Porting a Natural Language Processing Algorithm to Extract Findings from Colonoscopy Pathology Reports

Jennifer A. Pacheco[1]; Andrew J. Gawron, MD, PhD[2]; Kenneth Borthwick[3]; Peggy L. Peissig, PhD[4]; David S Carrell, PhD[5]; Luke V. Rasmussen[1]; Huan Mo, MD[6]; William K. Thompson, PhD[1]

[1]Northwestern University, Chicago, IL; [2]University of Utah, Salt Lake City, UT; [3]Geisinger Health System, Danville, PA; [4]Marshfield Clinic, Marshfield, WI; [4]Group Health Research Institute, Seattle, WA; [6]Vanderbilt University, Nashville, TN

Introduction

The NIH-funded Electronic Medical Records and Genomics Network (eMERGE) has led the development of over 40 electronic health record (EHR) driven phenotype algorithms (`https://emerge.mc.vanderbilt.edu`), many of which are publicly available for use by other institutions via the PheKB website (`https://phekb.org`). However, most of these algorithms are specified as descriptive text files and flowcharts, which in order to be implemented locally must first be translated into code by informatics professionals. This is an especially difficult task when the algorithm depends on natural language processing (NLP) of clinical notes. Several sites within eMERGE have been exploring tools to help overcome these challenges. We describe here one such effort to develop and port an NLP algorithm for selecting cohorts of patients who have undergone colonoscopies resulting in the detection of one or more colon polyps. We relied on a suite of three software tools to develop and share the algorithm: (1) a rapidly configurable implementation of an NLP pipeline for processing notes, (2) a document abstractor tool to aid creation of gold standard corpora for validation, and (3) executable workflows to simplify and standardize execution of the NLP pipeline and evaluation of results.

Methods

To satisfy our research goals with respect to this cohort, we required information both on the histological type of a colon polyp (adenoma, serrated, hyperplastic), and its anatomic location in the colon (proximal or distal). This information is not typically stored in EHRs as structured data, but must be extracted from pathology notes using NLP. The NLP algorithm was implemented as a pipeline of Java components assembled using the Unstructured Information Management Architecture (UIMA) (`https://uima.apache.org/`). The pipeline components (Figure 1A) consist of: (a) a document sectionizer, (b) a sentence detector, (c) a concept detector mapping relevant words and phrases to corresponding SNOMED-CT codes, (d) a negation detector to deal with negated findings, and (e) a relation extractor to appropriately combine histology and location concepts. Relation extraction relied on recurring regular patterns of histology and location mentions in the pathology notes, which allowed for a straightforward rule-based approach. Configuration of these components can be easily accomplished through modification of script files containing regular expressions and core algorithm logic.

A lightweight abstractor tool (`https://github.com/lrasmus/DocumentAbstraction`) with a simple graphical user interface was created to assist the development of gold standard corpora for algorithm validation (Figure 1B). This tool allows an annotator to rapidly cycle through a corpus of notes, creating document level annotations (polyp histologies and locations) using selections from drop-down lists. Corpus document annotations are exported to a plain text comma-separated value (CSV) file.

Finally, we wrapped the NLP library inside an executable workflow based on the open-source and extensible Java-based KNIME data analytics platform (`https://www.knime.org/`). This platform enables the graphical creation and execution of data workflows that can read, transform, visualize, and write data in various formats. It can also integrate code written in other languages, such as Java and R. KNIME workflows can be exported as zip files and shared with other users. For our colon polyps algorithm, we developed two exportable KNIME workflows. The first workflow (Figure 1C) handles execution of the NLP pipeline on data from a local database or file, and exports a CSV file containing structured data generated by the NLP. The second workflow (Figure 1D) validates the algorithm by reading in the CSV file exported by the NLP workflow, reading in the CSV file exported by the annotator tool, and comparing the results. The validation workflow then automatically calculates and displays sensitivity, positive predictive value (PPV), and F-measure.

