MEDINFO 2021: One World, One Health – Global Partnership for Digital Innovation
P. Otero et al. (Eds.)

doi:10.3233/SHTI220268

A Natural Language Processing Tool Offering Data Extraction for COVID-19 Related Information (DECOVRI)

Paul M. Heider[a], Ronak M. Pipaliya[b], Stéphane M. Meystre[a]

[a] Biomedical Informatics Center [b]College of Medicine, Medical University of South Carolina, Charleston, SC, USA

Abstract

A new natural language processing (NLP) application for COVID-19 related information extraction from clinical text notes is being developed as part of our pandemic response efforts. This NLP application called DECOVRI (Data Extraction for COVID-19 Related Information) will be released as a free and open source tool to convert unstructured notes into structured data within an OMOP CDM-based ecosystem. The DECOVRI prototype is being continuously improved and will be released early (beta) and in a full version.

Keywords:

COVID-19; Natural Language Processing; Machine Learning

Introduction

As part of the Medical University of South Carolina's (MUSC) efforts to combat the COVID-19 pandemic, a central database (COVID Datamart) was designed to collect all electronic health record (EHR) data for patients treated or assessed for COVID-19 at MUSC. The primary means for the general public to be tested at MUSC required using a telehealth application (Zipnosis) to collect symptoms, medical history, and COVID-19 exposure data and interact with medical professionals. The results of these interactions were included in MUSC's EHR system as natural language generated unstructured text. Initial efforts in March 2020 included the development of a new NLP application prototype to extract COVID-19 related information from this text. Subsequent efforts focused on multiple performance improvements, expansion of the types of clinical information extracted from text notes and generalization to a large variety of clinical text notes. The resulting system – DECOVRI (Data Extraction for COVID-19 Related Information) – is presented below and released as open-source software.

Methods

DECOVRI is built on the Apache UIMA framework (specifically, uimaFIT [1]) and combines modules we used off-the-shelf, modules we reused, and new custom developments (Fig. 1). Core features include multiple input and output type readers (i.e., from/to both file system and database), text pre-processing (tokenization, sentence boundary detection), sectionizing, and dictionary look-up for symptoms and comorbidities. In addition, rules and regular expression are used to extract demographics and exposure risk information. Two modules based on deep neural networks (Bidirectional LSTM Recurrent Neural Network [2]) extract medication and laboratory test information, respectively. Finally, local context analysis (e.g., negation, information subject, uncertain) completes text processing. All extracted information is exported using the OMOP common data model (CDM). The general architecture was split across multiple machines for performance reasons with four primary machine clusters: the database, uimaFIT, Keras, and log monitoring. Custom SSL connections are used between uimaFIT and the Python-based (Keras) Recurrent Neural Network. DECOVRI automatically runs daily (cron job) to process any notes newly added to MUSC's COVID Datamart and older notes flagged for reprocessing. Downstream uses for this data initially included a purely data-driven symptom "checker" giving testing recommendations to patients worrying about COVID-19 [3, 4]. Later uses included predicting SARS-CoV-2 test results to then enable more efficient testing with pooled samples.

For testing, an initial small corpus of 15 randomly selected clinical notes was annotated by domain experts and used to evaluate the information extraction accuracy of the prototype. A new larger reference standard of 400 clinical notes for training and further evaluations is currently in development.

Results

The initial prototype (alpha version) of DECOVRI was assembled between March 16th and March 26th, 2020. Every week through the end of April, new modules were added or pre-existing modules were upgraded. Module upgrades included changes required by our shifting understanding of COVD-19 (e.g., adding new symptoms to lexicons), changes required by internal technical forces (e.g., the sectionizer had to adapt to new standard templates), and changes brought about by performance bottlenecks (e.g., the watermarking system and batch reprocessing logic had to be adjusted as caseloads and note backlogs grew). Features were integrated across modules, as well. For instance, section and negation information impacted which terms were written to the OMOP CDM NOTE_NLP table and which, if any, term_modifier flags were included. Beyond the standard Apache UIMA CAS XMI output and OMOP CDM NOTE_NLP table output, we generated brat and CAS XMI files optimized to work as input to WebAnno [5], a web-based annotation tool. We also designed a NOTE_SEGMENTS table to parallel the NOTE_NLP table for holding the entirety of a section span in each entry. (NOTE_NLP extractions are too limited in size to fit this need.) This separate table allowed for other researchers to easily focus their investigations into a select set of the most promising sections without having to repartition all notes.

