11-26-2014

Automating Data Abstraction in a Quality Improvement Platform for Surgical and Interventional Procedures

Meliha Yetisgen
University of Washington - Seattle Campus, melihay@uw.edu

Prescott Klassen
University of Washington, klassp@uw.edu

Peter Tarczy-Hornoch
University of Washington, pth@uw.edu

Recommended Citation

Yetisgen, Meliha; Klassen, Prescott; and Tarczy-Hornoch, Peter (2014) "Automating Data Abstraction in a Quality Improvement Platform for Surgical and Interventional Procedures," *eGEMs (Generating Evidence & Methods to improve patient outcomes)*: Vol. 2: Iss. 1, Article 17.
DOI: https://doi.org/10.13063/2327-9214.1114
Available at: http://repository.edm-forum.org/egems/vol2/iss1/17

The Electronic Data Methods (EDM) Forum is supported by the Agency for Healthcare Research and Quality (AHRQ), Grant 1U18HS022789-01. eGEMs publications do not reflect the official views of AHRQ or the United States Department of Health and Human Services.

Automating Data Abstraction in a Quality Improvement Platform for Surgical and Interventional Procedures

Abstract

Objective: This paper describes a text processing system designed to automate the manual data abstraction process in a quality improvement (QI) program. The Surgical Care and Outcomes Assessment Program (SCOAP) is a clinician-led, statewide performance benchmarking QI platform for surgical and interventional procedures. The data elements abstracted as part of this program cover a wide range of clinical information from patient medical history to details of surgical interventions.

Methods: Statistical and rule-based extractors were developed to automatically abstract data elements. A preprocessing pipeline was created to chunk free-text notes into its sections, sentences, and tokens. The information extracted in this preprocessing step was used by the statistical and rule-based extractors as features.

Findings: Performance results for 25 extractors (14 statistical, 11 rule based) are presented. The average f1-scores for 11 rule-based extractors and 14 statistical extractors are 0.785 (min=0.576,max=0.931,std-dev=0.113) and 0.812 (min=0.571,max=0.993,std-dev=0.135) respectively.

Discussion: Our error analysis revealed that most extraction errors were due either to data imbalance in the data set or the way the gold standard had been created.

Conclusion: As future work, more experiments will be conducted with a more comprehensive data set from multiple institutions contributing to the QI project.

Acknowledgements

This project was supported by Grant Number R01HS020025 from the Agency for Healthcare Research and Quality. The content is solely the responsibility of the authors and does not necessarily represent the official views of the Agency for Healthcare Research and Quality. The Surgical Care and Outcomes Assessment Program (SCOAP) is a Coordinated Quality Improvement Program of the Foundation for Health Care Quality. CERTAIN is a program of the University of Washington, the academic research and development partner of SCOAP. Personnel contributing to this study: Centers for Comparative and Health Systems Effectiveness (CHASE Alliance), University of Washington, Seattle, WA: Daniel Capurro, MD; Allison Devlin, MS; E. Beth Devine, PharmD, MBA, PhD; Prescott Klassen, MS; Kevin Middleton; Michael Tepper, PhD; Peter Tarczy-Hornoch, MD; Erik Van Eaton, MD; N. David Yanez III, PhD; Meliha Yetisgen-Yildiz, PhD, MSc; Megan Zadworny, MHA.

Keywords

Natural language processing, Quality improvement, SCOAP CERTAIN

Disciplines

Health Services Research

Automating Data Abstraction in a Quality Improvement Platform for Surgical and Interventional Procedures

Meliha Yetisgen, PhD; Prescott Klassen, MS; Peter Tarczy-Hornoch, MD[i]

Abstract

Objective: This paper describes a text processing system designed to automate the manual data abstraction process in a quality improvement (QI) program. The Surgical Care and Outcomes Assessment Program (SCOAP) is a clinician-led, statewide performance benchmarking QI platform for surgical and interventional procedures. The data elements abstracted as part of this program cover a wide range of clinical information from patient medical history to details of surgical interventions.

Methods: Statistical and rule-based extractors were developed to automatically abstract data elements. A preprocessing pipeline was created to chunk free-text notes into its sections, sentences, and tokens. The information extracted in this preprocessing step was used by the statistical and rule-based extractors as features.

Findings: Performance results for 25 extractors (14 statistical, 11 rule based) are presented. The average f1-scores for 11 rule-based extractors and 14 statistical extractors are 0.785 (min=0.576,max=0.931,std-dev=0.113) and 0.812 (min=0.571,max=0.993,std-dev=0.135) respectively.

Discussion: Our error analysis revealed that most extraction errors were due either to data imbalance in the data set or the way the gold standard had been created.

Conclusion: As future work, more experiments will be conducted with a more comprehensive data set from multiple institutions contributing to the QI project.

