General Symptom Extraction from VA Electronic Medical Notes

Guy Divita[a,b], Gang Luo, PhD[c], Le-Thuy T. Tran, PhD[a,b], T. Elizabeth Workman, PhD[a,b], Adi V. Gundlapalli, MD, PhD[a,b], Matthew H. Samore, MD[a,b]

[a] VA Salt Lake City Health Care System, Salt Lake City, Utah, USA
[b] University of Utah School of Medicine, Salt Lake City, Utah, USA
[c] Department of Biomedical Informatics and Medical Education, University of Washington, Seattle, USA

Abstract

There is need for cataloging signs and symptoms, but not all are documented in structured data. The text from clinical records are an additional source of signs and symptoms. We describe a Natural Language Processing (NLP) technique to identify symptoms from text. Using a human-annotated reference corpus from VA electronic medical notes we trained and tested an NLP pipeline to identify and categorize symptoms. The technique includes a model created from an automatic machine learning model selection tool. Tested on a hold-out set, its precision at the mention level was 0.80, recall 0.74 and an overall f-score of 0.80. The tool was scaled-up to process a large corpus of 964,105 patient records.

Keywords:

Natural Language Processing, Machine Learning, Diagnosis

Introduction

There is need for more granular data, in particular, signs and symptoms, from the medical record in clinical and research domains such as quality improvement, population health metrics, patient recruitment for clinical trials, surveillance for adverse events, post-marketing surveillance, genomic medicine (genotype-phenotype associations), and epidemiologic studies. Structured data elements holding signs and symptoms have been shown to be underreported in clinical data repositories[1]. Advances in natural language processing (NLP) have begun to unlock information from the free text of medical notes[2]. This paper reports on an effort to extract signs and symptoms from clinical text using NLP. While prior work focused on specific signs and symptoms or worked within limited domains, this effort is more general.

Symptom Definition

Symptoms are the essence of the patient's experience of illness. A medical encounter starts with what a patient conveys to his or her provider in the form of symptoms and concerns. The traditional distinction between symptoms and signs is that symptoms are subjective experiences whereas signs are objectively observed. A more formal definition of a symptom is that it is "a bodily feature of a patient that is observed by the patient and is hypothesized by the patient to be a realization of a disease [3]. An operational criterion of a symptom includes the elements of patient experience, abnormal characteristics, and clinical relevance.

Symptoms are captured in the medical record as reported by providers. The patient's own words are filtered through the experience of the medical provider. Moreover, not all of the symptoms expressed by the patient are entered into the medical record [4]. Often, there is paraphrasing and summarization. When a symptom is difficult to paraphrase or capture in a single concept, it may be quoted verbatim in the record using quotation marks. Once the medical documentation is complete and the encounter is closed, there are only inferred references to the symptoms. Some, but not all symptoms are reflected in problem lists or ICD-9-CM coding of the medical encounter[5] or inferred on the basis of prescribed medications. Forbush noted that while problem lists included on average three symptoms per document, six symptoms on average are mentioned in the clinical note. Most documentation of symptoms is in the form of free text in clinical notes.

Common to other NLP extraction tasks, this work recognizes symptoms by a dictionary lookup methodology but with a large (over 92k) symptom concept dictionary. Additional notable methods described in this paper include recognizing common lexical patterns indicative of symptom phrases, recognizing only asserted mentions and an optimized machine learned component to filter out fallacious symptom phrases.

The impetus for this work revolves around improving patient care for veterans. US military personnel who have served in combat theaters experience various symptoms and illnesses attributable to their deployment [6, 7]. Of the most common conditions noted in administrative data of recently returned combat veterans are "non-specific signs and symptoms" represented by ICD-9-CM codes 780-799 [7]. While this is an appropriate starting point for epidemiologic studies, there is a need to identify symptoms in free text to address the true extent of post-deployment illnesses among Veterans seen in US Department of Veterans Affairs (VA) medical facilities.

