Lab 4.1 - NLP in Healthcare

David Strube

dstrube3@gatech.edu

***Abstract*—**Interoperability is a significant challenge in the field of Health Informatics. The use of Natural Language Processing (NLP) could be used to help mitigate this challenge. In this paper, the topic of NLP in healthcare will be explored, and ClarityNLP in particular will be discussed.

1 BACKGROUND

NLP is the means by which human languages (as opposed to programming languages) are made understandable by computers. When a human language (like English) is made understandable by computers, then large amounts of information can be processed more quickly and efficiently than any human.

Some common tasks that can be found in NLP are: Text and speech processing (Yi C, Tian Y, 2012), Morphological analysis (Gyansetu, 2020), Syntactic analysis (Klein D, Manning CD, 2002), and Higher-level NLP applications like Book generation (UBUWEB, Racter, 1984). For the purposes of this paper, we shall stick to exploring text processing, as that is what ClarityNLP specializes in.

2 SIGNIFICANCE

Every day, in clinics and hospitals around the world, large amounts of information comes in about patients and their conditions. Sometimes it is too much for doctors and nurses to keep up with, which can lead to mistakes that lead to suffering and death. If NLP is utilized more in these settings, many hours of precious time can be salvaged, billions of dollars can be saved, countless lives may be improved, and many unnecessary deaths can be prevented.

3 NLP IN HEALTHCARE

Phenotype is a term in genetics that refers to the collection of observable traits or characteristics of an organism. ClarityNLP is a framework for clinical phenotyping; that is, it is a means of taking a set of observable traits or characteristics of natural language and making them accessible to computers. ClarityNLP utilizes NLP techniques along with a powerful query language, NLPQL, to extract information from text to identify patients and their medical conditions.

4 HEALTHCARE NLP IN ADDRESSING SPECIFIC TOPICS

One example use of ClarityNLP is extracting patient information from a nurse's notes (ClarityNLP, 2018a). In this example, the phenotype is defined with the name "Patient Temperatures" with a specified version. The version is specified so that future iterations of this phenotype can be distinguished from older ones. The library `ClarityCore` is imported with the keyword `include` and shortened to just `Clarity` for the sake of concise calls to the library. A set of documents named *NursingNotes* is defined and instantiated with the following: `Clarity.createReportTagList(["Nurse"]);`. A set of terms named *TemperatureTerms* is defined and instantiated. The range of expected string values indicating the presence of a temperature term includes: "temp", "temperature", and "t".

Next, the phenotype features are defined. A feature is the text data that is being sought out in the source text to be extracted and to represent information about a patient. A feature represents a task for the NLP query language (NLPQL) to be executed, along with parameters (which are not mandatory) like patient cohorts, term sets, document sets, and others. In this example, a feature named *Temperature* is defined as calling the Clarity function `ValueExtraction` which takes on parameters for the termset, the documentset, a minimum value of 96, and a maximum value of 106 (for degrees in Fahrenheit). (Alternative to the call to ValueExtraction, one could use the External NLP Method Integration to specify a method call to some other Python library or an API external to this project (ClarityNLP, 2018b).) Finally, the `final` feature named `hasFever` is defined as a boolean function that returns true if the patient's temperature in the nurse's notes is greater than or equal to 100.4. Note that the `final` keyword in the definition of the feature indicates that the output of the `hasFever` feature is the output that is being sought from this phenotype.

When ClarityNLP is installed and running, one can run queries on some input text by calling the NLPQL API endpoints on one's local server, thereby efficiently parsing through text data to extract information from large amounts of text faster than any human could.

5 REFERENCES

1. Yi C, Tian Y. (2012). Assistive Text Reading from Complex Background for Blind Persons. Camera-Based Document Analysis and Recognition, Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-29364-1_2. Accessed April 2, 2021.

2. Gyansetu. (2020). What is Natural Language Processing? Intro to NLP in Machine Learning. Gyansetu. https://www.gyansetu.in/what-is-natural-language-processing/. Accessed April 2, 2021.

3. Klein D, Manning CD. (2002). Natural language grammar induction using a constituent-context model. Advances in Neural Information Processing Systems. https://proceedings.neurips.cc/paper/2001/file/2d00f43f07911355d4151f13925ff292-Paper.pdf. Accessed April 2, 2021.

4. UBUWEB, Racter. (1984). The Policeman's Beard Is Half-Constructed. https://www.ubu.com/historical/racter/index.html. Accessed April 2, 2021.

5. ClarityNLP. (2018a). ClarityNLP at a Glance - ClarityNLP documentation. https://claritynlp.readthedocs.io/en/latest/user_guide/intro/overview.html. Accessed April 4, 2021.

6. ClarityNLP. (2018b). Custom Task Algorithms - ClarityNLP documentation. https://claritynlp.readthedocs.io/en/latest/developer_guide/custom/custom.html. Accessed April 4, 2021.