Introduction

The Surgical Care and Outcomes Assessment Program (SCOAP) is a collaboration of multiple hospitals in Washington state with the purpose of improving the quality and comparing the effectiveness of surgical procedures.[1] The research is led by clinicians in 55 hospitals and covers multiple types of surgery. Contributions of the SCOAP program have resulted in reduced surgical complications and in cost savings. One of the barriers to scaling up SCOAP is the lack of automated data collection. Data from each patient is manually abstracted from unstructured free-text clinical notes and structured clinical records and is entered through a web-based form into a database. The data collection forms are complex and can include more than 700 individual data elements. The average time spent on manual abstraction for a given case is between 35–40 minutes depending on the complexity of the form and the details of the case. To address the costly manual abstraction approach, the Agency for Healthcare Research and Quality (AHRQ) funded the Comparative Effectiveness Research and Translation Network (CERTAIN) initiative, a project designed to strengthen SCOAP by adding automated data abstraction and to prove its utility to comparative effectiveness research. To achieve this, SCOAP CERTAIN investigators implemented a clinical data repository (CDR) in which data is automatically retrieved from original source hospitals and stored for later analysis.

The investigators analyzed the data elements in the data collection forms to identify data elements that have the potential to be automatically abstracted from structured data and unstructured clinical records. They specifically targeted data elements in the *general SCOAP form*, which is used to abstract information about general surgical procedures (e.g., appendectomies, colectomies, and bariatric surgeries) and includes multiple data elements that range from simple demographic information such as age, gender, and insurance, to more complex data such as whether there was an unplanned intensive care unit (ICU) stay in the postoperative period. Figure 1 presents one section of the form that collects information about *indication of operation*.

A subset of the form is common to all types of procedures (core data elements) and includes information such as the procedure date and surgeon information. The rest of the form is specific to the details of each type of procedure, such as the presence or absence of a colostomy in a procedure involving the colon. In a prior study, investigators analyzed each data element and identified 64.1 percent of the data elements as being manually abstracted from free-text clinical notes.[2] We built a text-processing pipeline based on natural language processing (NLP) and machine learning to automatically

[i]University of Washington

abstract a subset of data elements from clinical notes. The main focus of this paper is to describe the text processing pipeline and its performance in the automatic abstraction task for the 25 data elements selected from the four main sections of the general SCOAP form.

Background and Related Work

Most information regarding patient state, diagnostic procedure, and disease progress is described in the narrative sections and semistructured lists or free-text fields of patient clinical records. These free-text parts of clinical notes provide an opportunity for NLP technologies to play a major role in clinical care research and to facilitate the analysis of information that otherwise has only been accessible through manual chart abstraction. NLP methods have been used in a variety of health care applications including development of decision support tools, quality improvement, and automated encoding for clinical research.[3-4]

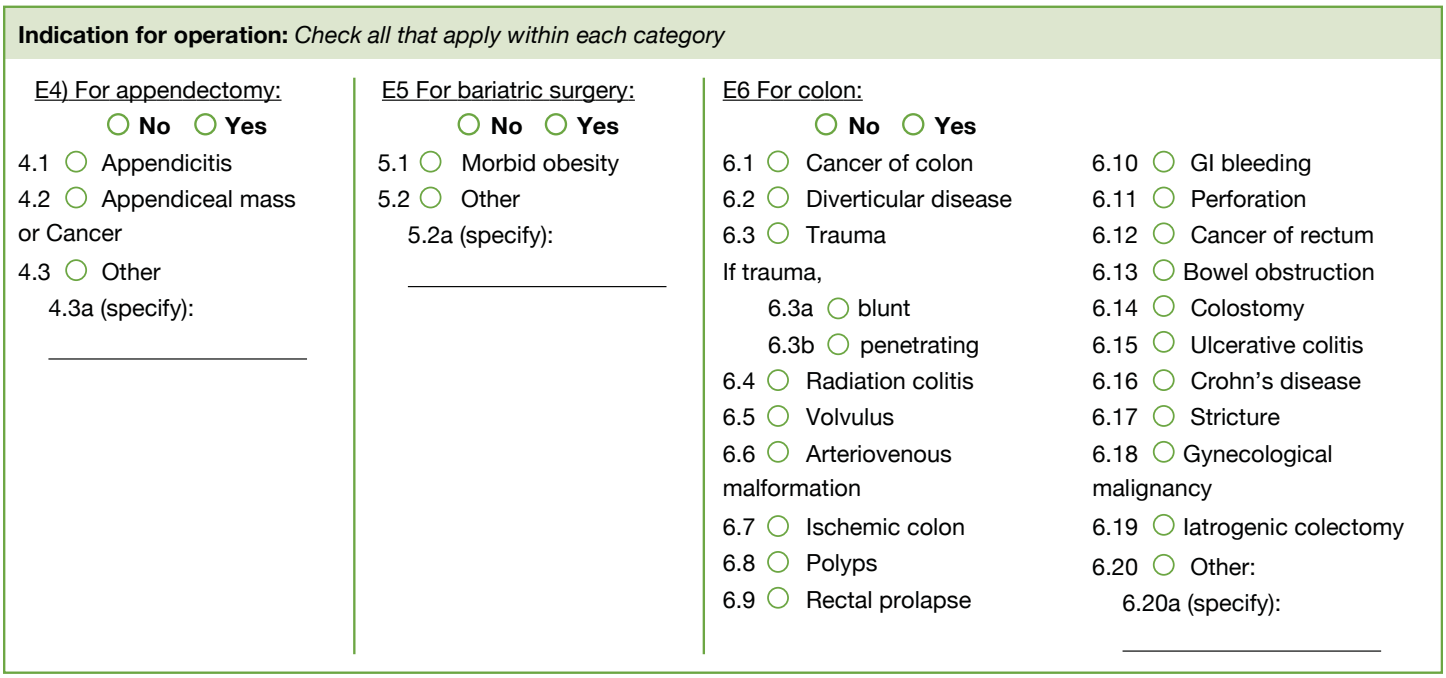
There are several general-purpose NLP systems that perform the task of taking free-text clinical records and transforming them into a set of structured data elements. One such system is MedLEE.[5] Its preprocessing stage includes sentence-breaking, tokenization (to break text into words), and POS tagging modules. In its main processing module, sentences are parsed using cascades of hand-coded rules and are ultimately transformed into semantic frames relevant to the extraction task at hand. In a very recent study, MedLEE has been applied to identify comorbidities from admission notes.[6]

