a reference standard was created by three physicians working independently. For the pain severity values, both lexical and numeric, the inter-annotator agreement had an observed agreement of $A_o = 0.79$ (79/100), an expected agreement of $A_e = 0.184$, and Fleiss's $\kappa = 0.743$ which is considered substantial.

Results were evaluated using the vertical metrics for measuring system performance on individual fields as outlined in [5] and consisted of phrase-level precision, recall and F-measure. (*line number*, *severity value*) pairs were compared against the gold standard. Only exact matches were considered correct. This resulted in a positive precision of 90.31% (233/258), a positive recall of 88.26% (233/264), and a positive F1-score of 89.27%.

In the effort described here, the extraction algorithm was updated to address the two main weaknesses of the prior work. The first weakness regarded pain mentions that were broken

by new line characters. While in well-behaved, grammatically clean text this could be done trivially by combining two adjacent lines, this approach resulted in a reduction in precision of almost 20% in the VA corpus. Instead, the best results were obtained by combining the current line with the first two alphanumeric tokens from the following line. The second weakness that affected recall was when a pain score was associated with body locations instead of to a target term, for example, "7.5 both hands." In this case, if a numeric pain severity was found in the absence of a target term, a dictionary lookup was performed on each alphanumeric token within a window of four terms in each direction of the pain score. The dictionary was made up of word tokens that were taken from terms contained in the Foundational Model of Anatomy ontology. These modifications resulted in a positive precision of 92.00% (230/250), a positive recall of 94.65% (230/243), and a positive F1-score of 93.31%. Most of the remaining recall errors are caused by not recognizing lexical severity values ("mildly"), misspellings ("dnied"), and missing whitespace ("Chest Pain-denies").

IV. DISCUSSION

The two biggest problems in the visualization were 1) the inability to stretch or scroll the graph and 2) to adjust the visualization to the density of the pain severity event data. In the future, the use of graphical gaming libraries might address this and allow the clinician to interact with the image in order to scroll, zoom, or stretch the image. It might also allow the clinician to hover over a data point in order to see the actual text with the extracted values highlighted. A more complete user interface for search would allow the clinician to refine the search results and to filter on aspects such as date, symptom type, or note type. If multiple symptoms are displayed, it may also be possible to layer them while allowing the clinician to selectively overlay them with each other. Other important related data that might be displayed are medication events that are used to treat the symptoms.

ACKNOWLEDGMENT

This work was supported in part by NSF Grant IIS-0910838 and by the Department of Defense through a supplement to IIP-0934364.

REFERENCES

- [1] Worldvista. <http://www.worldvista.org/>, accessed June 2015
- [2] C. Grasso, A. Joshi, E. Siegel, Beyond ner: towards semantics in clinical text, in International Workshop on Biomedical Data Mining, Modeling, and Semantic Integration (ISWC-BDM2I 2015), in press.
- [3] H. Harkema, J. Dowling, T. Thornblade, W. Chapman. Context: an algorithm for determining negation, experienter, and temporal status from clinical reports. *J Biomed Inform* 42(5), 839–851 (2009)
- [4] JB Lamy, H Berthelot. Ontopy : programmation oriente ontologie en Python. Actes du congrs d'Ingierie des Connaissances (IC2015) 2015. <https://hal.archives-ouvertes.fr/hal-01168286>
- [5] O Uzuner, I Solti, E Cadag. Extracting medication information from clinical text. *J Am Med Inform Assoc* 17(5), 514–518 (2010)
- [6] B Glimm, I Horrocks, B Motik, G Stoilos, and Zhe Wang. Hermit: an owl 2 reasoner. *J. Autom. Reason* 53(3), 245–269 (2014).