Clinical Tractor: A Framework for Automatic Natural Language Understanding of Clinical Practice Guidelines

Daniel R. Schlegel, PhD[1], Kate Gordon[1], Carmelo Gaudioso, MD, PhD[2], Mor Peleg, PhD[3]
[1]SUNY Oswego, Oswego, NY, USA; [2]Roswell Park Cancer Institute, Buffalo, NY, USA;
[3]University of Haifa, Haifa, Israel

Abstract *Formalizing plain text clinical practice guidelines (CPGs) into computer interpretable guidelines (CIGs) manually is a laborious and burdensome task, even using CIG tools and languages that have been designed specifically to improve the process. Natural language understanding (NLU) systems employ techniques from computational linguistics to perform automated reading comprehension, parsing text and using reasoning to convert syntactic information from unstructured text into semantic information. Influenced by successful systems used in other domains, we present the architecture for a system which performs this task on CPGs, creating semantic representations of entire guidelines. The future goal of the system is that this semantic representation may be used to generated CIGs.*

1 Introduction

Clinical practice guidelines (CPGs), as they exist today, form the basis for evidence-based medicine. Most CPGs are distributed in a natural language (*e.g.,* English) text document by professional organizations such as the American Diabetes Association (*e.g.*, [1]). It is well known that compliance with paper guidelines is lacking, but that compliance improves greatly with the introduction of clinical decision support systems (CDSS), which implement guideline recommendations and are integrated into electronic health record (EHR) systems (see, *e.g.,* [2]).

Over the past two decades, methods and formalisms have been developed for representing guidelines computationally as Computer Interpretable Guidelines (CIGs, see [3]). Unfortunately, there are still very few actively maintained CIGs since the process of creating them is extremely time-consuming and burdensome. Further complicating matters, the creation of CIGs is not a one-time effort; since guidelines for each individual disorder are often produced yearly, and may each be hundreds of pages, keeping CIGs up to date is a significant problem. Some guidelines do provide information about what has changed from one version to the next, so an individual or team could use these highlighted changes to try to keep CIGs up to date. However, these periodic updates are often wide ranging and require significant re-engineering of the CIG.

Because of the lack of easily created and maintained CIGs, hospital systems have individually implemented large numbers of CDSS rules in an ad-hoc manner. As EHRs are increasingly used in rural hospitals and clinics without the funding for large health IT teams, there is a growing discrepancy in the CDSS used, and the resulting care provided (see, *e.g.,* [4]). A method to automatically generate CIGs from CPG text could go a long way toward building up-to-date CDSS, available to all levels of care organizations.

In order to automatically generate CIGs accurately and comprehensively, we believe it is necessary to represent the semantic content of CPGs to the greatest extent possible (See Section 5 for a discussion of related work supporting this belief). This semantic representation may then be used to create the associated CIG. We present a framework based on natural language understanding (NLU) techniques to perform this semantic extraction.

Natural language understanding is a subtopic of natural language processing in which the goal is to build a computer system which performs reading comprehension on a given input text. These techniques are not widely used in the biomedical informatics community in part because the language used is complex, presupposing a significant amount of implicit knowledge. There is also a need for high precision due to the safety-critical domain. Implementing custom tools to perform the NLU task while addressing these issues requires wide-ranging expertise (biomedicine, computational linguistics, and knowledge representation and reasoning) and can be quite laborious.

The framework presented here adopts its high-level design from a previous NLU system, *Tractor*, designed for understanding short intelligence messages in the counter-insurgency (COIN) domain [5, 6, 7], and adapts it to the clinical domain. The Tractor system was successful in its task – it converted input text to a knowledge base (KB) containing over 92% semantic relations using rules that fired correctly nearly 98% of the time [6]. In an article currently in preparation we report the transformation to be on par with what a human is capable of performing. Our new system, currently under active development, is dubbed *Clinical Tractor*.

We begin with a discussion comparing the domains of Tractor and Clinical Tractor to illustrate that adopting some of the Tractor framework is reasonable, while also showing where the differences lie. Clinical Tractor's architecture is detailed in Section 3 along with a worked example in Section 4. We conclude with a discussion of related work (Section 5), and the future of the Clinical Tractor project.

2 A Comparison of the Domains of Tractor and Clinical Tractor

Tractor was initially developed for the COIN domain, requiring a large portion of reality to be modeled. Persons about whom intelligence messages are written are usually performing the activities of daily life – shopping, driving, making phone calls, interacting with other persons, carrying items, *etc*. The persons and items are described in varying amounts of detail. Problematically, it is unknown in advance which of these activities or attributes will be important when the messages are combined to form a complete picture of what is happening in an area. This uncertainty forced Tractor to be developed in a highly general way, so as to model a large number of activities and attributes at once, modeling specifics only where the general models were insufficient.

In this regard the domain of clinical medicine is significantly simpler. In general there is only a single person being discussed, the patient (though discussions of family history may also be present). In guidelines there is another person, the clinician, who is asked to perform some actions. The attributes of import and the actions that are or should be taken encompass only those related to health, not all of reality. A significant advantage to working in the clinical domain is the existence of a wide variety of controlled vocabularies, terminologies, and ontologies, which allow the identification of a large number of these actions and attributes. The strategy to model these actions and attributes can be adopted from Tractor with little modification, in fact the general rules used for modeling many activities and attributes at once can be used with little or no modification.