Results

The algorithm was initially developed and validated at Northwestern University.[1] Using the annotator tool, we created an annotated corpus of 200 randomly selected pathology notes. At the polyp finding level (histology + location match), we achieved sensitivity of 0.95 and PPV 0.95. We also validated the algorithm at a cohort level, counting as a case anyone with a colon polyp finding. Results of evaluation against a corpus of 50 cases and 50 controls were: 94% PPV (cases) and 98% PPV (controls). The algorithm and tools were subsequently shared with Geisinger to perform secondary validation. Geisinger's preliminary validation results on 25 cases and 25 controls were: 100% PPV (cases) and 96% PPV (controls). In order to achieve this performance, the NLP source code had to be modified to reflect differences in the structure of Geisinger's pathology notes. These modifications were confined to script files regulating the document sectionizer and relation extractor components. Marshfield and Group Health performed a manual validation on a convenience sample, leading to similar modifications to the NLP pipeline. After running the algorithm across all four sites, we were able to extract a genotyped cohort of 5839 patients, including 2233 from Northwestern, 2597 from Marshfield, 609 from Geisinger, and 400 from Group Health.

Discussion

Although porting EHR-driven algorithms across sites remains a difficult challenge, the tools we describe here are an improvement over sharing non-executable text descriptions and flowcharts. Their use decreased implementation time and ensured consistency of output across sites, which is especially relevant in the context of eMERGE where patient results are pooled for genetic analysis. The KNIME platform has proven itself to be a particularly useful mechanism for sharing algorithms as executable workflows, and has been subsequently used in eMERGE to implement a cohort selection algorithm for patients with abdominal aortic aneurysm.[2] One limitation of the approach taken here is the necessity of modifying the NLP source code to accommodate differences in note structure across different sites. To improve this process, we are currently working on extending the KNIME platform with fully integrated support for clinical NLP, enabling users to configure specified aspects of an NLP algorithm (such as sectioning documents and specifying relation extraction rules) without the need to modify and re-compile source code.

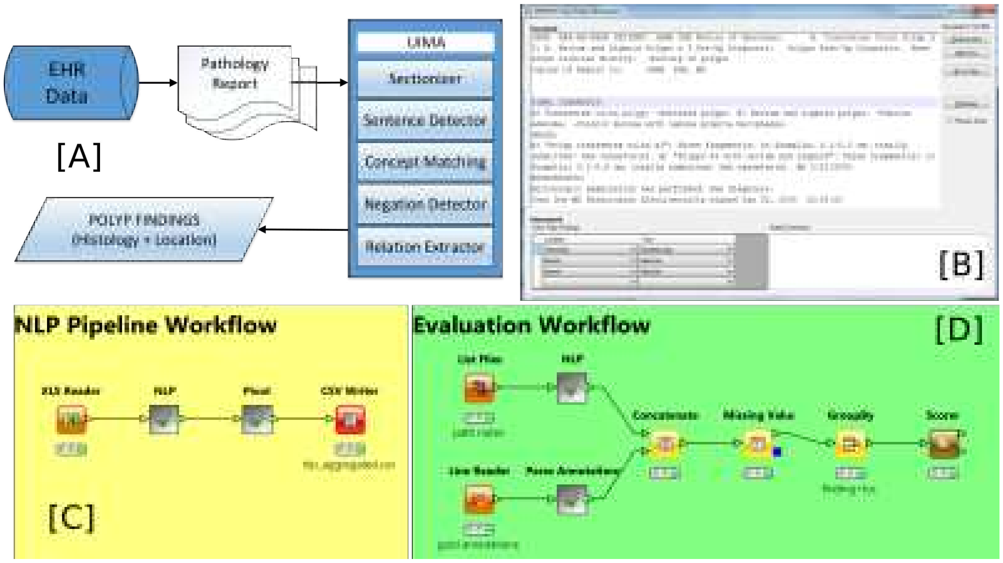

Figure 1: [A] NLP pipeline; [B] Document annotator tool; [C] KNIME workflow for executing NLP; [D] KNIME workflow for evaluation of results against gold standard

References

[1] Gawron AJ, Thompson WK, Keswani RN, Rasmussen LV, Kho AN. Anatomic and advanced adenoma detection rates as quality metrics determined via natural language processing. The American Journal of Gastroenterology. 2014 Dec;109(12):1844–1849.

[2] Borthwick KM, et al. ePhenotyping for Abdominal Aortic Aneurysm in the Electronic Medical Records and Genomics (eMERGE) Network: Algorithm Development and Konstanz Information Miner Workflow. International Journal of Biomedical Data Mining. 2015;4(113).