More broadly, tracking and assessing the presence of symptoms is useful for surveillance of syndromes[8], staging of disease, and evaluation of treatment response. Phenotyping, which involves the characterization of a set of clinical features, is incomplete without the inclusion of the patient's subjective experience.

The objective of this project was to develop a natural language processing (NLP) pipeline that reliably identifies and extracts mentions of any positively asserted symptoms from the free text of clinical notes. We also address challenges facing current information extraction techniques such as the vast heterogeneity of expression and boilerplating commonly seen in electronic medical records.

Related Work

Extracting concepts from the free text of medical records (clinical text) has been the holy grail of NLP researchers. The challenges of processing clinical text over biomedical text have resulted in slow progress over the years[9]. Starting with outpatient and emergency department encounters, efforts have been underway to process the free text associated with chief complaint data, problem lists for continuity between visits, family history, chest x-ray reports, pathology reports and discharge summaries[10]. Several studies have focused on the free text of the medical encounter (both outpatient and inpatient) in looking for clues to adverse events or for bio surveillance[11]. Limited studies have focused on signs and symptoms associated with specific diseases or conditions; these include infectious diseases such as pneumonia[12] and influenza[13], and cancer staging[14]. Major impetuses to advance NLP of clinical text have been the serial i2b2 challenges. The 2010 challenge focused on problems, assertions and relationships[15]. Dligach, et al.[16] mentioned symptom extraction as a component of discovering body site and severity within clinical texts via cTAKES, but the signs and symptoms were not the focus of this work. More recently, Roberts[17] describes identifying symptom mentions through semantic categorization, extracting patterns that involve spatial relations between disorders and anatomical structures from well-formed prose. However, a very limited number of studies that have focused on symptoms expressed by patients in the body of the electronic note; there are virtually no studies on looking broadly at symptoms across sets of patients.

Methods

Reference Document Corpus

A sample of 948 records were extracted from a cohort of 6 million patient records from Veterans that had recently returned from deployment in Afghanistan and Iraq. The clinical notes were pulled from the VA's Corporate Data Warehouse (CDW) using the Veterans Informatics and Computing Infrastructure (VINCI) [18]. The records extracted were from 164 pre-selected document types. These records were human annotated to identify 5,819 positively asserted symptoms. Forbush describes the corpus characteristics and the annotation task.[5]. This corpus was divided into a training set and a hold-out testing set.

Natural Language Processing Pipeline

V3NLP Framework described in Divita et al[19] was used to build an NLP pipeline. V3NLP Framework is a framework built upon the Apache UIMA project[20].

This symptom extraction task is accomplished by a symptom dictionary lookup mechanism augmented with a statistical machine-learning filter. A UIMA pipeline was assembled using V3NLP framework components. UIMA pipelines are composed of a series of annotators, where the output of one annotator is in turn the input to the next. The annotators chosen at the front of the symptom pipeline decompose the text into constituent document element parts[21] including sections, content headings, lists, sentences, phrases, lines, tokens, slots and their values, questions and their answers, and check boxes, as well as other boilerplate entities. Additional annotators are included to add relevant features that will enable the downstream machine learning annotators to make an informed decision about whether a potential symptom is a true symptom or not. These annotators include a part of speech tagger and multi-word term identification to identify symptoms and non-symptoms. An annotator was created specifically for this task to identify potential symptoms by rules and patterns formed from annotations created by the dictionary lookup and document decomposition. The ConText assertion (negation, assertion, subject, hypothetical, conditional, historical) annotator [22] was included to add assertion attributes to potential symptoms to filter out negated and hypothetical symptom mentions such as *denies pain*, and *prn dizziness*.

A tail-end annotator was created for this task that employs a machine-learned model trained on 65 features gleaned from the upstream annotators. Figure **1** shows the production pipeline. The subsections that follow here describe the novel annotators within the symptom pipeline.

