Team number 56

Submitted By:-

Kalpish Singhal 201505513

Chinmay Bapna 201302182

Juhi Tandon 201225032

Medical Named Entity Recognition on Twitter Data

April 14, 2016

Overview:-

Medical Entity Recognition is a crucial step towards efficient medical texts analysis. *The task of a Medical Name Entity Recognizer is two fold.*

(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).

Project Scope:-

This project is mainly focused on feature extraction of medical entities on the set of tweets. This will be divided into three phases namely.

- Feature Identification and extraction
- Model training using CRF
- Testing using trained model
- Evaluating the output obtained for different feature-models using metrics such as Precision, Recall, F-score and Accuracy.
- Selecting some feature models and plot their precision , recall , F-scores for each label.

Challenges Faced:-

Some of the challenges faced are as follows:

A. Term Variability

Term variation is a major challenge for medical NER. Term variability refers to the problem of expressing the same concept using different lexical representation, including synonyms, acronyms, word-order and derivational variations. They may result in inadequate coverage. Synonyms and word-variation in particular tend to be common in medical terminology. For example, 'birth defect' is also referred to as 'congenital abnormality', 'congenital disorder', 'congenital defect', 'congenital malformation', or 'deformity'.

B. Term ambiguity

Term ambiguity is another challenge for term classification in the biomedical domain. Term ambiguity arises from terms that have several senses or meanings in different contexts. For example, the term 'cold' is associated with several senses in the UMLS Metathesaurus:
(i) cold sensation (Physiological Function)
(ii) common cold(Disease or Syndrome)and
(iii) cold temperature (Natural Phenomenon or Process)
Finally, cold therapy is a potential fourth sense not recognised by UMLS. In the medical domain, term ambiguity has largely been reported with regards to acronyms/abbreviations.

C. Abbreviations and acronyms

High occurrences of acronyms and abbreviations have been observed. Even if the recognition phase (identifying a mention of an acronym or abbreviation) is successful, it is estimated that acronyms are overloaded 33% of the time, resulting in significant challenge to a subsequent classification task [20]. For example, 'HD' may refer to 'Hansen disease', 'Hodgkin disease', 'Huntington disease', or refer to the temporal expression 'hospital day'. The most common method reported to handle abbreviation and acronyms are automated expansion of short-hand terms (using domain knowledge) during pre-processing. However, there is a lack of evaluation of these in terms of gain.

D. Term complexity

Complex medical terms include multi-word or long descriptive concept phrases, e.g., 'subtle decrease flow signal within the sylvian branches'. Complex names may arise from term variability. For example, drug names may be represented by simply the drug name on its own, or the drug name with drug descriptors or some combination of (i.e., regime (dosage), frequency, and routes of administration), in brackets or parenthesis nested in various orders. Drug names in particular have been reported as problematic for classification and understanding of the internal structure of concepts.

Challenges specific to tweets-

1. Tweets are informal, noisy with linguistic errors and idiosyncratic style which degrades the performance of NLP tools on them.
2. Learning distributed representations for medical tweets.
3. Use of Non standard medical terminology
4. Spell correction and normalisation is difficult

These obstacles limit the scalability of methods relying on dictionaries and/or gazetteers. Domain independent techniques such as machine learning and natural language processing tools helped in facing such challenges.

Applications of Medical NER:

1. To get feedback for different types of treatments available/applicable on certain diseases (for which research is still ongoing) or mutation in disease causing microbes making them resistant to previously working drugs.
2. For getting information on side-effects of new drugs.
3. For getting information on symptoms of newly evolving diseases.

Tools used:-

- **Mallet** :- A Java-based package for statistical natural language processing,text classification and information extraction tool using Command line scripts.
  MALLET includes sophisticated tools for document classification: efficient routines for converting text to "features",a wide variety of algorithms eg. CRF
- **Meta-Map** : -MetaMap is a highly configurable program developed to map biomedical text to the UMLS Metathesaurus or, equivalently, to discover Metathesaurus concepts referred to in text.
- **Tweet-NLP** :- Provide a tokenizer, a part-of-speech tagger, hierarchical word clusters, and a dependency parser for tweet. We used it to address the problem of part-of-speech tagging for English data from the popular microblogging service Twitter. The tool reports tagging results nearing 90% accuracy.

Feature Analysis :-

Our aim was to find the most suitable set of features for the task. We carried out our experiments with different set of features, analyzed our results and then introduced new features based on the analysis.

A generic line in feature file looks like this:

Token Meta-tag PoS-tag Ortho-tag average_word_length 4_prefixes 4_suffixed cluster-id next-Noun next-Verb token-1 Meta-tag PoS-tag token-2 ....token+1 Meta-tag PoS-tag token+2 Meta-tag PoS-tag.... Label

Word and Word-Context:

Since we are constructing features term-wise, the term holds utmost importance in categorising its nature. Apart from that, it also a fact that “similar words lie together,” that is a word/term’s gravity/weight is enhanced by the terms lying closed to it. Words in a sentence

form a sequence, and the decision on a word's category can be influenced by the decision on the category of the preceding word.This is the main reason why we have considered a 2-term window (on left and on right) as features for categorising each term. Improves efficiency too, so useful.