Clinical guidelines have the advantage of containing, in general, grammatical text. Intelligence messages, like medical records, do not share this property – they often contain sentence fragments, semi-structured components, and unconventional punctuation/abbreviations. The Tractor system was built to be somewhat resilient to these issues, using only surface features where possible, working around mistakes made by the linguistic parsers in non-grammatical portions of text, and containing a system for specifying the components of semi-structured text. With CPGs we do not anticipate significant issues of this form, except perhaps in inclusion criteria where it is present, but it will be significant in planned future extensions of the work to include EHR data. Guidelines do contain some structured components in the form of document structure, which we account for.

Whereas intelligence messages are a record of what *has* happened, CPGs suggest what is to happen *next* in the form of recommendations. This is significant as recommendations in CPGs often contain modal verbs, qualifying action phrases with words such as "should", "may", "might consider", and so forth, in general covering the modalities of likelihood, ability, permission, and obligation. These, importantly, provide a weight to the recommendation. Weights also may be derived from the degree of evidence upon which the recommendation is based, usually provided on a scale somewhere in the guideline.

Intelligence messages and individual CPG recommendations are both similar in that they are short, avoiding issues such as topic shift and rhetorical/discourse relations. On the other hand, CPG recommendations often have temporal semantics dictated by their order. Within sections of the narrative of a guideline topic shift is generally avoided, and some discourse relations that arise in storytelling are eschewed. There is the potential for sections to exhibit rhetorical relations such as narrative strengthening [8], though we do not believe this requires any architectural additions.

3 Clinical Tractor

Because of the domain differences, the architecture of Clinical Tractor is different from that of Tractor, with more of a focus on extraction of data based on document structure, and making use of background knowledge. The architecture, seen in Figure 1, consists of four main components: text processing using various *processing resources* (PRs) operating within the open-source General Architecture for Text Engineering (GATE) [9]; converting the GATE output to a

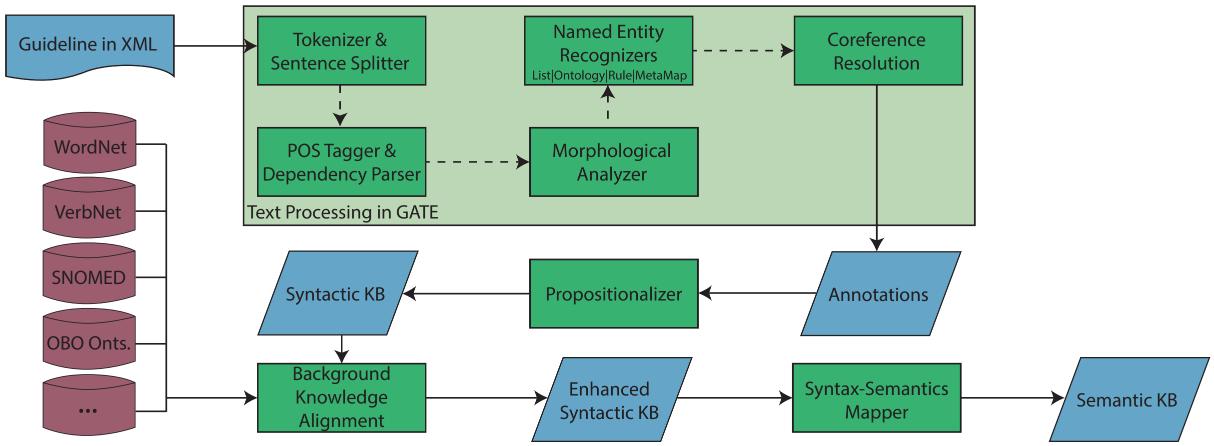

Figure 1: Clinical Tractor system architecture. English text is processed through a natural language processing pipeline in GATE. The annotations from GATE are converted to a knowledge base, enriched with background knowledge, then converted to a semantic knowledge base.

syntactic KB consisting of propositions in a first order logic; aligning terms in the KB with background knowledge and importing relevant data; and mapping syntactic relations to semantic relations using both domain-neutral and domain-specific mapping rules informed by the background knowledge.[1]

3.1 Input Data

Guidelines are distributed in several formats. In order to standardize them for our pipeline, we manually convert the guidelines of interest to an XML format capturing the document structure such as headings, tables, and inset boxes, and including the graph structure in figures and algorithms (as in the NCCN guidelines). No guideline-specific semantic features are included. The XML format used is based on a combination of the Journal Article Tag Suite (JATS) [10] and GraphML [11]. In the future this transformation would either be an automated process, or, in a more ideal future, guidelines would be distributed in (possibly one of several) standardized format(s).