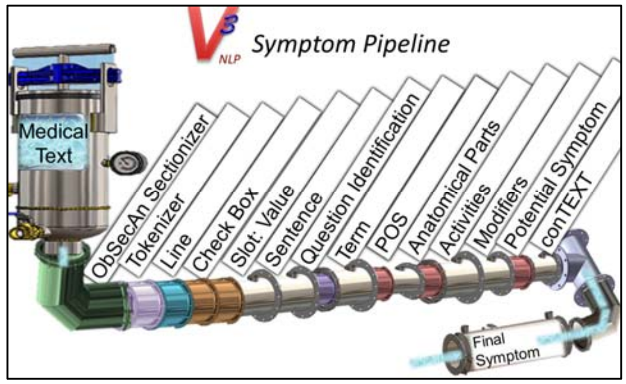

*Figure **1**- V3 NLP Symptom Pipeline Used for Information Extraction from Free Text of VA Electronic Medical Notes*

Document Element Decomposition Annotators

Section Identification

Section identification is accomplished by a wrapper around OBSecAn[23], which is a sectionizer built from the attributes of what makes up sections from a database of 35,000 document templates used within the VA's Veterans Health Information Systems and Technology Architecture (VistA) system.

Term identification Annotator

The term identification annotator creates term annotations from longest matching spans within sentence boundaries. This is the dictionary lookup portion of the pipeline. Term lookup uses the lookup algorithm described in Sophia, an Expedient Concept Extraction Tool[21]. Tokens are looked up from right to left across a sliding window, matching longest matching chains of tokens from an index that is similarly composed of reverse order tokens from terms.

Terms within the dictionaries include one or more categorizations or semantic types. For this task, multiple dictionaries are used, a distinctive v3NLP functionality. General and medical terminology is covered through the use of the SPECIALIST Lexicon. The identification of general and medical terminology is used to absorb multi-word terms such as "pain scale", that would otherwise cause ambiguity and fallacious symptoms if seen as individual words. A dictionary of 92,000 concepts (122,941 symptom forms) was created from Unified Medical Language System (UMLS) sources for this task, described by Tran[24]. Terms within this resource are tagged with a symptom category along with a set of 15 organ system sub-categories. A dictionary of idiosyncratic symptom phrases and symptoms not covered by the symptom dictionary (but seen in training data) is also employed. Terms from this resource are tagged with just the symptom category. A dictionary of symptom exceptions, or pertinent negatives, is included as a convenient way to quickly incorporate exceptions for specific purposes. Such was needed to address failures seen from the training set.

Potential Symptom Annotator

The potential symptom annotator is a rule-based method that identifies those terms in the text marked with a symptom category and creates a potential symptom if these instances are not observed to be in content headings. The method looks for symptom mentions identified in the dictionary lookup within a window of a sentence and no evidence that would indicate the mention is not a symptom before promoting the symptom mention to a potential symptom. There are nearly 123 thousand strings that could make up possible symptom mentions. Thus far, this technique wildly over-generates potential symptoms.

Addressing Boilerplated Content

Clinical records are replete with boilerplated text. Such text is telegraphic, underspecified shorthand used to convey meaning by shortening the lengthy narrative that would otherwise be required. There is a large amount of variety and variability to the boilerplated content within clinical text. Such content includes check boxes, slots and their values, questions and their answers, and pre-written prose text that has been copied-and-pasted into the record. The assertion semantics of symptom mentions found within check-boxes, slots and their values and questions is different for each of these entities. For instance, a symptom contained within the content heading of a check-box is only asserted if the box is checked. A symptom mention found within the content heading of a slot and filler structure is only asserted if the value or filler is filled out and has a non-negative kind of value. The assertion semantics are similar for symptom mentions within questions. Integrations of prior work in this area was extended to this pipeline[21].

