Medical Named Entity Recognition on Twitter Data

March 28th, 2016

Submitted By:-

Team number :- 56

Juhi Tandon - 201225032

Kalpish Singhal - 201505513

Chinmay Bapna - 201302182

Description (Problem Statement):

Medical Entity Recognition is a crucial step towards efficient medical texts analysis.

The task of a Medical Name Entity Recognizer is twofold -

(i) identification of entity boundaries in the sentences.

(ii) entity categorization.

Our objective is to extend medical entity recognition for tweets.
Medical entities can be diseases, drugs, symptoms, etc. Previously, researchers in the field have used hand crafted features to identify medical entities in medical literature. It has been found that in contrast with semantic approaches which require rich domain-knowledge for rule or pattern construction, statistical approaches are more scalable.

Dataset:

We have a dataset of 1 year of tweets about 4 diseases and 32 drugs. A team of domain experts has annotated about 2000 tweets with entities (around 20 types: diseases, drugs, symptoms) and also relations (around 40 relation types: cures, causes, etc).

Implementation:

We are using Supervised learning with hand crafted features to predict the Medical NER tag.
For this preliminary submission the tags in consideration are "**DRUGS", "DISEASE", "SYMPTOM-OR-SIDE-EFFECT".**

Mallet , a java based package has been used for the classification task.
Two models have been trained. The features used are :
LEMMA
LEMMA and 5-GRAM CONTEXT WINDOW

Instructions to run the codes:

- Execute the bash.sh_5gram or bash.sh_1gram in your data/twitter/ folder
- Set the java_home variable according to your system
- The script makes a combined training feature file with tag, across all the folders using features.py
- It trains a model using mallet
- It similarly makes a testing feature file across all folders
- It then gives the testing file to the model to predict the tag and calculates accuracy comparing it with gold data.

Accuracy Obtained:

With only lemma as feature : 0.873522458629
With lemma and n-gram as feature : 0.873522458629

We also tried feature induction and varying intensity of weights for different features using available flags in the tool mallet. Some of the options that were explored while making the training model and the accuracies that we achieved from the same are as follows:

- --feature-induction true --weights some-dense --default-label None : 0.874704491726
- --feature-induction true --weights sparse --fully-connected false : 0.877541371158
- --feature-induction true --weights some-dense --fully-connected false : 0.877541371158
- --feature-induction true --weights dense: 0.876241134752
- --feature-induction true --weights some-dense : **0.878132387707**
- --feature-induction true --weights sparse : **0.878132387707**