DECOVRI's primary performance bottleneck currently occurs between the core UIMA pipeline and the Python deep learning modules. The overhead to create secure socket connections

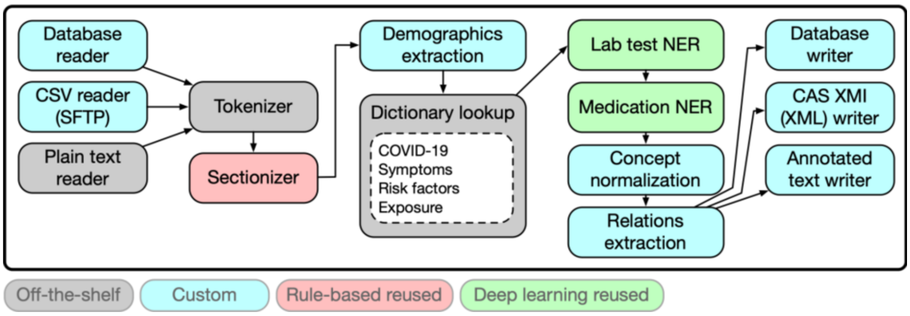

Figure 1 – DECOVRI Components and Architecture

dominates all other speed issues. However, the bottleneck is still not sufficient to hinder standard throughput rates as these only reach 100-200 sentences per hour.

In terms of information extraction performance, the stable deployed version (that has been running for the second half of 2020 and the first quarter of 2021) has higher recall than precision in almost every evaluation category when compared against our 15 note reference corpus. As seen in Table 1, the micro-averaged F_1-measure is 0.7254 and the macro-averaged F_1-measure is 0.8316. The high overlap between COVID-19 symptoms (e.g., 'cough') and comorbidities or problems (e.g., children of 'respiratory disease' in the UMLS ontology) caused conflicts in our automatically generated lexicons. More work needs to be done to allow control over how lexical entries are assigned to concepts with high overlap within lex_gen [6] (an open source tool for automatically generating lexicons from the UMLS Metathesaurus given seed concepts) and within the pipeline itself to prioritize or resolve overlapping annotations.

Table 1– Information Extraction Performance Metrics

	Precision	**Recall**	**F_1-measure**
micro-average	0.7204	0.7304	0.7254
macro-average	0.8145	0.8579	0.8316

The final main pain point in our architecture was tracking output and extractions from multiple pipeline versions in a workflow that easily accommodated comparing differences between versions (e.g., for regression testing), archiving old version, and supporting batched or watermark-based reprocessing.

Conclusions

One of the more important take-aways was the clear benefits of separating the unstructured data extraction from the data modeling. Because we stored our unstructured extractions to a structured database as its own self-contained step, developing new tools and models derived from this information was an unencumbered task.

In the end, the speed of development and simplicity of integration was driven by our backlog of prior projects built around a shared paradigm of standard tools. We could freely refactor Apache uimaFIT components, OMOP CDM modules, SHARPn types, UMLS-based lexicons, etc. from previous work. For instance, context analysis was not part of the alpha version because we did not have a functioning module for it but will be part of the early access beta release. Likewise, after the integration of our deep learning-based medication and laboratory test extractors, we now have an architectural template for multi-machine, hybrid Java/Python pipelines. Code and instructions for deploying DECOVRI will be available at: https://github.com/MUSC-TBIC.

Acknowledgements

This work was supported in part by funding from PCORI (ME-2018C3-14549).

References

[1] P. Ogren, S. Bethard, Building Test Suits for UIMA Components, Proceedings of the Workshop on Software Engineering, Testing, and Quality Assurance for Natural Language Processing (SETQA-NLP 2009), (2019).

[2] N. Reimers and I. Gurevych, Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP), Copenhagen, Denmark, 2017, pp. 338–348.

[3] S.M. Meystre, Y. Kim and P.M. Heider, COVID-19 Information Extraction Rapid Deployment Using Natural Language Processing and Machine Learning, in: AMIA NLP WG Pre-Symposium, 2020.

[4] S.M. Meystre, P.M. Heider and Y. Kim, COVID-19 Diagnostic Testing Prediction Using Natural Language Processing to Power a Data-Driven Symptom Checker, in: AMIA Summits Transl Sci Proc, 2021.

[5] R. Eckart de Castilho, É. Mújdricza-Maydt, S.M. Yimam, S. Hartmann, I. Gurevych, A. Frank, and C. Biemann, A Web-based Tool for the Integrated Annotation of Semantic and Syntactic Structures, in: Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH), The COLING 2016 Organizing Committee, Osaka, Japan, 2016: pp. 76–84

[5] P.M. Heider and S.M. Meystre, Targeted Terminology Generation Tool for Natural Language Processing Applications, in: Presented at the AMIA NLP-WG Pre-Symposium, 2019.

Address for correspondence

Paul Heider at heiderp@musc.edu