3.2 Text Processing in GATE

Each input CPG is processed by a set of PRs operating within GATE. Most of these PRs are from the ANNIE (a Nearly-New Information Extraction System) suite [9]. Shown in Figure 1 are: the ANNIE English Tokenizer and Sentence Splitter that divide the input into linguistic units; the Stanford Dependency Parser, for part-of-speech tagging and parsing (discussed further in Section 3.2.1); the GATE Morphological Analyser for finding the root forms of inflected nouns and verbs; a group of named-entity recognizers – list based, ontology-based, rule based, and using MetaMap (discussed further in Section 3.2.2); and a group of PRs that perform co-reference resolution. GATE uses a plugin architecture allowing for the use of many other PRs as well as the creation of custom PRs. It also allows customization of each of the selected PRs according to the domain.

3.2.1 Dependency Parsing

The notion of dependency relations in language is ancient, going back to Pāṇini's grammar in the 5th century BCE. Phrase structure grammar which is more commonly covered in introductory linguistics courses, on the other hand, is a modern invention. Dependency grammars represent syntactic structure as (often binary) relations between tokens in the text. These relations are known as dependencies. The Universal Dependencies [12] used in the Stanford Dependency Parser contain mostly syntactic relations, but also relations consistent with a shallow semantic parse. Dependency parses contain more semantic data than that afforded by phrase structure parses. This semantic information makes the task of developing syntax-semantics parsing rules to determine semantic roles somewhat easier.

[1] We outline the framework making use of tools we have selected, but these tools could be swapped for others which perform the same tasks.

3.2.2 Named Entity Recognition

Named entity recognition (NER) is a component often associated with information extraction (IE) systems in which structured data is extracted from free text for one or more classes of entities. These classes often include the names of persons, locations, and organizations, but also dates, addresses, quantities, etc. As discussed above in Section 2, the person entities in a guideline are fairly straightforward: there is the patient, perhaps some family of the patient, and the clinician. On the other hand, there are very large classes of entities such as drugs, procedures, diseases, symptoms, and anatomical locations. There are also a significant number of entities related to measurements and temporal relation. Guidelines also include evidence levels represented in various forms. To appropriately recognize these we make use of several forms of NER: list-based, ontology-based, rule-based, and we also use MetaMap for NER.

- **List-Based NER** GATE contains a "gazetteer" PR for identifying entities from lists. Lists may contain complete named entities such as names or locations, or words (keys) which given context can indicate that a named entity begins with or ends with the key (e.g., "Hospital" in a hospital name or "Jr." in a person's name). A key type of great importance in CPGs is that which indicates the current sentence, paragraph, or section is or contains recommendations. Gazetteer items have a major and minor type, allowing for a shallow ontology.
- **Ontology-Based NER** Related to list-based NER is ontology based NER. Terms and their synonyms are identified in the text through simple matching. We have developed tools which extract this data from ontologies to store them in gazetteer lists for matching. By storing them in lists, additional synonyms can easily be added. without modifying the underlying ontology.
- **MetaMap** Probably the most popular method for recognizing terms from medical vocabularies in text, and making use of the UMLS, MetaMap is sometimes criticized for its precision/recall. In combination with other approaches, it can be a useful addition to a complete NER suite.
- **Rule-Based NER** Rules allow identification of named entities through regular expressions over annotations using the Java Annotation Patterns Engine (JAPE). These rules allow for recognition of complete entities for which keys were noted in the list-based NERs. Entities with semi-structured formats such as prescription drugs may also be recognized. Rules provide an opportunity for a first pass at disambiguation and the removal of over-matches given the context available in word orderings.

Downstream processing in the syntax-semantics mapping rules makes use of the dependency parse to perform the bulk of the NLU task. One source of confusion in designing an NLU system of this type is how much NER to do using rules and lists, and how much to do later using the dependency parse. Dependency parsing captures structural relationships (dependencies) well, but recognition based on word order given a dependency parse is quite difficult. Therefore we limit ourselves to recognition which is word-order dependent at this stage.

3.3 Propositionalizer

The result of GATE processing is a set of *annotations*, each consisting of an identifier, a start and end position within the CPG's text, a type, and a set of attribute-value pairs. Each of GATE's PRs produces these annotations, so the set consists of information about tokens, sentences, paragraphs, dependencies, results of the NERs and so on. Multiple annotations may span the same region of text. Document structure from the input XML documents is also reported as annotations.

Given the input from GATE, the propositionalizer merges annotations which have the same start and end positions (*e.g.*, a token and one or more results from the NERs). The result of this is a set of annotations each with unique start and end positions, and each with a unique identifier. The propositionalizer re-constructs the hierarchy of document-related XML tags and produces logical assertions in a form subsuming that of DoCO, the Document Components Ontology [13]. In addition to what DoCO offers, head words of sentences (found via the dependency parse) are attached to the sentences for use in the syntax-semantics mapper.

The propositionalizer produces a KB, consisting of a set of propositions (expressions which may have a truth value assigned to them), in the logical language of the CSNePS knowledge representation and reasoning (KRR) system [14, 15, 16]. CSNePS is used to represent and perform reasoning on all of the KBs created by Clinical Tractor from the English CPGs. CSNePS, like its predecessor SNePS 3 [17], is simultaneously a logic-, frame-, and graph-based

KR system [18]. It is the latest member of the SNePS family of KR systems [19].