Additional Annotators

Previous work in this area has described colorful and often vague descriptions of symptoms [5]. Such descriptions include mentions of an anatomical location. An additional symptom pattern includes a normal activity and modifiers to that normal activity with some negative or pathologic connotation. Symptom patterns also often included some indication of severity and duration. A similar insight was further observed in reviewing false positives: a large portion of them involved an activity with some kind of positive modifier. For instance, mentions of *sleep* prefaced by *improved* were seen as false positives, but not poor sleep. Those observations led to the compilation of a dictionary of modifiers, activities, and anatomical locations from UMLS resources and the creation of annotators for each.

We used multiple methods to build these resources. To form a high-level list of anatomical locations, we extracted UMLS concept unique identifiers (CUIs) from the Consumer Health Vocabulary (CHV) project files[25] and mapped them to the corresponding terms in the 2014 SNOMED CT terminology. These terms were hand-curated to identify the surface type of terms one finds in a patient's symptom description. These were augmented with additional terms from the CHV's last terminology, found in the 2011AA UMLS Metathesaurus release. To locate normal activities that also intersect with findings and functions, we extracted terms from almost every English vocabulary in the 2014AA Metathesaurus UMLS release that had a personal behaviors semantic type (Activity, Behavior, Daily or Recreational Activity, Individual Behavior, or Social Behavior) and then terms for semantic types Finding and Organism Function. The final list consisted of words occurring in both of these groups. For the modifiers list, we extracted every adverbial term from the UMLS Specialist Lexicon LRAGR file (2014AA release). These were augmented with terms extracted from Patients Like Me symptom descriptions[26] and other symptom descriptions on the Internet. Those activities, modifiers, and anatomical locations within the sentence that included a potential symptom were added as features to the machine learning.

Machine Learning Annotator: Training

Initially, the dictionary and rule based mechanisms produced approximately nine false symptom mentions for each true symptom mention. An additional mechanism using the surrounding context was needed to filter down the false positive mentions. An annotator was developed to create Weka ARFF data rows filled with the feature values needed to train Weka machine learning models.[27] This annotator was placed at the tail end of the training pipeline building an ARFF training row for each mention found, noting if the mention also overlapped a human marked symptom. The subsequent ARFF file involving **16,353** training rows (5,819 positive examples, 10,534 negative examples) was used to create a machine learned model based on features and whether a human annotation overlapped the mention.

We used the automatic machine learning model selection tool built by Luo et al.[28, 29] to systematically test every classification algorithm in Weka and tune hyper-parameters. Compared to other similar ones, this tool can greatly reduce search time and classification error rate[29].

A technical note here: all mentions from all 948 notes were used to create the initial ARFF file. The rows from the ARFF file were then randomized, separated into a training set consuming 90% of the examples and a hold-out 10% used for testing.

Machine Learning Features

Features were chosen on the basis of adding evidence to identify a possible symptom as a true symptom. Five words to the left and to the right of the potential symptom and their respective parts of speech, and the part of speech of the potential symptom are included. In earlier versions, these were all grouped into a bag-of-words vector. The current iteration includes a feature for each of the fifth, fourth, third, second and first words to the left and right. Each feature includes enumerated values for what words appear in that position keeping words that appear more than 2 times in that position. Positional features intended to capture boilerplate clues are included, such as if the symptom appears in a checkbox, slot, value, question, list, or sentence, and if it is within a section. Also included are features associated with section information including the section name, if the line that the symptom appears in has been indented, and if the line includes camel case or all upper case. Activity, modifier, and anatomical part features are included based on the afore mentioned insight. Also included is the assertion status of the symptom. An analysis of the attributes that contributed most to the outcome revealed that the symptom words, followed by symptom category, section name, the forth, and fifth words to the left and second and third words to the right were the most salient attributes.

Production Pipeline

A final annotator was developed that mirrored the machine learning annotator in that it creates Weka instances around potential symptoms, which are subsequently passed through the Weka trained model to classify whether they are true or not. Symptom annotation instances are created for those that are classified as true symptoms.

Results

The automatic machine learning model selection tool[28, 29]selected support vector machine coupled with stochastic gradient descent as the classification algorithm. Tested on five iterations of different randomly assigned 90% training, 10% hold-out mentions, this model held a consistent performance for identifying asserted symptoms. Table 1 shows the information retrieval metrics for this model.