The HiTEX system[7-8] provides a set of extensible modules that can be combined into text processing pipelines and includes a section splitter, section filter, tokenizer, Part-of-Speech (POS) tagger, noun phrase finder, Unified Medical Language System (UMLS) concept finder, negation finder, regular expression-based concept finder, and sentence splitter.

Text analytics systems released by Open Health Natural Language Processing Consortium (OHNLPC) (http://www.ohnlp.org/index.php/Main_Page) are designed to identify clinically relevant named entities in clinical notes, which can then be used in information retrieval and data mining tasks. Two Apache UIMA-based (http://uima.apache.org) text analytics systems have been used to build OHNLPC compliant NLP pipelines: medKAT, which extracts cancer characteristics from pathology reports, and cTAKES[9] which is more general purpose and has been applied to varied tasks including the identification of disorders, drugs, anatomical sites, and procedures in clinical notes.

Our approach is designed to be a highly customizable lightweight NLP pipeline that enables the rapid prototyping of text classification tasks based on a simple set of XML-based templates and the integration of existing standalone NLP tools (openNLP, libSVM, and Mallet). We decided to build our own lightweight pipeline after a review of available generalized clinical systems, which for various reasons did not meet the specific requirements of our project. Some systems (MedLEE) are proprietary and others (cTAKES, HiTEX) rely on third party frameworks like Apache UIMA and GATE, which we felt required too much time and effort to extend and customize to accommodate our particular data and system requirements. We considered another freely available standalone system, MediClass[10] but it did not include modules related directly to our task. After careful consideration, we decided to build our own simple, lightweight NLP pipeline, integrating existing standalone NLP and machine learning tools.

Figure 1. Indication for Operation Section of SCOAP General Data Form



Indication for operation: *Check all that apply within each category*

E4) For appendectomy:
○ No ○ Yes
4.1 ○ Appendicitis
4.2 ○ Appendiceal mass or Cancer
4.3 ○ Other
4.3a (specify): _____

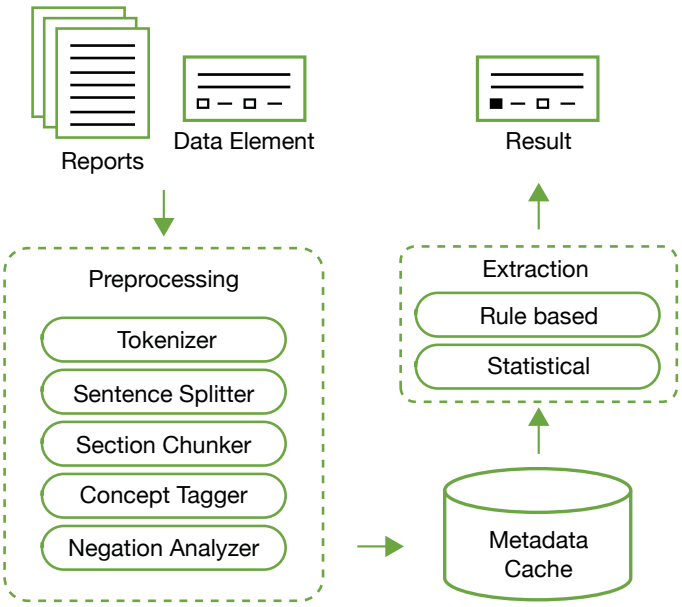
E5 For bariatric surgery:
○ No ○ Yes
5.1 ○ Morbid obesity
5.2 ○ Other
5.2a (specify): _____