POS:

POS tags are useful because tags of concern skewed towards one class , nouns mostly. But, majority of entities are 'None'-type entities and they can lie in any other part of speech and so can symptoms so thats a problem. Examples:
Microbial -> Adjective, **Metabolic** -> Adjective, **Adverse** -> Adjective ,**Exacerbation** -> Noun, **Airways** -> Noun, etc - symptoms mostly- Nouns and Adjectives.
Diseases and drugs always Nouns.
Still it improves efficiency, since very useful in categorising drugs, diseases and none-entities.

Metamap:

Metamap - used because they are domain oriented tags and diseases, symptoms, drugs lie in corresponding medically specific class-clusters - which allows it to identify and categorize medical terms significantly. Metamap uses a UMLS Metathesaurus dictionary made specifically for this purpose according to which it assigns tags correspondingly..
Definitely useful for NER. **Example**:
dsyn for disease, **phsu**, **imft** for drugs, **sosy** for symptoms, etc.
Improves efficiency in categorising disease, drugs and symptoms - useful.

Problems with Meta-Map

MetaMap leads however to some residual problems,which we can arrange into three classes:

(i) Noun phrase chunking is not at the same level of performance as some specialized NLP tools.

(ii) Medical entity detection often retrieves general words and verbs which are not medical entities.
(iii) Some ambiguity is left in entity categorization since MetaMap can provide several concepts for the same term as well as several semantic types for the same concept.

Orthographic:

- **Punctuations:**
  Language related features -- mostly words having special characters such as #,@,$,etc do not convey significant information for labelling necessary data, Generally most of such words will lie in the 'None' class. So, this feature is quite useful.

  Examples: #asthma -> None, #COPD -> None, @Ramkrishna -> None, etc
  We can spot a pattern here!

- **Capitalization:**
  For capital-small letters, it is a general notion that diseases start with capital letters or some pattern similar to that, Orthographic features can be used to track and learn such information. But this approach might also backfire, since we are dealing with twitter data, which is highly unstructured and makes use of informal language and hence often violates general rules of the English language... -- So using capital-small letters might not be a very good idea. Example:
  asthma -> disease, Childhood asthma -> disease, COPD -> disease,
  Multiple sclerosis -> disease, etc.
  Not much of a pattern, is there?

- **Word-Length:**
  Generically including word length as a feature might help in discriminating between open and closed classes. In addition to this we used word length with the intuition that symptoms span over more than one words and their average word length would help as their classifying feature. Drugs have the smallest word length. But twitter data is

pretty unorganised, which doesn't seem to follow any specific rules, training on something like word-length might not help much in classifying something like diseases, symptoms or drugs.

One example is use of abbreviations for diseases, symptoms, etc training it might spoil the significance/weights of other more important features.

- **Prefix-Suffix:**

  Morphological features are essentially related to the words affixes and root word.

  Using leading and trailing character n-grams in words, which help capture valuable morphological and orthographic clues that would indicate or counter-indicate the presence of NE's.

  A word can be broken into one or more meaningful words just like संधि विच्छेद in Hindi so - word-parts might help, since they might convey some meaning by themselves and this also captures starting common-letters, or trailing common-letters, etc.

Digit Pattern:

Digits can express wide range of useful information such as dates, percentages, intervals, identi ers etc. Certain patterns of digits gives strong signal about the type of named entities. For example, two digits and four digit numbers can stand for years and when followed by an "s", they can stand for a decade. Digits followed by units stands for quantity such as 10Kg. However this does not seem to help in our task of Medical NER. We did not find cases where diseases had digits in them except a few like H1N1 which might come while testing.

Next-Noun and Next-Verb:

The immediate next noun and next verb following the token in concern are added as features as NER occur in collocation and mostly correspond to noun phrase. Some of these features have proved to be very useful for NER.

Brown-Clustering:

Brown-Clustering :- Brown clustering (Brown et al., 1992) is an agglomerative algorithm that induces a hierarchical clustering of words. It takes a tokenized corpus and groups words into k clusters identified by bit strings, representing paths in the induced binary tree in which the leaves are word clusters. For most statistical NLP tasks, performance is limited by the quality of the underlying representation (i.e., features) of words. Consequently, recent research has focused on improving word representations (Bengio et al. 2003; Baroni, Dinu, and Kruszewski 2014) and shown that a simple unsupervised technique, Brown clustering, produces excellent features that are competitive with far newer approaches (Bansal, Gimpel, and Livescu 2014; Qu et al.2015).

The obtained clusters containwords that are semantically related, or are paradigmatic or orthographic variants.It is unsupervised and can be used to create powerful word representations for machine learning. The longer the common bit substring between clusters,the closer they are in the hierarchy. Clusters are capable of grouping orthographic variants and diminutives.

It is observed that apart from using context window for capturing spatial characteristics, clustering text can also give us a clearer picture of the term-semantics and doing so it is possible that semantically similar terms will get same tags. Hence this is also a useful feature for NER.

Lemma:

The lemma of a word can also provide significant information for capturing terms being derived from the same root. Such words usually represent the same thing. But since we are operating on twitter data which is highly unstructured and informal in nature, normal rules of NER such as using lemma might not work here. We tried using lemma as a feature as well, but the results weren't very impressive. Also we couldn't find a state of the Art - lemmatizer to work on twitter data which caused problems in finding roots of words having too many special characters, since we aren't tokenizing the data and are just using space separated words. Some of the characters weren't decoded properly and Unicode warning was raised.