CSNePS uses a term logic, in which all expressions are terms – even those that in first order logic would not be. This means that propositions may have propositions as arguments (allowing for meta-knowledge). We will use "assertion" to refer to a proposition that is taken to be true in the KB, and say "assert a proposition" to mean adding a proposition to the KB as an assertion. We will also speak of "unasserting a proposition" to mean removing an assertion from the KB. The arguments of a proposition are terms that could denote words, occurrences of words in the guideline (*tokens*), syntactic categories, entities and events in the domain, classes (also referred to as "categories") of these entities and events, or attributes of these entities and events.

We can classify relations, and the propositions in which they occur, as either: *syntactic*, taking as arguments terms denoting words, tokens, and syntactic categories; or as *semantic*, taking as arguments entities and events in the domain and their categories and properties. A KB is syntactic to the extent that its assertions are syntactic, and is semantic to the extent that its assertions are semantic. The KB produced by the propositionalizer is mostly syntactic, and hence we refer to it as the *syntactic KB*.

3.4 Background Knowledge Alignment

The syntactic KB is enhanced by a background knowledge alignment system (BKAS). This system matches spans of text against lexical resources such as WordNet and VerbNet, and locates ontological terms based on the results of NER in SNOMED (via MetaMap CUIs) and biomedical ontologies. The matched data is imported into the KB. Where the data is hierarchical as in the WordNet hypernym hierarchy, the VerbNet hierarchy, and ontological subsumption hierarchies, relevant hierarchies are imported into the KB. Where other logical relations are present, those are imported as well.

Clinical Tractor is designed to operate in an ontologically heterogeneous environment, in which a single concept in the text may be annotated with multiple ontological terms from different sources. But, the more entirely separate sources there are, the more complex downstream processing in the mapping rules becomes. Therefore, in order to minimize this, we make heavy use of the OBO Library ontologies which have been co-developed to be inter-operable.

The BKAS is meant to be generic, allowing for the easy addition of whatever resources are needed. In future work, we intend for this to include knowledge extracted from other materials such as journal articles and other guidelines. The result of the BKAS is an *enhanced syntactic KB*.

3.5 Syntax → Semantics Mapper

The enhanced syntactic KB is operated on by the mapping rules, converting the mostly syntactic KB to a mostly semantic representation. Whereas IE approaches aim to identify "within text instances of *specified classes* of entities and of predications involving these entities" [20, emphasis added], we aim to convert the entire syntactic content of the guideline into semantic content, doing true automatic reading comprehension. This includes understanding of all parts of the text, not only verb relations or noun phrases matching some pre-specified patterns as is done in other systems (see Section 5). The mapping rules are represented in the CSNePS rule language and are executed within the CSNePS KR system.

The mapping rules, designed to be general, come in two major types – those that convert syntactic representations to more easily processable syntactic representations, and those that convert syntactic representations to semantic ones. The left side of Figure 2 shows a rule that simplifies syntactic representations, transforming phrases in the passive voice to the active voice. This rule *fires* (*i.e.*, is executed) when an `nsubjpass` (passive nominal subject) relation is identified in the dependency parse. This relation occurs between a verb and its passive subject. It converts this into a `dobj` (direct object) relation and unasserts the `nsubjpass` relation. The rule also looks to see, in a subrule, if the verb is in a prepositional relation (`prep`) with the word "*by*", and makes the nominal subject of the verb the object of the prepositional relation. This rule would transform the parse of "*morphine should be prescribed to the patient*" to the parse of "*the patient should be prescribed morphine*". In building NLU systems, the number of rules can quickly grow out of hand; rules such as this simplify the problem somewhat by requiring no special rules to be written for

Syntax -> Syntax Rule

```
(defrule passiveToActive
  (nsubjpass ?verb ?passsubj)
  =>
  (assert `(dobj ,?verb ,?passsubj))
  (unassert `(nsubjpass ,?verb ,?passsubj))
  (:subrule
    (prep ?verb ?bytok)
    (TextOf by ?bytok)
    (pobj ?bytok ?subj)
    =>
    (assert `(nsubj ,?verb ,?subj))
    (unassert `(prep ,?verb ,?bytok))
    (unassert `(pobj ,?bytok ,?subj))))
```

Syntax -> Semantics Rules

```
(defrule dobjAction
  (dobj ?action ?obj)
  (Isa ?action Action)
  =>
  (assert `(theme ,?action ,?obj))
  (unassert `(dobj ,?action ,?obj)))

(defrule dobjPerception
  (dobj ?perception ?obj)
  (Isa ?perception Perception)
  =>
  (assert `(topic ,?perception ,?obj))
  (unassert `(dobj ,?perception ,?obj)))
```

Figure 2: Example mapping rules. The left rule simplifies the syntax of a phrase, converting the passive voice to the active voice. The right rules convert syntactic `dobj` relations to semantic `theme` and `topic` relations.

handling passive phrases.