E6 For colon:
○ No ○ Yes
6.1 ○ Cancer of colon
6.2 ○ Diverticular disease
6.3 ○ Trauma
If trauma,
6.3a ○ blunt
6.3b ○ penetrating
6.4 ○ Radiation colitis
6.5 ○ Volvulus
6.6 ○ Arteriovenous malformation
6.7 ○ Ischemic colon
6.8 ○ Polyps
6.9 ○ Rectal prolapse
6.10 ○ GI bleeding
6.11 ○ Perforation
6.12 ○ Cancer of rectum
6.13 ○ Bowel obstruction
6.14 ○ Colostomy
6.15 ○ Ulcerative colitis
6.16 ○ Crohn's disease
6.17 ○ Stricture
6.18 ○ Gynecological malignancy
6.19 ○ Iatrogenic colectomy
6.20 ○ Other:
6.20a (specify): _____

Methods

The main components of our text-processing pipeline are depicted in Figure 2. The data set and components of the pipeline are described in the following sections.

Figure 2. SCOAP Text Processing Pipeline



Data Set

In order to develop and evaluate our text-processing pipeline, we created a data set consisting of free-text reports pulled from 618 general surgical encounters performed at University of Washington Medical Center and Harborview Medical Center in 2010. The retrospective review of reports was approved by the University of Washington Human Subjects Committee of Institutional Review Board. A total of 20,760 reports (averaging just over 33 reports per encounter) were selected for inclusion in the data set and represented a wide range of report types (e.g., admit note, operative note, nursing notes, radiology reports, discharge summary) created between admission and discharge times of the patient. The reports capture the details of the surgery and overall state of the patient.

Annotators were trained in manual review of medical records for SCOAP data collection. They filled out the general form by abstracting data elements from 618 encounters contained in the clinical reports. We used the abstractions over patient encounters as a gold standard to train and test our pipeline. Because the annotation task was completed by the SCOAP team before we initiated the NLP project, each encounter was annotated by only a single annotator and interrater agreement could not be calculated. We call this data set 2010 SCOAP general data set.

Text Processing Pipeline

The input to our pipeline is a request consisting of two components: a data element and its unique identifier from the general SCOAP form and a set of free-text records documenting a patient encounter (Figure 2). Each note in the patient record encounter is sent to a preprocessing pipeline and the resulting metadata is cached. All records and metadata are then routed to the information extraction component and associated with the requested data element. If reports have already been preprocessed in a previous session, the preprocessing step is skipped, and the cached metadata for the reports is retrieved from the cache and associated with the requested data element. The output is the result of rule-based or statistical classification processed by the extraction component. In the following sections, we describe the main components of our pipeline.

Preprocessing

The preprocessing component is a customizable NLP pipeline that takes a medical record as input and produces for output a version of the medical record that has been annotated with metadata created in each step of the pipeline. The steps of the pipeline are implemented in five components including the following: (1) a tokenizer, (2) a sentence breaker, (3) a section chunker, (4) a concept tagger, and (5) a negation analyzer. The components are applied to each record sequentially. For example, the sentence breaker used by the pipeline requires that the input already be tokenized. Therefore, it requires that the tokenizer must be run before the sentence breaker.

We use the OpenNLP English tokenizer and sentence breaker (http://opennlp.apache.org) to tokenize and split text into sentences. We then apply a statistical approach to section chunking based on our previous work identifying the boundaries and types of the sections in clinical records.[11] Our approach requires (1) the construction of an ontology of section headers for a selected clinical report type (e.g., discharge summary), and (2) the annotation of a corpus of notes based on the constructed ontology. The annotated notes serve as a training corpus for our statistical section chunker.

In our approach to section segmentation each line in a document is classified based on its probability of inclusion in a section type. We incorporated two separate models and steps for section segmentation and classification. First, for section segmentation, section boundaries are identified by labeling each line with a "B" (beginning of section), an "I" (inside of section), or an "O" (outside of section) tag. The same text used in the first step is then passed unlabeled to the second step for section classification, where a separate classifier is called upon to label each section with the appropriate section category. We achieved 0.921 f1-score in chunking radiology reports and 0.968 f1-score in chunking discharge summaries based on a gold standard composed of manually annotated (100) radiology reports and (191) discharge summaries with respect to both the section segmentation and section classification tasks.

After chunking reports into sections, we use MetaMap[12-13] to identify the UMLS concepts and NegEx[14] to extract the negation state of the identified concepts.