Example: **Parkinsonâ€™s , ðŸ’šðŸ’š , haveâ€¦ , fromâ€¦ , ðŸ˜**

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”. We are following a BIO (Begin-Inside-Outside) model which allows us to provide

One among 7 labels to each term, which include:

{Disease-begin, Disease-inside, Drug-begin, Drug-inside, Symptom-begin, Symptom-inside, None}

Example for disease - Multiple sclerosis -> Multiple=Disease-Begin and sclerosis=Disease-Inside

The task given to us, can be divided into 3 main steps:

1. Making A Feature File:

- ★ Feature file consists of 2 main things - **Features and Labels.**
  Each tweet is stored in a separate file along with a corresponding annotation file with manually labelled entities. The most basic feature file was formed by iterating over each space separated term and tags extracted from matching annotation files for corresponding terms.
- ★ Next comes the word-context feature, along with the term, a 2-term context window, preceding and succeeding the concerned term were added as another feature. Example for term ‘you’ in “What are you doing today?”
  ‘you’ -> ‘are’ -> ‘What’ -> ‘doing’ -> ‘today?’
- ★ Metamap tagging: For each tweet, metamap tags were extracted (using the Metamap tool) and stored into a dictionary , which was used to assign tags when we iterated over that tweet’s terms for making feature file. These medically acclaimed tags were

added as a feature for their respective terms.

Example: dsyn for disease, phsu, imft for drugs, sosy for symptoms, etc.

★ POS tagging: For each tweet, part of speech tags were extracted (using the Tweetnlp tool) and stored into a dictionary , which was used to assign tags when we iterated over that tweet's terms for making feature file. Example:

"Rahul is suffering from asthma."

Rahul -> N , is -> P, suffering -> V, from -> P, asthma -> N.

★ Orthographic tagging: For each term, orthographic features were identified and labelled accordingly,

1. Capitalization (C): If a term contains --

   All capital letters, it is given a tag 'AA'

   All small letters its given tag 'aa'

   Capital and small mixed 'Aa'

   If only a single character and Capital , it is given a tag 'A'

   If only a single character and lowercase , it is given a tag 'a'

2. Punctuation (P): If a term contains special characters it's given a tag 'p', And if it doesn't then 'n' tag is given.

3. Digits (D): If a term contains numbers it's given a tag 'd', And if it doesn't then 'n' tag is given.

4. Average Word-Length (L): Word length is also used as a feature. For entities comprising of multiple terms, average length of those terms is applied for each term in that entity.

5. Prefix (Pre) and Suffix (Suf) : A range of prefixes and suffixes of length Ranging from 1-4 were also applied as features for each term.

Example for term Headache:

C-> Aa, P-> n ,D-> n ,L-> 8 ,Pre-> H He Hea Head , Suf-> e he che ache

★ Cluster-Id : Brown-Clustering was used to cluster combined data for all tweets from training and testing data. Each term was assigned to a cluster based on certain factors such as semantic features and word-context. These cluster-id's for every term

was also used as a feature, since terms belonging to same cluster are considered to be similar semantically.

★ Lemma: We also tried using lemma/root of a word as a feature, but were facing errors, mainly Unicode Error due to non-standard nature of tweets. A stemmer -> NLTK Porterstemmer was used for this purpose.
Root for term facilitates -> facilit

2. Running Mallet on Feature Files:

★ Separate Feature Files were made for training and testing data, Mallet was trained on the feature file made from training data and a trained model was created.

★ From the Feature file created for Testing data, labels were extracted as a separate file, which would later be used for evaluation purposes.

★ The rest of the features from the Testing Feature file (all except labels) were fed to the tool Mallet as a testing file using the trained model constructed on training data. (from step 1).

★ The output tags obtained from the previous step -> were stored in a separate file, which would now be sent for evaluation.

3. Evaluation:

★ The two files obtained from the previous process, the system(mallet) generated tag file and the tag file generated from predefined annotations will be used for evaluating the efficiency in NER for considered set of features.

★ **4 Types of metrics were applied:**

1. Accuracy: $\frac{Total\ number\ of\ matching\ tags}{Total\ Number\ of\ tags}$

2. Precision: $\frac{True\ positive}{True\ positive + False\ positive}$

$$= \frac{No\ of\ labels\ marked\ correctly\ as\ 'entity'\ by\ system}{All\ no\ of\ labels\ marked\ as\ 'entity'\ by\ system\ in\ testing}$$

