**CRFClassifer (https://nlp.stanford.edu/software/CRF-NER.shtml)**

This classifier is used for Named Entity Recognition (NER) and uses Conditional Random Field sequence model. More details in the URL link provided. This code in a different version is used in our NLP pipeline. **I have provided just an example usage and is not the production implementation.**

Logical steps of execution –

1. Clean the document, as in remove everything except 1 product description per row (which will be addressed as 1 document going forward)
2. The list of documents is stored as flat file
3. Tokenize each document (input flat file) and remove special characters which dirty tokens
4. Manually tag the tokenized data for required NER as UOM (unit of measure) and QTY (quantity)
5. Train the model with the Step 4 tokenized file and save the model as a serialized output to disk
6. Test the model with test data

Sampling of data

1. After analyzing the data, stratified sampling was done per UOM, however the since LABELLING of tokens needs manually intervention, I was not able to create an enough TRAINING sample dataset.

Output analysis

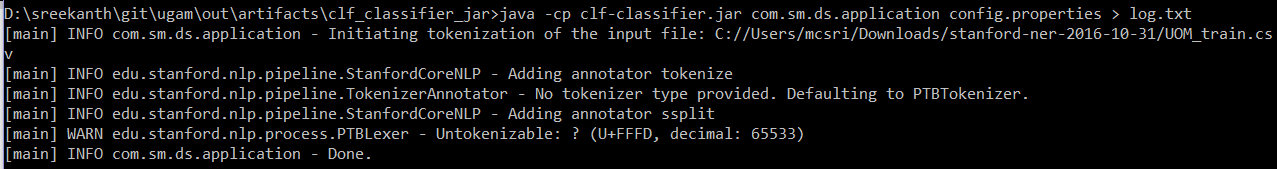
1. In the absence of training dataset, the model performed very bad.
2. The model approach and functionality was tested against publicly available examples and the examples worked as excepted.
   1. Training - <https://nlp.stanford.edu/software/ner-example/jane-austen-emma-ch1.txt/>
   2. Testing - <https://nlp.stanford.edu/software/ner-example/jane-austen-emma-ch2.txt/>
3. For the NER to work efficiently we need a bigger training sample (more of manual labor)

Java App execution using CMD

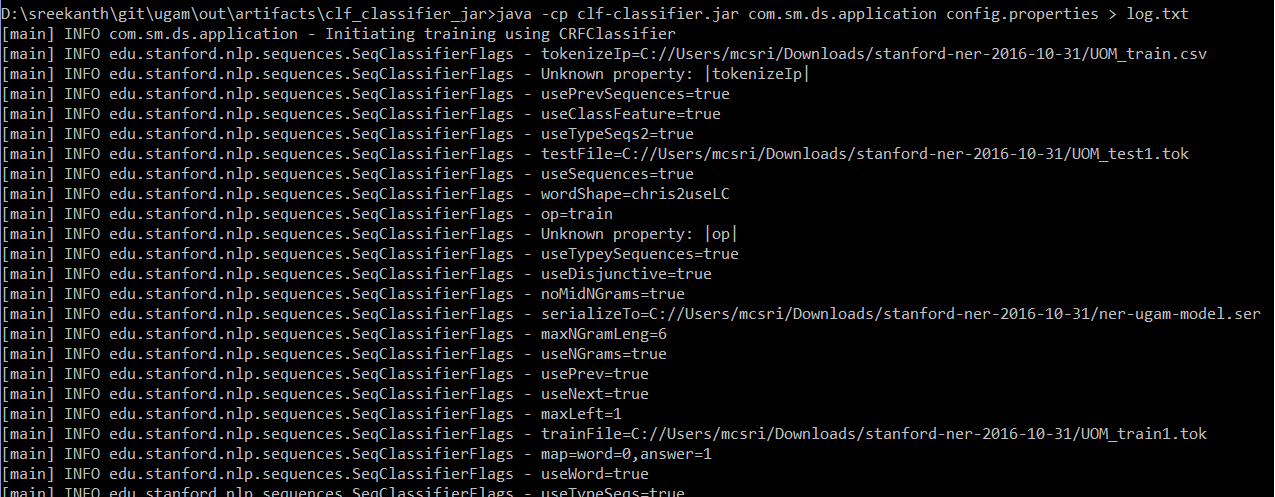
* java -cp clf-classifier.jar com.sm.ds.application config.properties > log.txt

1. result.txt – contains output of model
2. config.properties – contains all the model settings and has to be supplied to the jar with correct documents. More details about the config.properties is available in comments of the file.
   1. Select correct “op” by comment/uncomment to do appropriate operation

Tokenizing output



Training output



Testing output (Please see the log file for more like confusion matrix)