Information Extraction

We developed both rule-based and statistical information-extraction approaches to extract 25 data elements from the 2010 SCOAP data collection form and general data set. A total of 25 extractors were created: 11 rule based and 14 statistical. The 25 data elements were selected based on the following two criteria: the interest value for the whole project team, and the size and distribution of the data set (not all data elements have coverage across all encounters in the data set).

Rule-Based Extraction

Rule-based extraction uses one or more text-based triggers to classify patient records with an appropriate label. There are two basic types of triggers: medical concepts; and keywords and regular expressions. Medical concepts are sourced from the UMLS ontology and each report is processed by MetaMap to identify the medical concepts. The manually selected UMLS concepts are compared against those extracted by MetaMap. Keywords and regular expressions are manually formulated to cover the cases where UMLS concepts are not sufficient. These two types of triggers can be used in combination to define an overall classification rule. The classifier accepts one of the following three types of decision-making meta rules to apply the text-based triggers: (1) an instance voting threshold, (2) an instance voting maximum, or (3) an aggregate classification. Figure 3 includes an example rule we wrote to capture hypertension.

Figure 3. Rule Set for Hypertension

```
<Decision type="aggregateClassify" defaultClassLabel="No"
predictEmptyInstances="false"/>
<Classifiers>
<Classifier type="ruleBased">
<Triggers>
<Trigger type="concept" value="C0020538"/>
<Trigger type="regex" value="[Hh]ypertension" polarity="positive"/>
<Trigger type="regex" value="HTN " polarity="positive"/>
<Trigger type="regex" value="[Ff]urosemide" polarity="positive"/>
<Trigger type="regex" value="[Hh]ydrochlorothiazide" polarity="positive"/>
<Trigger type="regex" value="[Cc]hlorothiazide" polarity="positive"/>
<Trigger type="regex" value="[Ss]pironolactone" polarity="positive"/>
<Trigger type="regex" value="[Mm]etoprolol" polarity="positive"/>
<Trigger type="regex" value="[Aa]tenolol" polarity="positive"/>
<Trigger type="regex" value="[Cc]arvedilol" polarity="positive"/>
<Trigger type="regex" value="[Ll]isinopril" polarity="positive"/>
<Trigger type="regex" value="[Bb]enazepril" polarity="positive"/>
<Trigger type="regex" value="[Rr]amapril" polarity="positive"/>
<Trigger type="regex" value="[Dd]iltiazem" polarity="positive"/>
<Trigger type="regex" value="[Aa]mlopidine" polarity="positive"/>
</Triggers>
```

The decisions to use specific keywords and regular expressions or UMLS concepts are made based on a combination of expert knowledge and tests of the rules' effectiveness when used against a set of development data. If a UMLS concept can be found for a data element, its effectiveness as a trigger is evaluated against the set of development data to determine if the addition of regular expressions or keywords can improve performance.

A polarity flag determines if the rule-based trigger should be considered neutral, positive, or negative. By default, all triggers have neutral polarity. Negation is addressed by associating a regular expression or keyword trigger with a negative polarity flag. Rule-based triggers are not constrained to only binary classification tasks. A trigger can be explicitly associated with one of many labels to address multiclass classification tasks.

Statistical Extraction

We designed our statistical extraction approach as a text classification task of clinical text (e.g., operative note) into categories of the data element (e.g., Figure 1: *data element*—indication for operation; *categories*—appendectomy, bariatric surgery, colon). Table 1 includes the list of basic features used to represent the content of the medical text. Some of the data elements are described throughout the reports (e.g., full-text operative reports for operation of indication). On the other hand, some elements are described in only certain sections of the reports (e.g., smoking history). To address the second kind of data elements, in our preliminary experiments, we found that using only sentences in which the data elements were mentioned improved the classification performance. As an example, for smoking history, we process only discharge summaries, preanesthesia reports, and pain management reports. In this limited collection of reports, we first identify sentences (or 15 word windows) with mentions of a manually selected list of smoking-related concepts (e.g., packs), and represent the content with the combined mention-window text as a feature vector, consisting of unigram words.[15]

We use the Maximum Entropy[16] implementation available in the Mallet toolkit (http://mallet.cs.umass.edu) as our algorithm for our all classification tasks.

Table 1. Features Used in Statistical Extraction

Feature Type	Features
Text	*uni-, bi-, trigrams*
Knowledge-based	*UMLS concepts*
Semantic	*Negation*
Structural	*Section type*

Findings

We used *5-fold cross validation* on the annotated data set for performance evaluation. In 5-fold cross validation, the original data set is randomly partitioned into five equal-size subsamples—where four subsamples are used as the training data to train a model and the remaining subsample is retained as the validation data to test the model. This cross validation process is repeated five times.

