Cristina Ross
CS410 Fall 2022
NetID: cb37

Best toolkits for healthcare note analysis

A recent poll in the United States stated that nearly 80% of office-based physicians and 96% of acute care hospitals have implemented a certified electronic health record (EHR) and with this wide adoption has come a boom in Big Data in healthcare.[1] Analyst and data scientist alike are able to access a wealth of knowledge about patients from digital imaging, clinical information, payor records, genomic sequencing, and many other sources. This diverse dataset allow these healthcare IT professionals to draw insights leading to advancements such as better patient access through efficient appointment scheduling and clinical decision support with intelligent alerting and early decision detection. Advancement like these and many more provide a promise to transform the healthcare space but this promise ends there, at simply a promise.[2] The problem with looking at Big Data to gain insights on a patient is that we attempt to cut out the initial source of data and our greatest resource - the clinician. The clinicians are the ones ordering a lab for us to evaluate, ordering a scan producing the images we see, interacting with the patient to understand the issue, and they take all of that information, process it and write down their understanding and plan for treatment in their notes along with the data supporting their opinion. The notes from physicians and nurses are a one-stop shop for a wealth of information on the patient and are our main target as a data source.

Looking at healthcare notes as a data source is not a novel concept, in fact there are multiple tools that use natural language processing (NLP) to help facilitate the parsing of healthcare notes. In reviewing these methods, including a general and widely used python program Natural Language Toolkit (NLTK), a published method for extracting note data, and two commercially available programs Spark NLP and cTakes, it was found that there is still a gap to be filled for an open-source robust python program or extension to fully capture insights from the free-text clinician note.

The first NLP engine we will explore is NLTK, a popular program for analyzing natural language with a variety of libraries for several common NLP steps such as parsing, stemming, semantic reasoning, and many more.[3] When using NLTK in the context of a healthcare note, we have many positive attributes including parsing a note into words accomplished due to the English sentence structure separated by spaces, stemming words to consolidate multiple forms of words into a single term, and part of speech tagging to understand how words are being used. This basic language processing is particularly important for breaking down sections of a note with free-text, which are arguably the most useful and unique insights we have for a patient. While there are many important insights in these sections of the note, they cannot give us the entire picture of the patient. For the full picture, we would also need the structured sections of the note, including patient registration information, historical illness information, allergies, imaging studies, tables of laboratory test results and vital signs. The structured sections are where NLTK fails, as we need to capture data that does not occur in the standard English sentence format using several different approaches such as key-value pairs for vital signs, parsing distinct sections of the note based on headers, and parsing short sentences that are associated with a particular imaging study. Overall, NLTK is a great starting point, but would require a significant extension with incorporated domain knowledge for it to be useful towards our goal of extracting insights from healthcare notes.

Published in 2021, our second method aims to do what NLTK could not - extract structured data from the electronic healthcare note.[4] This method makes use of note templates provided by certain EHR vendors, which provide a list of options for clinicians to include (and sometimes auto-populate) specific fields available to the user. There are many positive attributes of this method, the first of which is that unique sections are able to be identified quickly and completely based on their back-end coding, and secondly that they make use of a parsing program that returns results in a useful tabular format with either a key-value match or a binary indicator to indicate the presence or absence of the related concept. In addition, this method sites a fast retrieval and parsing time of 72 seconds and 2 seconds, respectively, for over 1,200 notes. While on the surface this program may seem to be the perfect pairing with NLTK to reach our goal, this method has many limitations. The study was done based on a back-end file structure unique to the Epic EMR and therefore could only be applied to hospitals solely utilizing the Epic system, only accounting for 37% of hospitals in the US.[5] Additionally, this method relies on a user to only use sections from the list given and not manipulate the auto-generated section, which are not reasonable assumptions for general note usage in practice. As stated by the author, this method is further limited by the data types it can handle - only supporting key-value pairs with an integer key and a string value. This method makes great strides beyond simply implementing NLTK in regards to extracting healthcare data, however the lack of EHR-agnostic generalizability and limited data types leaves this method to not be a viable solution towards our goal.

Once we shift our focus to commercially available programs, we see programs that are significantly more robust. The first such program is SparkNLP. SparkNLP is an open-source NLP library released by John Snow Labs.[6] Much like NLTK, SparkNLP offers basic NLP tools to extract relevant information from text including tokenization, part-of-speech tagging, entity recognization, and sentiment analysis. SparkNLP further their NLP libraries by including translation, summarization, question answering, text generation, establishing context, image classification, and many other advanced NLP tasks in their standard libraries. Additionally, this library is built on top of Apache Spark, an engine used for large-scale data processing. The fact that it is built on top of a data processing engine ready for implementation in a Big Data environment is a big step up from the previous programs and brings up a topic we have not yet discussed, which is scalability - an essential topic when discussing parsing multiple multi-page notes for each patient visit to the hospital in a desired cohort. On top of their robust NLP engine, SparkNLP incorporates domain-specific NLP variations including clinical entity recognition and clinical entity linking, models for domain-specific tasks such as de-identification of patient information required for some research tasks, and over 700 other pre-trained healthcare-specific models. At first glance, this is the perfect library for our goal. However the healthcare lens over SparkNLP is marketed as the separate product of HealthcareNLP and is only available through the cost prohibitive Enterprise version of SparkNLP.[7] While this product does not meet our requirement of an open-sourced program, it does provide us with a clear model of what a program would need to comprehensively hit our target.

We must turn once again to attempt to identify a library that moves toward our goal but remains open-sourced and so we take a look at Apache cTakes (clinical Text Analysis and Knowledge Extraction System). Apache cTakes is an NLP engine that extracts clinical information from EHR unstructured text.[8] Apache cTakes takes the unstructured text data in a healthcare note, applies NLP techniques and outputs all of the clinical terms, with their associated NLP tags including healthcare mapped code, context, part-of-speech, and negation. While cTakes does a great job at entity recognition within the healthcare space, there are significantly fewer pre-trained models than in the enterprise SparkNLP. The lack of pre-trained models is replaced with flexibility as cTakes allows functionality to build unique models to be trained on a specific dataset towards a specific purpose, albeit this functionality is lost without adequate enough data in a dataset to train a model. While cTakes would be a great jumping off point, it also only supports the Java programming language, which can still be used as is within a python modeling environment but requires Java programming to create the needed additional NLP pre-processing models.

After reviewing some of the most promising options to meet our goal of achieving a robust open-sourced NLP engine that provides a complete array of clinical insights from different types of healthcare notes, it can be concluded that there is not one solution that can completely reach our goal. The most promising options that get us most of the way to our goal lie in the two commercially released engines: Apache cTakes and SparkNLP. While the open-sourced NLP engine Apache cTakes incorporates domain knowledge it lacks additional functionality supported by pre-trained models. Open-sourced SparkNLP is an extremely robust traditional NLP engine, but lacks both the domain knowledge and pre-trained domain-specific models to add functionality. In conclusion, there is not an open-sourced NLP engine specific to the healthcare space that is robust enough to extract a full array of clinical insights from a variety of different note types, but existing packages could be used with a robust extension to fully extract insights for use in furthering predictive and prescriptive modeling.

References

1. https://www.healthit.gov/data/quickstats
2. https://catalyst.nejm.org/doi/full/10.1056/CAT.18.0290
3. https://www.nltk.org
4. https://link.springer.com/article/10.1007/s11606-020-06110-8
5. https://www.definitivehc.com/blog/top-hospitals-using-epic-ehr
6. https://nlp.johnsnowlabs.com
7. https://www.johnsnowlabs.com/spark-nlp-health/
8. https://ctakes.apache.org