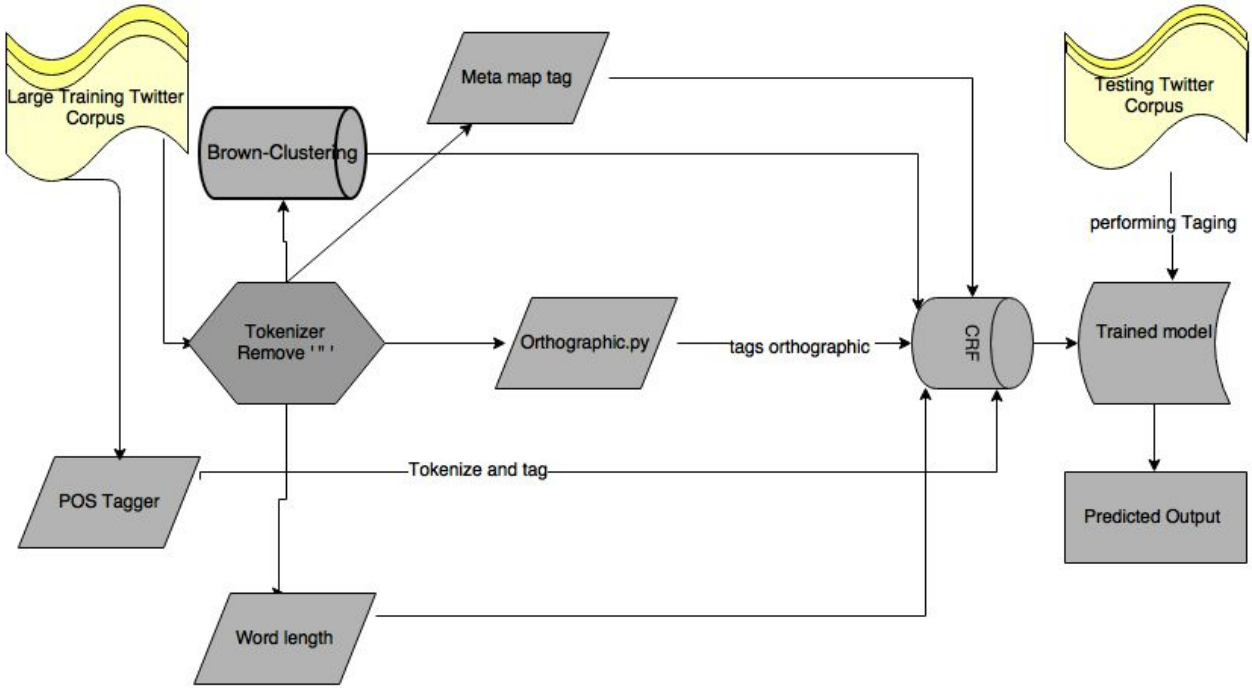
3. Recall: $\frac{True\ positive}{True\ positive + False\ negative}$

$$= \frac{No\ of\ 'entity'\ labels\ correctly\ marked\ by\ system}{Actual\ no\ of\ 'entity'\ labels\ in\ testing\ file}$$

4. F-Score: $\frac{2 \times Precision \times Recall}{Precision + Recall}$

All these steps are carried out in a proper sequence . We have created a bash file, for this purpose which sequentially carries out each step taking input from the previous process and giving output to its next. And in the end evaluation scores for the selected feature set are displayed.

Overall Architecture of the System:-



RESULT:-

Accuracy:-

Initially we used accuracy as an evaluation metric, the result for different feature models is as follows:

2nd Deliverable:

- Feature induction true weights some dense defaultlabel 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 somedense : 0.878132387707
- Feature induction true weights sparse : 0.878132387707

3rd Deliverable:

- Metamap+Pos+extra_features -- 0.890009492169
- Metamap+extra_features -- 0.892026578073
- Metamap+Pos -- 0.887517797817
- Pos -- 0.889890840057
- Metamap -- 0.887449964681
- Pos+extra_features -- 0.893925011865
- Lemma+Pos+extra_features -- 0.891077361177
- Lemma+Pos -- 0.892382534409
- Lemma+Metamap+Pos -- 0.885975320361
- Lemma+Metamap -- 0.888348362601
- Lemma+Metamap+extra_features -- 0.889772187945
- Lemma+Metamap+extra_features+Pos -- 0.889060275273
- Lemma+Metamap+extra_features+Pos+Ortho -- 0.895704793545
- Metamap+extra_features+Pos+Ortho -- 0.894280968201
- extra_features+Pos+Ortho -- 0.896298054105
- extra_features+Ortho -- 0.896416706217
- word-length+Ortho+first-letter -- 0.895942097769
- 2-4 length prefixes and 1-4 length suffixes + next-verb and next-noun -- 0.897721879449
- Only Ortho -- 0.897959183673
- Metamap+extra_features+Pos+Ortho+Cluster -- 0.896750647516

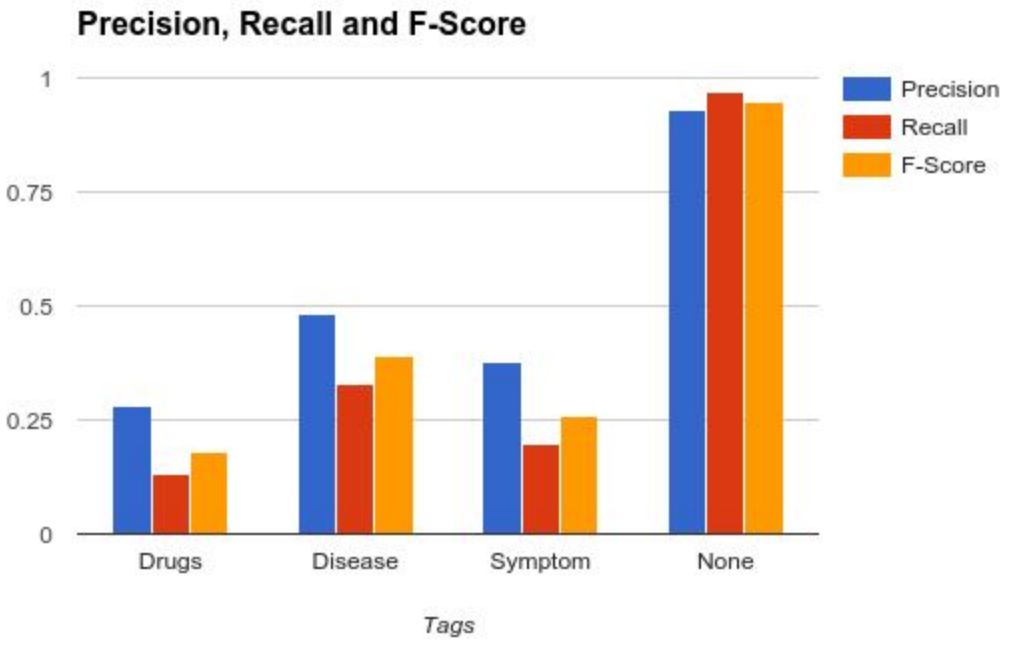
The dataset provided to us consists of 85% of ‘None’ entity tags, hence if we label all the terms as none - we’ll have 85% accuracy. This conveys that accuracy is not the correct metric for correct evaluation of feature models.