The model was folded into the NLP tool, scaled-up and run on a larger set of 964,105 records randomly chosen from the larger OEF/OEF cohort. The process ran 32 concurrent pipelines and took 11 hours to run, at an average speed of 40 ms per record. In all, 59,412 symptom mentions were found from 19,914 documents from 10,397 patients. Figure 2 shows the distribution of organ system classes of the symptoms found in this cohort.

Table 1- SGD Model Performance on Hold-out Set

Class	Precision	Recall	F-Measure
Symptom	0.80	0.74	0.80

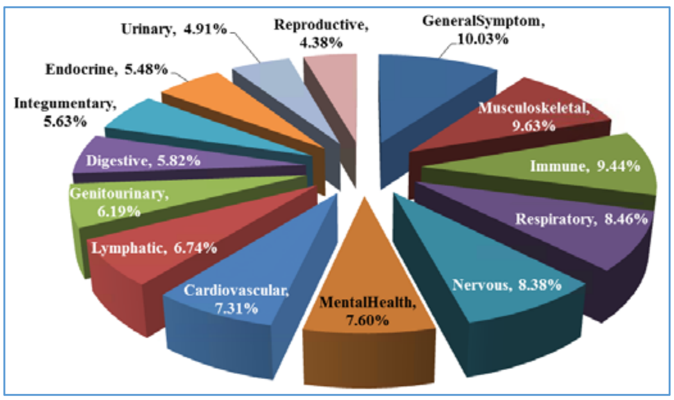

Figure 2- OEF/OIF Patient Symptom Distribution by Organ System

Failure Analysis

Snippets were generated around the false positives, and a sampling of 200 of these were categorized by type of failures. Issues in recognizing the numerous ways a mention can be negated are far more prevalent than any other type of error (57%). There were a number of false positives that could be plausibly true (16%). The rate of these are explainable due to the difficulty of the human annotation task. There were false positives caused by incorrectly parsing templates (5%) and checkboxes (5%). There was a smattering of other issues (21%) that could not easily be classified.

Discussion

Symptom extraction brings with it many challenges. Among them, a similarity and a continuum in context between signs, symptoms, findings, and diagnoses, making the distinction between these via explicit dictionary lookup and rules difficult. The term *depression* is a good example. It occurs 1160 times in this corpus. The depression mentions included references to patient reported symptoms, to provider observations and findings as well as the provider diagnoses. Our attempts at such without a machine learning component were disappointing. An additional requirement was the need to create a curated lexicon that significantly extended the pre-existing resources within UMLS. This additional resource was necessary to remedy the incompleteness of relevant terms within UMLS and to resolve the inconsistent distribution of symptoms across multiple UMLS semantic types. Moreover, our curated resource made it feasible to classify symptoms according to sub-type and organ system.

While the pipeline was developed specifically for VA medical notes, the general principles would be applicable to other large health care systems with commercial EMR's that contain free text and semi-structured notes with templates. The technique is useful for applications extracting patient described indications, useful for adding to the phenotype for conditions. The lessons learned with regards to document element decomposition and identification of slot value pairs would also be portable and generalization to other settings where EMRs are used.

Limitations

We were not able to accurately calculate metrics at the document or patient level due to randomizing the mentions before splitting the training and testing sets.

Despite related work that has greatly expanded the ConText patterns and sped up the application of the algorithm, negation continues to be the greatest source of false positives, at a rate of 60%.

Identifying a potential symptom is challenging for several reasons. The first is the observation that a large set of symptoms within the symptom dictionary are concepts that are a *finding* as well as a *normal* behavior, activity, or function. Such forms were observed to be a large portion of initial false positives. The heterogeneity of document types and the frequency and variety of boilerplated semi-structured elements continues to be troublesome. Despite the use of document element decomposition annotators, review of VA electronic medical notes reveals that symptom mentions are frequently found in telegraphic, boilerplated lists, check boxes and questions.