Table 2 shows the 5-fold cross validation results achieved by our system on the 2010 SCOAP general data set for the 25 data elements. The results for each data element are reported in terms of precision, recall, f1-score (harmonic mean of precision and recall), and accuracy. We report performance values for each

Table 2. Performance Evaluation

Data Element Type	#	Data Element	Extraction Approach	Categories	TP	FP	FN	TN	Pre	Rec	F1	Acc
Comorbidities	1	Hypertension	Rule based	Yes	200	26	20	216	0.885	0.909	0.897	0.900
				No	216	20	26	200	0.915	0.893	0.904	
	2	Diabetes	Statistical (Mention)	Yes	114	13	14	321	0.898	0.891	0.894	0.942
				No	321	14	13	114	0.958	0.961	0.960	
	3	Asthma	Rule based	Yes	55	13	23	371	0.809	0.705	0.753	0.922
				No	371	23	13	55	0.942	0.966	0.954	
	4	Sleep apnea	Statistical (Mention)	Yes	122	11	15	314	0.917	0.891	0.904	0.944
				No	314	15	11	122	0.954	0.966	0.960	
	5	CAD	Statistical (Mention)	Yes	16	5	20	421	0.762	0.444	0.561	0.946
				No	421	20	5	16	0.955	0.988	0.971	
	6	HIV	Rule based	Yes	2	19	0	441	0.095	1.000	0.174	0.959
				No	441	0	19	2	1.000	0.959	0.979	
Risk factors	7	Cigarette smoker	Statistical (Mention)	Yes	30	17	48	398	0.638	0.385	0.480	0.868
				No	398	48	17	30	0.892	0.959	0.925	
Indication of operation	8	Op-indication	Statistical	colon	236	3	1	320	0.987	0.996	0.992	0.993
				appendectomy	104	0	2	452	1.000	0.981	0.990	0.996
				bariatric surgery	216	1	1	340	0.995	0.995	0.995	0.996
	9	Cancer of colon	Rule based	Yes	15	7	11	204	0.682	0.577	0.625	0.924
				No	204	11	7	15	0.949	0.967	0.958	
	10	Diverticulitis	Rule based	Yes	25	13	0	199	0.658	1.000	0.794	0.945
				No	199	0	13	25	1.000	0.939	0.968	
	11	Polyps	Rule based	Yes	7	6	5	219	0.538	0.583	0.560	0.954
				No	219	5	6	7	0.978	0.973	0.976	
	12	Rectal prolapse	Rule based	Yes	4	5	0	228	0.444	1.000	0.615	0.979
				No	228	0	5	4	1.000	0.979	0.989	
	13	Bowel obstruction	Statistical	Yes	7	5	9	216	0.583	0.438	0.500	0.941
				No	216	9	5	7	0.960	0.977	0.969	
	14	Colostomy	Statistical	Yes	4	2	10	221	0.667	0.286	0.400	0.949
				No	211	10	2	4	0.957	0.991	0.974	
	15	Ulcerative colitis	Statistical	Yes	22	0	1	214	1.000	0.957	0.978	0.996
				No	214	1	0	22	0.995	1.000	0.998	
	16	Crohn's disease	Rule based	Yes	30	8	0	199	0.789	1.000	0.882	0.966
				No	199	0	8	30	1.000	0.961	0.980	
	17	Stricture	Rule based	Yes	18	27	0	192	0.400	1.000	0.571	0.886
				No	192	0	27	18	1.000	0.877	0.934	
Operation type (colon)	18	Right hemicolectomy	Statistical	Yes	51	0	6	180	1.000	0.895	0.944	0.975
				No	180	6	0	51	0.968	1.000	0.984	
	19	Left hemicolectomy	Statistical	Yes	17	0	7	213	1.000	0.708	0.829	0.970
				No	213	7	0	17	0.968	1.000	0.984	
	20	Low anterior resection	Statistical	Yes	57	19	29	132	0.750	0.663	0.704	0.797
				No	132	29	19	57	0.820	0.874	0.846	
	21	Abdominal perineal resection	Statistical	Yes	5	2	11	219	0.714	0.313	0.435	0.945
				No	219	11	2	5	0.952	0.991	0.971	
	22	Total abdominal colectomy	Rule based	Yes	23	4	19	191	0.852	0.548	0.667	0.903
				No	191	19	4	23	0.910	0.979	0.943	
	23	Stoma takedown	Statistical (Mention)	Yes	2	0	18	217	1.000	0.100	0.182	0.924
				No	217	18	0	2	0.923	1.000	0.960	
	24	Perineal proctectomy	Rule based	Yes	5	29	6	197	0.147	0.455	0.222	0.852
				No	197	6	29	5	0.970	0.872	0.918	
	25	Abdominal protectomy	Statistical (Mention)	Yes	17	7	24	189	0.708	0.415	0.523	0.869
				No	189	24	7	17	0.887	0.964	0.924	

Notes: TP: True positives, FP: False positives, FN: False negatives, FP: False positives, Pre: Precision, Rec: Recall, F1: F1-Score, Acc: Accuracy.

data element category (e.g., "yes," "no") since the importance of each category is equivalent for this study. The average f1-scores for 11 rule-based extractors and 14 statistical extractors are 0.785 (min=0.576,max=0.931,std-dev=0.113) and 0.812 (min=0.571,-max=0.993,std-dev=0.135) respectively. The performance of extractors across data elements varies. We analyzed the performance of extractors presented in Table 2 for each data element type in the following sections.