Since this metric isn't good-enough to tell us significance of one feature model over another, we tried using more application specific metrics such as Precision, Recall and F-Score.

Precision, Recall and F-Score for different features :-

1. Feature file using <Term, Meta-tag, PoS-tag, Ortho-tag, word-length, prefix-4, suffix-4, clust-id, next-N, next-V, left, Met, PoS, ….. ,Final-Tag>

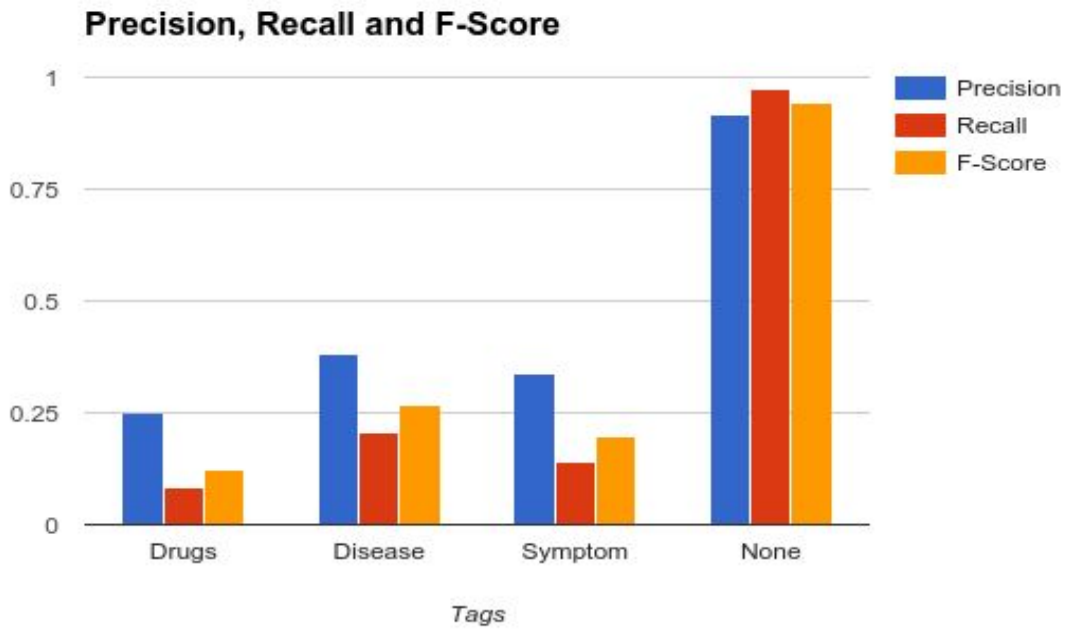
Tags	Precision	Recall	F-Score
Drugs	0.2807017544	0.132231405	0.1797752809
Disease	0.4819277108	0.3311258278	0.3925417076
Symptom	0.3780487805	0.1962025316	0.2583333333
None	0.9282115869	0.9683353042	0.9478490129



- Accuracy =7617/8494=0.896750647516

2. Term, Term-Context

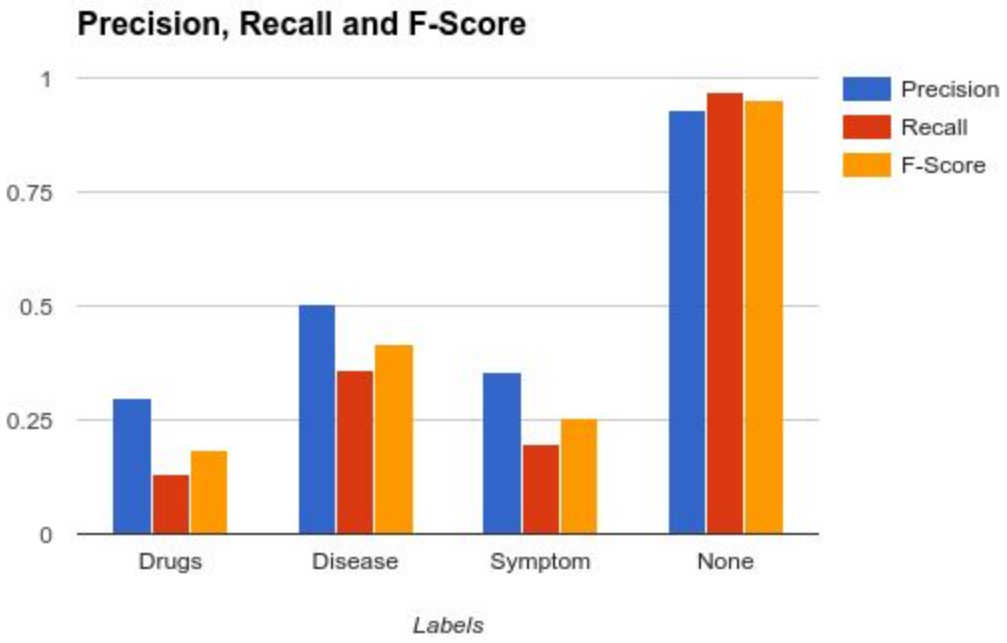
Tags	Precision	Recall	F-Score
Drugs	0.25	0.0826446280992	0.124223602484
Disease	0.382445141066	0.204697986577	0.266666666667
Symptom	0.338461538462	0.139240506329	0.19730941704
None	0.918540729635	0.973388057725	0.945169377129