The right side of Figure 2 shows two syntax-semantics mapping rules. The first of these, `dobjAction`, would make *the patient* the `theme` of the 'prescribe' action in the previous example. The relation `theme` reflects one of the linguistic thematic relations [21, 22, 23], often used to express the action of a verb. This rule fires when the verb, *prescribe* in this case, is a member of the class `Action`. This is derivable from background knowledge sources. The `dobjPerception` rule fires when the verb is a `Perception` action, a more specific case than `dobjAction`. A verb would be known to be a perception by making use of imported data from the BKAS, such as VerbNet. In this case the `topic` thematic role is used. For example, a guideline might contain the text "... *when complications are discovered.*" Here *complication* is the `topic` of *discover*. Determination of the thematic roles to use is done by making use of the Unified Verb Index [24].

Background knowledge sources play an important role in the mapping rules. The lexical relations available from WordNet and VerbNet allow the creation of general rules which are specific to the kinds of things discussed in guidelines. While an above example makes use of the "*prescribe*" verb, many others could be used (*e.g.*, receive, take, be given). In general, these verbs have some medication or treatment as their direct object, and indicate a transference of ownership. Verbs of this type are covered by a small set of upper-level concepts in the lexical resources, which may be used in the mapping rules. While our goal is to use general rules wherever possible, we will use more specific rules as discussed here when necessary. Using this technique, the rules can identify many of the "Action Palette" [25] action types used in guidelines.

Clinical Tractor also aims to understand noun phrases, including those that otherwise might be given only a single code by an NER system. Using a common, consistent, semantic structure exposes the relation between long expressions which may not have a code, such as "cellulitis of left hallux" and shorter expressions (e.g., "cellulitis") and other long expressions that do have codes, such as "cellulitis of toe of left foot". Negation may also be understood using mapping rules.

The development of Clinical Tractor involves the creation of new domain-neutral rules and many more domain-specific rules. As discussed above, CPGs discuss choices and decision making, often including evidence in the form of in-text citations, statements of evidence level, and modality. Guidelines also provide plans for treatment, or give guidance on creating such plans. They also include conditions for performing actions. Rules for understanding these are part of the syntax-semantics mapper.

4 A Worked Example

Consider the following recommendation from the ADA Standards of Medical Care in Diabetes 2017 [1]: "*Patients found to have elevated blood pressure should have blood pressure confirmed on a separate day.*" To illustrate the

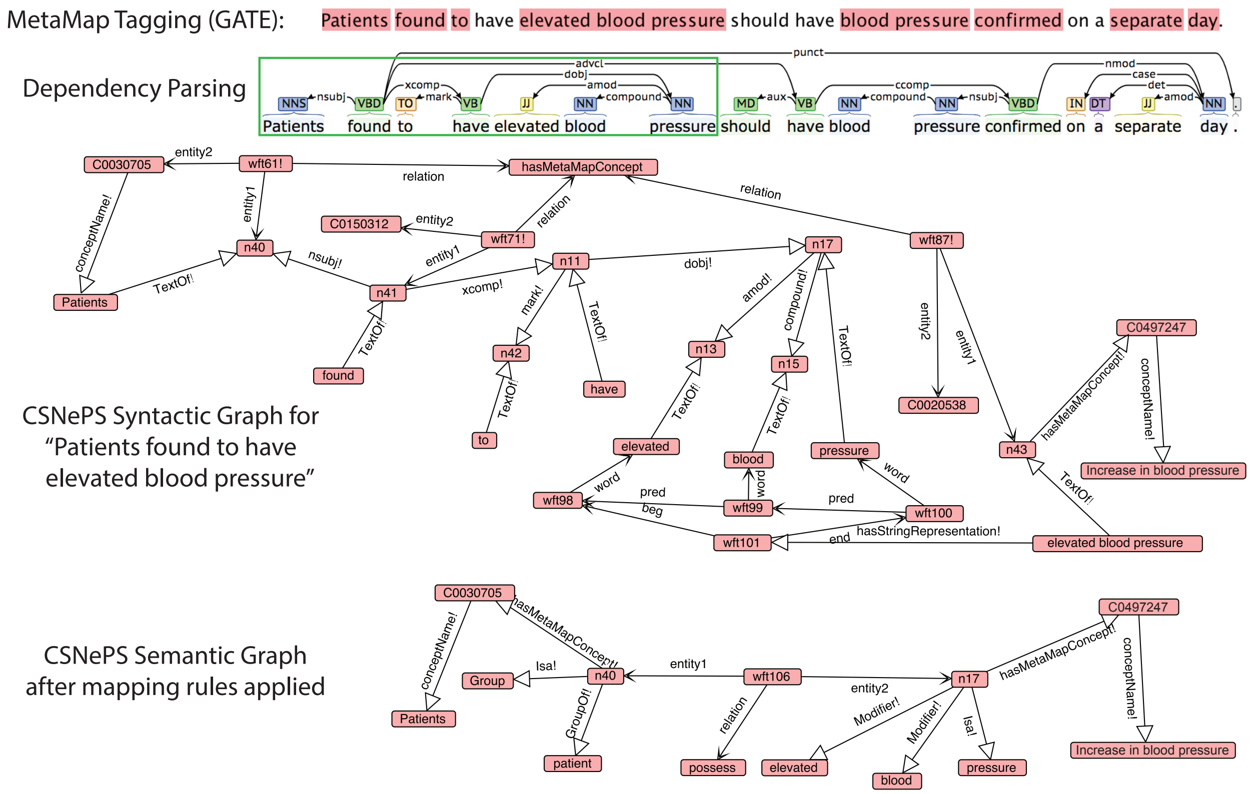

Figure 3: A small example using a prototype implementation of Clinical Tractor, with some by-hand augmentation. MetaMap matches and the dependency parse for a single recommendation are shown at the top, with a subset of the CSNePS syntactic KB after propositionalization, and semantic KB after the mapping rules have been applied shown for the phrase “*Patients found to have elevated blood pressure*.”