Comorbidities

In Table 2, comorbidities are automatically extracted from discharge summaries available in the data set. There are three rule-based and three statistical extractors trained to extract the comorbidities. Among the rule-based extractors, we achieve the best average f1-score for *hypertension* (Avg=0.900 where Yes: f1-score=0.897, No: f1-score=0.904) and the worst average f1-score for *HIV* (Avg=0.576 where Yes: f1-score=0.174, No: f1-score=0.979). The reason for the performance difference is that the data set is too skewed for HIV (Yes: 2, No: 460). Although we achieve perfect recall with the defined rules for the "Yes" *HIV* category, the precision is low (0.095) due to 19 false positive cases.

For the statistical extractors, the average f1-scores for *diabetes* (Avg=0.925 where Yes: f1-score=0.894, No: f1-score=0.960) and *sleep apnea* (Avg=0.932 where Yes: f1-score=0.904, No: f1-score=0.960) are higher than those of *coronary artery disease* (CAD) (Avg=0.766 where Yes: f1-score=0.561, No: f1-score=0.971).

Risk Factors

We built a statistical extractor for smoking history, in Table 2. We processed discharge summaries, preanesthesia reports, and pain management reports in our extraction. The average f1-score is 0.868 (Yes: f1-score=0.480, No: f1-score=0.925).

Indication of Operation

Data elements from the "indication of operation" section of Table 2 are extracted from operative notes. The SCOAP general form data element for indication of operation covers three main surgery types including (1) colon, (2) appendectomy, and (3) bariatric surgery. We trained a three-way classifier to identify the surgery type for this data element. Operative reports provide an in-depth description of the procedure and, as a result, our classifier identifies the surgery type very accurately with f1-scores 0.992 for colon, 0.990 for appendectomy, 0.995 for bariatric surgery. The other nine data elements under this section are related to the details of the surgery performed (e.g., cancer of colon for colon surgery). We built rule-based extractors for six of the nine data elements (e.g., cancer of colon) and statistical extractors (e.g., bowel obstruction) for the remaining three data elements. Although, the performance is high for op-indication, for more detailed data elements, the performance is lower, especially for the cases where the data set was imbalanced (e.g., cancer of colon, bowel obstruction, colostomy). The average f1-scores for the rule-based and statistical classifiers are 0.821 and 0.803 respectively.

Operation Type (Colon)

Table 2's "Operation type (colon)" section includes eight data elements that described the specific colon operation performed and were extracted from operative notes. We built two rule-based and six statistical extractors. The overall performance of the statistical classifiers is good in general with the exception of the classifier created for stoma takedown (Avg=0.571 where Yes: f1-score=0.182, No: f1-score=0.960). Out of the two rule-based extractors, the performance of the extractor for perineal proctectomy (Avg=0.570 where Yes: f1-score=0.222, No: f1-score=0.918) is lower than that of the extractor for abdominal perineal resection (Avg=0.703 where Yes: f1-score=0.435, No: f1-score=0.971).

Discussion

Our data set has limitations. First, the data set covers 618 patients from one institution. For some of the data elements (e.g., abdominal perineal resection), the data set is too imbalanced, which causes low performance for the underrepresented classes (Yes: 16, No: 221). Second, the annotation of the data set was completed before the initiation of the NLP project. The annotators had access to the complete patient charts while they manually abstracted the forms. While designing our system, we created—for each of the 25 data elements—a list of report types the annotators typically used for extraction. However, our error analysis revealed that many of the false negatives for a given data element were due to report types not processed by the NLP system for the given data element. In addition, some reports types are stored in the EMR as scans of handwritten documents (e.g., history and physical reports). We decided not to include such reports although we knew information about certain data elements (e.g., comorbidities) were extracted from those reports during manual abstraction. The performance results presented in Table 2 are lower bounds for the real system performance.

Conclusion

In this paper, we describe a text-processing pipeline based on statistical and rule-based approaches. We report performance results for 25 data elements collected for a surgical quality improvement program. We trained and tested our approaches on a limited data set composed of cases whose surgeries were performed in our institution during 2010.

The overall performance of both statistical and rule-based extractors is encouraging. Our error analysis of the low performing extractors revealed that the size of the data set was not enough to capture the characteristics of some of the data elements. In addition, we achieved higher statistical extraction performance when the data sets were more balanced. The number of cases manually abstracted since we created the data set has increased dramatically. As future work, we plan to run experiments on a larger data set covering a longer period. We believe a larger data set will improve overall performance of the extractors. We also plan to extend our data set by including cases from other institutions contributing to SCOAP. This more comprehensive data set will enable us to run domain adaptability experiments to test the generalizability of our approaches.