- Accuracy =7506/8428=0.890602752729

3. Term, Meta-tag, PoS-tag, Ortho-tag, word-length, prefix-4, suffix-4, clust-id, next-N, next-V, left, Met, PoS, ….. ,Label

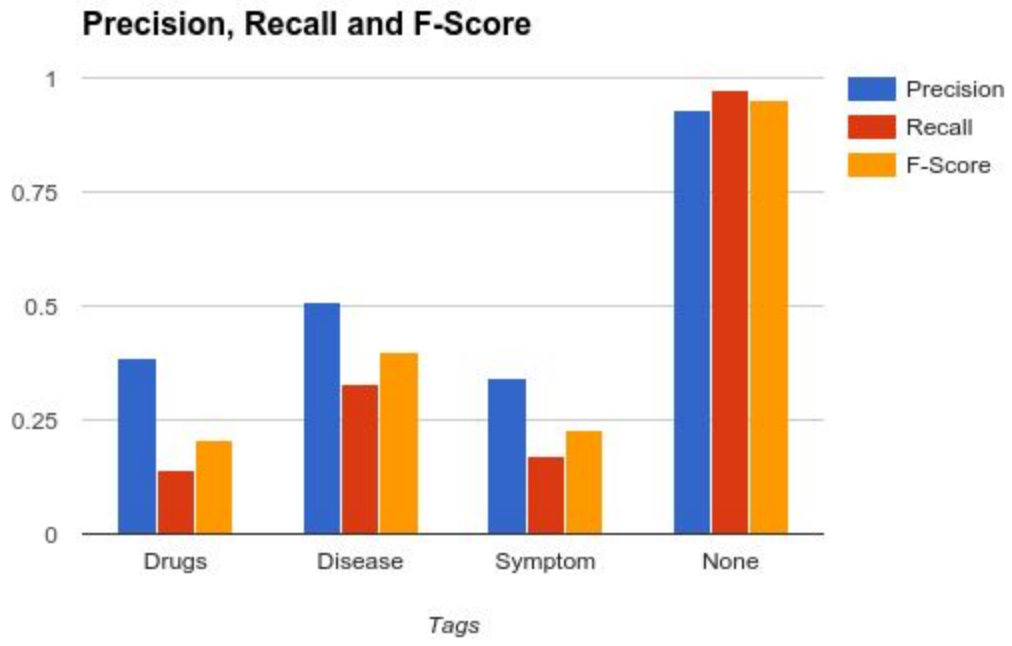
Tags	Precision	Recall	F-Score
Drugs	0.296296296296	0.132231404959	0.182857142857
Disease	0.503496503497	0.35761589404	0.418199419167
Symptom	0.35632183908	0.196202531646	0.25306122449
None	0.93109540636	0.9693864144	0.949855165755



- Accuracy =7641/8494=0.899576171415

4. Term, PoS-tag, Ortho-tag, word-length, prefix-4, suffix-4, clust-id, next-N, next-V, left, PoS, ….. ,Final-Tag

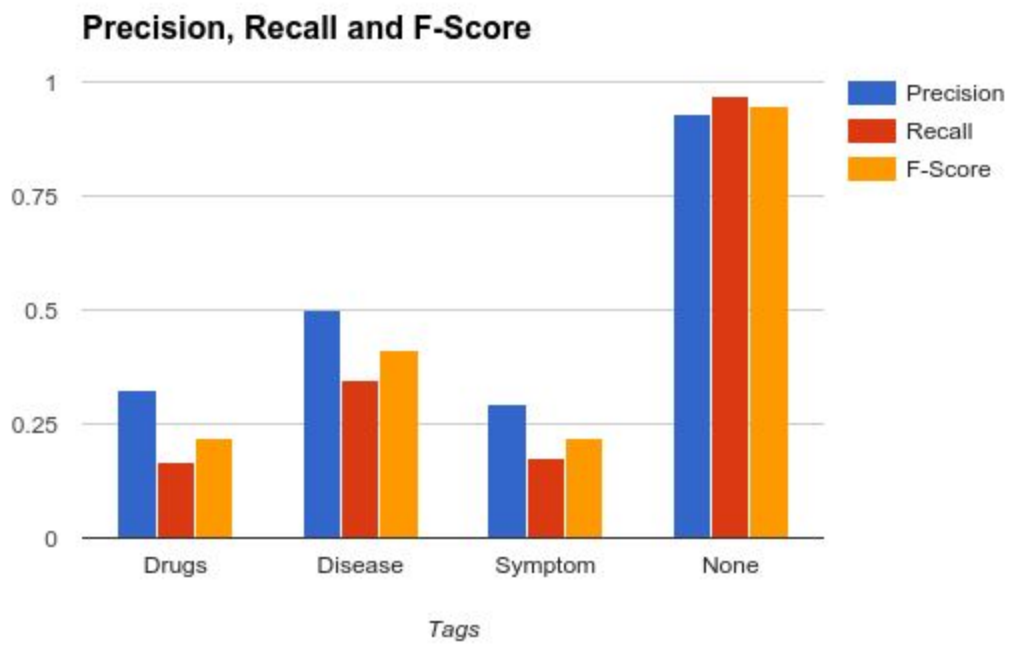
Tags	Precision	Recall	F-Score
Drugs	0.386363636364	0.140495867769	0.206060606061
Disease	0.507614213198	0.331125827815	0.400801603206
Symptom	0.341772151899	0.170886075949	0.227848101266
None	0.928795286449	0.973459466562	0.95060302797