The practice of using terms denoting symptoms with both an activity and a modifier also poses a challenge for information extraction. No easy mechanism was identified to mark the modifier polarity; such information would be of benefit for future iterations of the pipeline.

The slot value annotator along with the question and answer annotator need more refined techniques to catch idiosyncratic formats, easy to understand visually, but difficult to generalize into patterns and rules. Beyond this, it should be noted that the training set contains inconsistencies with how some boilerplated sections were annotated or not annotated.

Word sense disambiguation, a challenge seen with the many acronyms and abbreviations, was not directly addressed in this pipeline. Co-reference resolution was partially addressed within the ConText algorithm which attributes if the symptom mention is attributed to the patient or other entity. Other than failures due to who to attribute the symptom to, co-reference resolution was not observed to be a point of failure. Neither issue rose to the levels of failure that negation or adequately parsing though check-boxes and questions currently pose.

The split for cross-validation was performed at the mention level. This may lead to some documents having mentions distributed into both the training set and the test set. Since a document may contain several occurrences of the same symptom, this is liable to result in an optimistic evaluation of the classifier results.

Future Work

This is being deployed in several applications where the focus is narrowed to specific conditions. We should learn how well the tool identifies specific kinds of symptoms from these studies. Recently added VINCI tools should allow us to compare our technique with cTAKES and CLAMP surrogates.

Conclusion

We have developed a technique to identify a variety of signs and symptoms within a wide range of document types. An exhaustive algorithm was used to find the most robust machine learning model to train with. The technique has been efficacy benchmarked with an f-metric of 0.80 against a hold-out set of symptom mentions. The technique was scaled-up and run across close to one million records. The pipeline and application is distributed under an Apache License at http://inlp.bmi.utah.edu/redmine/docs/v3nlp-framework

Acknowledgements

This work is funded by the United States Department of Veterans Affairs, OR&D, Health Services Research and Development grants VINCI HIR-08-204, CHIR HIR 08-374, ProWATCH grants HIR-10-001 and HIR 10-002. VA Informatics and Computing Infrastructure (VINCI) resources were used to develop and run this work.

The views expressed in this paper are those of the authors and do not necessarily represent the views of the United States Department of Veterans Affairs or the United States government.