We released some components of the described text-processing pipeline including section chunker to the NLP community as open source tools. These tools are downloadable at our research website (http://depts.washington.edu/bionlp).

Acknowledgements

This project was supported by Grant Number R01HS020025 from the Agency for Healthcare Research and Quality. The content is solely the responsibility of the authors and does not necessarily represent the official views of the Agency for Healthcare Research and Quality. The Surgical Care and Outcomes Assessment Program (SCOAP) is a Coordinated Quality Improvement Program of the Foundation for Health Care Quality. CERTAIN is a program of the University of Washington, the academic research and development partner of SCOAP. Personnel contributing to this study: Centers for Comparative and Health Systems Effectiveness (CHASE Alliance), University of Washington, Seattle, WA: Daniel Capurro, MD; Allison Devlin, MS; E. Beth Devine, PharmD, MBA, PhD; Prescott Klassen, MS; Kevin Middleton; Michael Tepper, PhD; Peter Tarczy-Hornoch, MD; Erik Van Eaton, MD; N. David Yanez III, PhD; Meliha Yetisgen-Yildiz, PhD, MSc; Megan Zadworny, MHA.

References

1. SCOAP: Surgical Clinical Outcomes Assessment Program [Internet]. SCOAP. Available from: http://www.scoap.org/
2. Devine EB, Capurro D, van Eaton E, Alfonso-Cristancho R, Devlin A, Yanez ND, Yetisgen-Yildiz M, Flum DR, Tarczy-Hornoch P, and Collaborative, CERTAIN (2013) "Preparing Electronic Clinical Data for Quality Improvement and Comparative Effectiveness Research: The SCOAP CERTAIN Automation and Validation Project," eGEMs (Generating Evidence & Methods to improve patient outcomes): Vol. 1: Iss. 1, Article 16.
3. Demner-Fushman D, Chapman WW, McDonald CJ. What can natural language processing do for clinical decision support? J Biomed Inform. 2009 Oct; 42(5):760-72.
4. Chapman WW, Cohen KB. Current issues in biomedical text mining and natural language processing. J Biomed Inform. 2009 Oct;42(5):757-9.
5. Hripcsak G, Friedman C, Alderson PO, et al. Unlocking clinical data from narrative reports: a study of natural language processing. Ann Intern Med 1995;122:681–8.
6. Salmasian H, Freedberg DE, Friedman C. Deriving comorbidities from medical records using natural language processing. J Am Med Inform Assoc. 2013.
7. Zeng QT, Goryachev S, Weiss S, Sordo M, Murphy SN, Lazarus R. Extracting principal diagnosis, co-morbidity and smoking status for asthma research: evaluation of a natural language processing system. BMC Med Inform Decis Mak. 2006 Jul 26;6:30.
8. Goryachev S, Sordo M, Zeng QT. A Suite of Natural Language Processing Tools Developed for the I2B2 Project. AMIA Annu Symp Proc. 2006;931
9. Savova GK, Masanz JJ, Orgen PV, Zheng J, Sohn S, Kipper-Schuler KC, Chute CG. Mayo Clinical Text Analysis and Knowledge Extraction System (cTAKES): Architecture, component evaluation and applications. J Am Med Inform Assoc, 2010. 17(5):p.507-513.
10. Hazlehurst B, Frost HR, Sittig DF, Stevens VJ. MediClass: A system for detecting and classifying encounter-based clinical events in any electronic medical record. J Am Med Inform Assoc. 2005 Sep-Oct;12(5):517-29.
11. Tepper M, Capurro D, Xia F, Vanderwende L, Yetisgen-Yildiz M. Statistical Section Segmentation in Free-Text Clinical Records. Proceeding of the International Conference on Language Resources and Evaluation (LREC), Istanbul, May, 2012.
12. Aronson, AR. Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. In Proceedings of AMIA Annu Symp, 2001. p.17-21.
13. Aronson AR, Lang F-M. An overview of MetaMap: historical perspective and recent advances. J Am Med Inform Assoc, 2010. 17:p.229-236.
14. Chapman WW, Bridewell W, Hanbury P, Cooper GF, Buchanan BG. A simple algorithm for identifying negated findings and diseases in discharge summaries. J Biomed Inform. 2001 Oct;34(5):301-10.
15. Tepper M, Xia F, and Yetisgen-Yildiz M. Smoking Status Detection Across Domains. In Proceedings of the American Medical Informatics Association Fall Symposium (AMIA'12). Chicago, IL, November, 2012.
16. Berger AL, Pietra SAD, Pietra VJD. A maximum entropy approach to natural language processing. Journal of Computational Linguistics. 1996; 22(1):39-71.