- Accuracy =7653/8494=0.900988933365

5. Term, Meta-tag, Ortho-tag, word-length, prefix-4, suffix-4, clust-id, next-N, next-V, left, Met, ….. ,Final-Tag

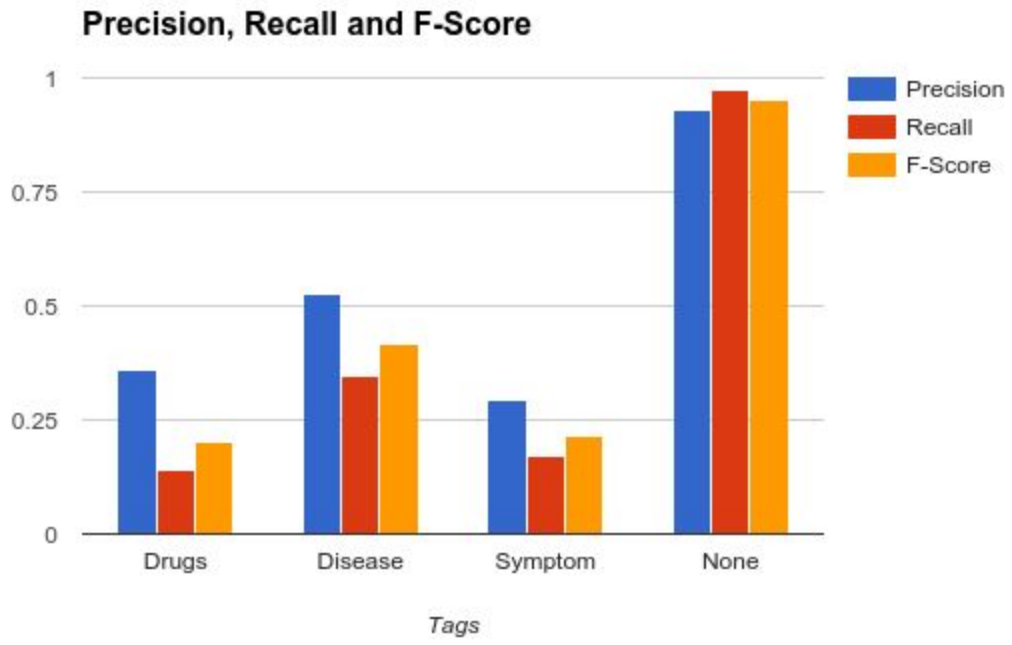
Tags	Precision	Recall	F-Score
Drugs	0.322580645161	0.165289256198	0.218579234973
Disease	0.501193317422	0.347682119205	0.410557184751
Symptom	0.291666666667	0.177215189873	0.220472440945
None	0.93116079323	0.968598081724	0.949510561566



- Accuracy =7630/8494=0.898281139628

6. Term, Ortho, Term-Context, Clust-id

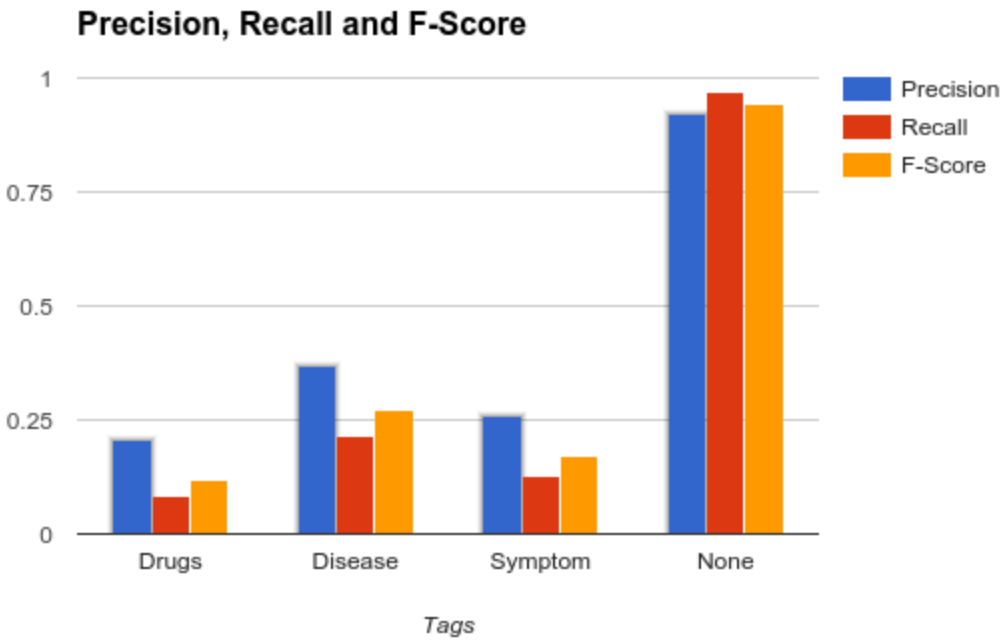
Tags	Precision	Recall	F-Score
Drugs	0.36170212766	0.140495867769	0.202380952381
Disease	0.525	0.347682119205	0.418326693227
Symptom	0.29347826087	0.170886075949	0.216
None	0.930861093652	0.972933911444	0.951432609534