pipeline, this was processed through an early prototype of the system with limited NER (using only MetaMap), and without BKAS. The output was then modified to account for unfinished components, for illustrative purposes.

The processing results are shown in Figure 3. Identified concepts in MetaMap are shown on top, with the Stanford dependency parse directly underneath. The propositionalizer converted the GATE output to a CSNePS KB, and we've visualized it as a *propositional graph*. Portions of the KB for the text “*patients found to have elevated blood pressure*” are shown. Each token can be seen attached to its identifier (beginning with `n`). Dependency relations may be seen in the graph. The string representation of the multi-word expression “elevated blood pressure” and its decomposition into single words can be seen at the bottom right of the syntactic graph. MetaMap CUIs and concept names for some of the tokens are also shown. This graph excludes many additional relations that are in the KB for easier readability.

The semantic graph is shown at the bottom of the figure; the result of the mapping rules. A subtle change that has occurred is that `n` terms that originally denoted syntactic entities now denote semantic entities. Previously `n40` denoted a token with the text “*patients*”. Now it denotes a group of entities, each of which is of the type `patient`. `n17` was a token with the text “*pressure*”, adjectivally modified by “*elevated*” but now denotes an entity of type `pressure`, with the modifiers `elevated` and `blood`. The MetaMap concept `Increase in blood pressure` applies to this entity instead of the string “*elevated blood pressure*” as it did in the syntactic graph. This entity is possessed by `n40`. In sum, this graph represents the group of patients possessing elevated blood pressure.

Only two changes to the semantic graph were handled manually for this example. Two MetaMap concepts were identified for “*elevated blood pressure*” - a disorder (hypertensive disease), and a finding (increase in blood pressure). We manually selected the finding concept, though the process for selecting the correct one is already well defined: “found” is past tense of “find”, a member of the verb frame for “discover” (using VerbNet) which indicates a clinical observation; a finding. Second, we moved MetaMap concepts for multi-word expressions to the head noun.

5 Discussion and Related Work

Clinical Tractor shares characteristics with many systems used in biomedical informatics for language processing tasks. Guideline-focused work tends to be centered on the task of aiding the creation of CIGs by performing IE tasks to retrieve, and possibly restructure, salient portions of the CPG. Several examples make use of semantically informed patterns over the text. Wenzina and Kaiser [26] use patterns over UMLS semantic types to identify condition-action sentences. They observed recall of 75% and precision of 88% on a small evaluation set. In other work, Kaiser, *et al.* worked to identify treatment activities in guideline text [27]. Here they used the UMLS semantic network types and relations to generate semantic patterns for activities such as performing, using, and analyzing. They made use of lists of verbs corresponding to the relations and a dependency parse to determine which MetaMap identified concepts in the sentence fit the the subject and object of the relation.

Serban and colleagues [28] presented an ontology-driven method for pattern matching on frequently recurring linguistic patterns, mapped to the control structures (*e.g.*, sequencing, if-then, action-effect) of the target CIG formalism. Medical thesauri have been used [29] to enhance the ability to identify portions of guideline text which map to reusable building blocks, useful for guideline formalization.

Machine learning techniques have also been applied to the recommendation identification task. Preliminary work extracting regular-expression-based heuristic patterns has shown some promise [30], but the inclusion of semantic data is needed. Other work has used part of speech tags as features to extract action sentences from CPGs [31], but again, without the use of semantic data. Neither of these approaches were particularly tailored to the CPG domain. While not using machine learning approaches, Taboada *et al.* [32] provide evidence for the need to tailor systems to the domain at hand. They used several off-the-shelf tools to extract descriptive knowledge about theraputic and diagnostic procedures, finding that adaptation of the tools to the task improved results, though their paper doesn't make tailoring vs. non-tailoring directly comparable.

It's important to note that in isolation each of the above systems cover only a small subset of what is necessary to derive the complete semantics of the recommendations of a CPG, let alone an entire CPG. In considering the approach taken by Clinical Tractor, there are similarities to the pattern-matching approaches. The primary difference is that instead of using UMLS resources combined with mostly surface structure, Clinical Tractor aims to make significant use of the dependency parse, other kinds of NER, additional lexical resources to enhance generalization, and rules applied in multiple steps to move toward a completely semantic representation. This goal of a complete semantic representation appears to be unique to Clinical Tractor in the domain of CPGs.