References

1. Pakhomov, S., et al., *Agreement between patient-reported symptoms and their documentation in the medical record.* The American journal of managed care, 2008. **14**(8): p. 530.
2. Nadkarni, P.M., L. Ohno-Machado, and W.W. Chapman, *Natural language processing: an introduction.* J Am Med Inform Assoc, 2011. **18**(5): p. 544-51.
3. Scheuermann, R.H., W. Ceusters, and B. Smith, *Toward an ontological treatment of disease and diagnosis.* Summit on translational bioinformatics, 2009. **2009**: p. 116.
4. Pakhomov, S.V., et al., *Agreement between patient-reported symptoms and their documentation in the medical record.* Am J Manag Care, 2008. **14**(8): p. 530-9.
5. Forbush, T.B., et al., *"Sitting on pins and needles": characterization of symptom descriptions in clinical notes".* AMIA Jt Summits Transl Sci Proc, 2013. **2013**: p. 67-71.
6. Smith, T.C., et al., *Chronic multisymptom illness: a comparison of Iraq and Afghanistan deployers with veterans of the 1991 Gulf War.* Am J Epidemiol, 2014. **180**(12): p. 1176-87.
7. Spelman, J.F., et al., *Post deployment care for returning combat veterans.* J Gen Intern Med, 2012. **27**(9): p. 1200-9.
8. Elkin, P.L., et al., *Comparison of natural language processing biosurveillance methods for identifying influenza from encounter notes.* Annals of Internal Medicine, 2012. **156**(1_Part_1): p. 11-18.
9. Meystre, S.M., et al., *Extracting information from textual documents in the electronic health record: a review of recent research.* Yearb Med Inform, 2008. **35**: p. 128-44.
10. Demner-Fushman, D., W.W. Chapman, and C.J. McDonald, *What can natural language processing do for clinical decision support?* Journal of biomedical informatics, 2009. **42**(5): p. 760-772.
11. Chapman, W.W., et al., *Natural language processing for biosurveillance*, in *Infectious Disease Informatics and Biosurveillance*. 2011, Springer. p. 279-310.
12. Chapman, W.W., et al., *Identifying respiratory findings in emergency department reports for biosurveillance using MetaMap.* Medinfo, 2004. **11**(Pt 1): p. 487-91.
13. South, B.R., et al. *Optimizing A syndromic surveillance text classifier for influenza-like illness: Does document source matter?* in *AMIA Annual Symposium Proceedings*. 2008. American Medical Informatics Association.
14. Nguyen, A.N., et al., *Symbolic rule-based classification of lung cancer stages from free-text pathology reports.* Journal of the American Medical Informatics Association, 2010. **17**(4): p. 440-445.
15. Uzuner, Ö., et al., *2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text.* Journal of the American Medical Informatics Association, 2011. **18**(5): p. 552-556.
16. Dligach, D., et al., *Discovering body site and severity modifiers in clinical texts.* Journal of the American Medical Informatics Association, 2013: p. amiajnl-2013-001766.
17. Roberts, K., et al. *Automatic Extraction and Post-coordination of Spatial Relations in Consumer Language*. in *AMIA Annual Symposium Proceedings*. 2015. American Medical Informatics Association.
18. US Department of Veterans Affairs. *VA Informatics and Computing Infrastructure (VINCI)*. 2015 [cited 2015; Available from: http://www.hsrd.research.va.gov/for_researchers/vinci/.
19. Divita, G., et al., *v3NLP Framework: Tools to Build Applications for Extracting Concepts from Clinical Text.* eGEMs, 2016. **4**(3).
20. Ferrucci, D. and A. Lally, *UIMA: an architectural approach to unstructured information processing in the corporate research environment.* Natural Language Engineering, 2004. **10**(3-4): p. 327-348.
21. Divita, G., et al., *Recognizing Questions and Answers in EMR Templates Using Natural Language Processing.* Stud Health Technol Inform, 2014. **202**: p. 149-52.
22. Chapman, W.W., D. Chu, and J.N. Dowling, *ConText: an algorithm for identifying contextual features from clinical text*, in *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*. 2007, Association for Computational Linguistics: Prague, Czech Republic. p. 81-88.
23. Tran, L.-T.T., et al., *OBSecAnnot: An Automated Section Annotator for Semi-structured Clinical Documents*, in *Medinfo*. 2015.
24. Tran, L.-T.T., et al., *Exploiting the UMLS Metathesaurus for Extracting and Categorizing Concepts Representing Signs and Symptoms to Anatomically Related Organ Systems.* Journal of Biomedical Informatics, 2015
25. Zeng, Q.T., et al., *Extracting principal diagnosis, co-morbidity and smoking status for asthma research: evaluation of a natural language processing system.* BMC Med Inform Decis Mak, 2006. **6**: p. 30.
26. *Patients Like Me.* 2014; Available from: www.patientsLikeMe.com.
27. Frank, E., et al., *Data mining in bioinformatics using Weka.* Bioinformatics, 2004. **20**(15): p. 2479-81.
28. Luo, G., *MLBCD: a machine learning tool for big clinical data.* Health information science and systems, 2015. **3**(1): p. 1.
29. Zeng, X. and G. Luo, *Progressive sampling-based Bayesian optimization for effi-cient and automatic machine learning model selection.*

Correspondence: Guy Divita, MS, guy.divita@hsc.utah.edu or Adi V. Gundlapalli, MD, PhD, Chief Health Informatics Officer and Staff Physician, VA Salt Lake City Health Care System, Salt Lake City, Utah, USA adi.gundlapalli@hsc.utah.edu