Accuracy =7659/8494=0.90169531434

7. Term, Meta, Term-Context+Meta, Clust-id

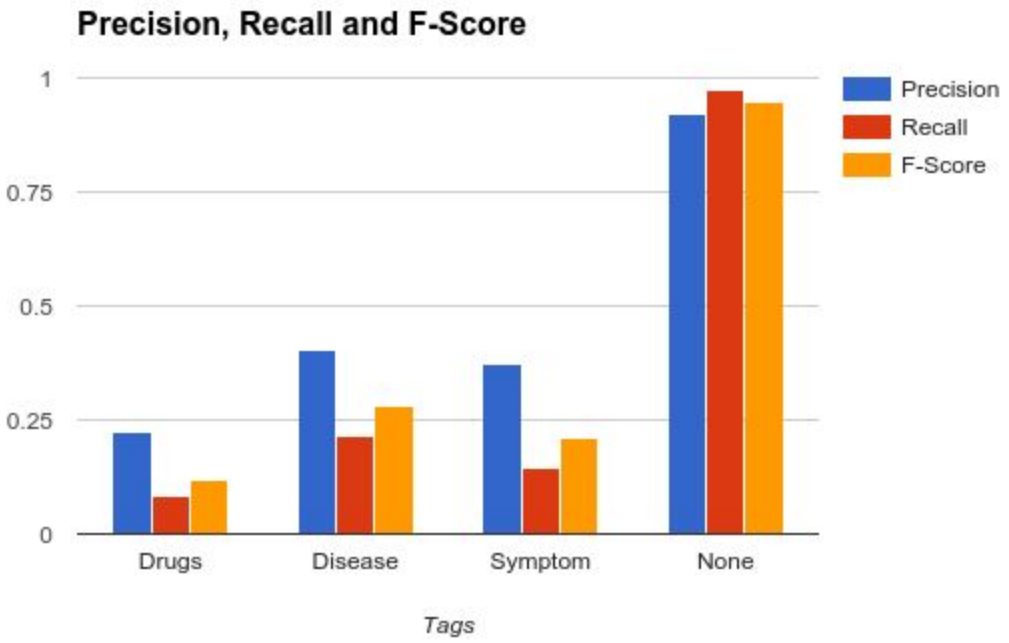
Tags	Precision	Recall	F-Score
Drugs	0.208333333333	0.0826446280992	0.118343195266
Disease	0.370056497175	0..216887417219	0.273486430063
Symptom	0.25974025974	0.126582278481	0.170212765957
None	0.920024953213	0.968860859283	0.943811596058



- Accuracy =7535/8494=0.887096774194

8. Term, POS, Term-Context+POS, Clust-id

Tags	Precision	Recall	F-Score
Drugs	0.222222222222	0.0826446280992	0.120481927711
Disease	0.403726708075	0.215231788079	0.280777537797
Symptom	0.370967741935	0.145569620253	0.209090909091
None	0.91952882827	0.974379188017	0.946159734626



- Accuracy =7579/8494=0.892276901342

Output format:-

We have used the BIO(Begin-Inside-Outside) model for labelling entities. For a whole corpus of tweets, we are labelling data term-wise. This model (BIO) lets us handle entities that span over multiple terms.
For instance, the words “childhood asthma” represent a single entity which is a disease,
In accordance with our implementation, the separate words childhood and asthma will be labelled as childhood -> Disease-Begin and asthma -> Disease-Inside.
So, this is the output format and the corresponding labels are as follows:

{Disease-begin, Disease-inside, Drug-begin, Drug-inside, Symptom-begin, Symptom-inside, None}
So, we basically assign 2 labels (Begin, Inside) per entity (Disease, Drug, Symptom) and 1 extra label (None) representing Outside tag in BIO model.

These labels are the output which are obtained in system generated file by the trained tool.

References Used :-

- Medical Entity Recognition: A comparison of semantic and statistical methods http://www.aclweb.org/anthology/W11-0207
- Enhancing clinical concept extraction with distributional semantics http://ac.els-cdn.com/S1532046411001730/1-s2.0-S1532046411001730-main.pdf?_tid=0ee50780-0281-11e6-944d-00000aacb360&acdnat=1460666580_8a4179d1464230d82e1e8907cb09ad9b
- Using Twitter Data and Sentiment Analysis to Study Diseases Dynamics http://link.springer.com/chapter/10.1007%2F978-3-319-22741-2_2#page-1

References/Documentation used for tools:-:

- Mallet: http://mallet.cs.umass.edu/
- Metamap: https://metamap.nlm.nih.gov/JavaApi.shtml
- Tweet-NLP: http://www.cs.cmu.edu/~ark/TweetNLP/
- Brown-Clustering: https://github.com/percyliang/brown-cluster

Links To Code and Video :-

- Git hub Webpage :- http://kalpishs.github.io/IRE--Medical_NER_Twitter
- Git code :- https://github.com/kalpishs/IRE--Medical_NER_Twitter
- Dropbox link :- https://goo.gl/3Plc8s
- YouTube:- https://youtu.be/dFKIy7CgMrg
- Presentation :- https://goo.gl/GN0AWr