In relying on, at least in part, a more "standard" NLP pipeline, Clinical Tractor also shares some characteristics with systems such as cTAKES [33], CLAMP [34], HITEx [35], and HTP-NLP [36]. The domain in which these tools are most commonly used is slightly different: information extraction for electronic health records (EHRs). While these domains have many similarities such as including some narrative structure, mentioning many of the same kinds of named entities, and a patient-centric focus, there are differences in content and appearance of the language. Medical records tend to contain more non-standard abbreviations, sentence fragments, and non-standard English. While only occasionally appearing in guidelines themselves, clinical protocol eligibility criteria do often share the property of being made up of sentence fragments. Work on ERGO [37] discusses many of the challenges involved in eligibility criteria, including acronyms, Boolean operators, and comparison statements. We have in-progress work addressing these issues mostly during the rule-based NER stage of Clinical Tractor. Even given the challenges of this type of text, we don't believe there's a need to go as far as to use IE components which are meant explicitly to account for such problems, such as NegEx, which appear frequently in EHR-focused pipelines using the above tools.

6 Conclusion

We have presented an architecture for an NLU system meant to perform, as near as possible, complete reading comprehension of CPGs. Our architecture makes use of a first-stage which is comparable to what many NLP systems in biomedicine perform, though with enhanced NER capabilities. It is unique in the attention paid to aligning background knowledge with the textual contents and going a step further than pattern matching rules over text, making use of a KRR system and syntax-semantics mapping rules to transform a syntactic KB into a semantic one utilizing NLU techniques. This approach is built upon that taken by Tractor, which has already shown to be successful in another domain.

Clinical Tractor is currently under active development as part of a larger system for the automatic generation of CIGs from CPGs. Our hope is that this systems proves useful to people working in and researching biomedicine, and that over time we can build a compendium of semantically represented knowledge. Our even longer term goal for Clinical Tractor is to generalize it to work wherever there is text in biomedicine, whether it be in guidelines, EHRs, journal articles, or clinical trial protocols. Moreover it is important to us that what we build be free for the world to use; as components of our system reach a usable state they will be released open source, under a non-restrictive license.

Acknowledgements

This work was supported by the National Library Of Medicine of the National Institutes of Health under Award Number R15LM013030. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

References

[1] American Diabetes Association. Standards of medical care in diabetes. *Diabetes Care*, 40(suppl 1), 2017.

[2] Lorenzo Moja, Koren H Kwag, Theodore Lytras, Lorenzo Bertizzolo, Linn Brandt, et al. Effectiveness of computerized decision support systems linked to electronic health records: a systematic review and meta-analysis. *American journal of public health*, 104(12):e12–e22, 2014.

[3] Mor Peleg. Computer-interpretable clinical guidelines: a methodological review. *Journal of biomedical informatics*, 46(4):744–763, 2013.

[4] Jordan Mitchell, Janice Probst, Amy Brock-Martin, Kevin Bennett, Saundra Glover, and James Hardin. Association between clinical decision support system use and rural quality disparities in the treatment of pneumonia. *The Journal of Rural Health*, 30(2):186–195, 2014.

[5] Michael Prentice, Michael Kandefer, and Stuart C. Shapiro. Tractor: A framework for soft information fusion. In *Fusion 2010*, page Th3.2.2, 2010.

[6] Stuart C. Shapiro and Daniel R. Schlegel. Natural language understanding for soft information fusion. In *Fusion 2013*. IFIP, July 2013. 9 pages, unpaginated.

[7] Stuart C Shapiro and Daniel R Schlegel. Use of background knowledge in natural language understanding for information fusion. In *Fusion 2015*, pages 901–907. IEEE, 2015.

[8] Jerry R Hobbs. On the coherence and structure of discourse. Technical Report CSLI-85-37, Center for the Study of Language and Information, Stanford University, 1985.

[9] H., D. Maynard, K. Bontcheva, and V. Tablan. GATE: A framework and graphical development environment for robust NLP tools and applications. In *ACL02*, 2002.

[10] ANSI/NISO Z39.96-2012 - JATS: Journal Article Tag Suite. Standard, August 2012.

[11] Ulrik Brandes, Markus Eiglsperger, Jürgen Lerner, and Christian Pich. *Graph markup language (GraphML)*. 2013.

[12] Sebastian Schuster and Christopher D Manning. Enhanced english universal dependencies: An improved representation for natural language understanding tasks. In *LREC*, pages 23–28. Portorož, Slovenia, 2016.

[13] Alexandru Constantin, Silvio Peroni, Steve Pettifer, David Shotton, and Fabio Vitali. The document components ontology (doco). *Semantic web*, 7(2):167–181, 2016.

[14] Daniel R. Schlegel. *Concurrent Inference Graphs*. PhD thesis, University at Buffalo, The State University of New York, 2015.

[15] Daniel R Schlegel and Stuart C Shapiro. Inference graphs: Combining natural deduction and subsumption inference in a concurrent reasoner. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15)*, 2015.

[16] Daniel R. Schlegel and Stuart C. Shapiro. Concurrent reasoning with inference graphs. In Madalina Croitoru, Sebastian Rudolph, Stefan Woltran, and Christophe Gonzales, editors, *Graph Structures for Knowledge Representation and Reasoning, Lecture Notes in Artificial Intelligence*, volume 8323 of *Lecture Notes in Artificial Intelligence*, pages 138–164. Springer International Publishing, Switzerland, 2014.

