**Data Science Research Group n2c2 2022 Task 1**

**Medication Extract**

To extract medications from the clinical notes, we will apply different NER techniques including rule-based, CRF model, and DNN+CRF. We will start with processing the clinical notes, extracting sentences, extracting features, etc. The following is a list of utility functions we need.

**1. def extract\_sentences(notesId, notesText): Yuhao Zhang and Carl Shen**

Input: notesId: clinical notes id

notesText: the text content of the clinical notes

Output: a DataFrame with columns {‘NotesId’, ‘SentenceId’, ‘Sentence’, ‘Start’, ‘End’, ‘Section’, ‘PrevSentId’, ‘NextSentId’}

This function breaks the input clinical notes into sentences. The output is a DataFrame containing information about sentences in the input clinical notes

notesId: the id of the clinical notes

SentenceId: the identifier of the sentence in the format of notesId\_xxxx

Sentence: the text of the sentence

Start: the start position of the sentence in the notes

End: the end position of the sentence in the notes

Section: the section name of the clinical notes such as ‘NARRATIVE’, ‘PHYSICAL EXAMINATION’

PrevSentId: the Id of the previous sentence

NextSentId: the Id of the next sentence

**2. def extract\_word\_labels(a record from the function extract\_sentences(), ann): Ziao You and Xiyuan Chang**

Input: an output record of the extract\_sentences() function and the annotation file of the clinical notes

Output: a DataFrame containing individual words and medication labels {‘SentenceId’, ‘Word’, ‘Label’}

This function splits a sentence into individual words and finds the mediation labels from the given annotation file. The output DataFrame contains the following 3 columns:

SentenceId: the id of the sentence

Word: an individual word in the order it appears in the sentence

Label: using {B, I, O} to label the word as

B: if the word is the beginning of a medication

I: if the word is a following word of a medication (multiple word medication)

O: if the word is not a medication

**3. def extract\_word\_linguistic\_features(a record from the function extract\_senteces()): Haoran Zhao (19) and Yantian Ding**

Input: an output record of the extract\_sentences() function

Ouptut: a DataFrame containing the lexical information of the individual words {‘SentenceId’, ‘Word’, ‘Tag1’, ‘Tag2’, …..}

This function uses the spacy tool to extract all linguistic features of each word

SenteceId: the id of the sentence

Word: an individual word in the order it appears in the sentence

Tag1: a linguistic feature

Tag2: a linguistic feature

...

**4. def extract\_word\_RxNorm\_features(a record from the function extract\_sentences(), RxNorm): Xiao Fang**

Input: an output record of the extract\_sentences() function

Output: a DataFrame containing the features related to RxNorm {‘SentenceId’, ‘Word’, ‘feature1’, ‘feature2’, …}

This function extracts any features about individual words related to RxNorm

SentenceId: the Id of the sentence

Word: an individual word in the order it appears in the sentence

Feature1: a feature related to RxNorm

Feature2: a feature related to RxNorm

….

**5. def extract\_word\_dep\_features(a record from the function extract\_sentences(), dependency tree): Harry Zhao and Yi Pan**

Input: an output record of the extract\_sentences() function

Output: a DataFrame containing the features related to dependency tree {‘SentenceId’, ‘Word’, ‘feature1’, ‘feature2’, …}

This function extracts any features about individual words related to dependency tree

SentenceId: the Id of the sentence

Word: an individual word in the order it appears in the sentence

Feature1: a feature related to dependency tree

Feature2: a feature related to dependency tree

…..

**6. def word2feaures(DataFrame containing the words and feature of a sentence, i):**

Input: a DataFrame containing the words and features of a sentence and the position of the word in the sentence

Ouptut: a feature dictionary for crfsuite

==================================================

xx. def clean\_sentence(sentence):

Input: the text of a sentence

Output: cleaned sentence by normalizing the words of the sentence such as lowercasing, stemming, lemmatization, removing abnormal symbols