[17] Stuart C. Shapiro. An introduction to SNePS 3. In Berhard Ganter and Guy W. Mineau, editors, *Conceptual Structures: Logical, Linguistic, and Computational Issues. Lecture Notes in Artificial Intelligence 1867*, pages 510–524. Springer-Verlag, Berlin, 2000.

[18] Daniel R. Schlegel and Stuart C. Shapiro. Visually interacting with a knowledge base using frames, logic, and propositional graphs. In Madalina Croitoru, Sebastian Rudolph, Nic Wilson, John Howse, and Olivier Corby, editors, *Graph Structures for Knowledge Representation and Reasoning, Lecture Notes in Artificial Intelligence 7205*, pages 188–207. Springer-Verlag, Berlin, 2012.

[19] Stuart C. Shapiro and William J. Rapaport. The SNePS family. *Computers & Mathematics with Applications*, 23(2–5):243–275, January–March 1992.

[20] Ralph Grishman. Information extraction: Capabilities and challenges. Notes prepared for the 2011 International Summer School in Language and Speech Technologies, Tarragona, Spain, August 2011.

[21] Jeffrey Steven Gruber. *Studies in lexical relations.* PhD thesis, Massachusetts Institute of Technology, 1965.

[22] Charles J Fillmore. The case for case. 1967.

[23] Martha Palmer, Daniel Gildea, and Nianwen Xue. Semantic role labeling. *Synthesis Lectures on Human Language Technologies*, 3(1):1–103, 2010.

[24] Karin Kipper, Anna Korhonen, Neville Ryant, and Martha Palmer. A large-scale classification of english verbs. *Language Resources and Evaluation*, 42(1):21–40, 2008.

[25] Abdelwaheb Essaihi, George Michel, and Richard N Shiffman. Comprehensive categorization of guideline recommendations: creating an action palette for implementers. In *AMIA Annual Symposium Proceedings*, volume 2003, page 220. American Medical Informatics Association, 2003.

[26] Reinhardt Wenzina and Katharina Kaiser. Identifying condition-action sentences using a heuristic-based information extraction method. In *Process Support and Knowledge Representation in Health Care*, pages 26–38. Springer, 2013.

[27] Katharina Kaiser, Andreas Seyfang, and Silvia Miksch. Identifying treatment activities for modelling computer-interpretable clinical practice guidelines. In *International Workshop on Knowledge Representation for Health Care*, pages 114–125. Springer, 2010.

[28] Radu Serban, Annette ten Teije, Frank van Harmelen, Mar Marcos, and Cristina Polo-Conde. Extraction and use of linguistic patterns for modelling medical guidelines. *Artificial intelligence in medicine*, 39(2):137–149, 2007.

[29] R Serban, A ten Teije, et al. Exploiting thesauri knowledge in medical guideline formalization. *Methods of Information in Medicine*, 48(5):468–474, 2009.

[30] Musarrat Hussain, Jamil Hussain, Muhammad Sadiq, Anees Ul Hassan, and Sungyoung Lee. Recommendation statements identification in clinical practice guidelines using heuristic patterns. In *SNPD 2018*, pages 152–156. IEEE, 2018.

[31] Hossein Hematialam and Wlodek Zadrozny. Identifying condition-action statements in medical guidelines using domain-independent features. *arXiv preprint arXiv:1706.04206*, 2017.

[32] Maria Taboada, Maria Meizoso, D Martínez, David Riaño, and Albert Alonso. Combining open-source natural language processing tools to parse clinical practice guidelines. *Expert Systems*, 30(1):3–11, 2013.

[33] Guergana K Savova, James J Masanz, Philip V Ogren, Jiaping Zheng, Sunghwan Sohn, et al. Mayo clinical text analysis and knowledge extraction system (cTAKES): architecture, component evaluation and applications. *Journal of the American Medical Informatics Association*, 17(5):507–513, 2010.

[34] Ergin Soysal, Jingqi Wang, Min Jiang, Yonghui Wu, Serguei Pakhomov, et al. CLAMP–a toolkit for efficiently building customized clinical natural language processing pipelines. *Journal of the American Medical Informatics Association*, 25(3):331–336, 2017.

[35] Qing T Zeng, Sergey Goryachev, Scott Weiss, Margarita Sordo, Shawn N Murphy, and Ross Lazarus. Extracting principal diagnosis, co-morbidity and smoking status for asthma research: evaluation of a natural language processing system. *BMC medical informatics and decision making*, 6(1):30, 2006.

[36] Daniel R Schlegel, C Crowner, F Lehoullier, and Peter L Elkin. HTP-NLP: A new NLP system for high throughput phenotyping. *Studies in health technology and informatics*, 235:276, 2017.

[37] Samson W Tu, Mor Peleg, Simona Carini, Michael Bobak, Jessica Ross, Daniel Rubin, and Ida Sim. A practical method for transforming free-text eligibility criteria into computable criteria. *Journal of biomedical informatics*, 44(2):239–250, 2011.