A MINI PROJECT REPORT

On

NER USING CRF ON CLINICAL PRESCRIPTION

Submitted in partial fulfillment of the requirement of
University of Mumbai for the Course

Natural Language Processing
In
Computer Engineering (VIII SEM)

Submitted By
G. Saiprasad
Sunil Pillai
Shweta Shekhar

Subject Incharge
Prof.Pranita Mahajan

Department Of Computer Engineering

SIES Graduate School of Technology

Nerul – 400706

UNIVERSITY OF MUMBAI

Academic Year 2019 – 20

CERTIFICATE

This is to certify that the requirements for the project report entitled '**NER using CRF on a clinical prescription**' have been successfully completed by the following students:

Name	**Roll No.**
G. Saiprasad	114A1012
Sunil Pillai	116A1058
Shweta Shekhar	116A1072

in partial fulfillment of the course Natural Language Processing in Computer Engineering (VIII SEM) of Mumbai University in the Department of Computer Engineering, SIES Graduate School of Technology, Nerul – 400706, during the Academic Year 2019 – 20.

Prof. Pranita Mahajan

Subject Incharge

Department of Computer Engineering
SIES Graduate School of Technology
Nerul – 400706

PROJECT APPROVAL

This project entitled “NER using CRF on clinical prescription ” by Student 1 Name, Student 2 Name, and Student 3 Name are approved for the course Natural Language Processing in Computer Engineering (VIII sem) of Mumbai University in the Department of Computer Engineering.

Examiners:

1. _____________

2. _____________

Subject Incharge:

1. _____________

Date:

Place: Nerul

Department of Computer Engineering
SIES Graduate School of Technology
Nerul – 400706

DECLARATION

We declare that this written submission for Natural Language Processing mini project entitled “NER using CRF on Clinical Prescription” represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any ideas / data / fact / source in our submission. We understand that any violation of the above will cause disciplinary action by the institute and also evoke penal action from the sources which have not been properly cited or from whom prior permission has not been taken when needed.

Project Group Members:

G. Saiprasad & Sign: ________________

Sunil Pillai & Sign: ________________

Shweta Shekhar & Sign: ________________

Date:

Place: Nerul

Table of Contents

Abstract

Named Entity Recognition and Classification (NERC) is a process of recognizing information units like names, including person, organization and location names, and numeric expressions including time, date, money and percent expressions from unstructured text. The goal is to develop practical and domain-independent techniques in order to detect named entities with high accuracy automatically.

The data here is a clinical prescription. It is a feature engineered corpus annotated with POS tags and every word is tagged with appropriate entities on which CRFs are applied which is often used for labeling or parsing of sequential data, such as natural language processing and CRFs find applications in POS Tagging, Named Entity Recognition, among others. Hence, by training CRF model for named entity recognition on our dataset, we are showing the topmost features that determine whether a word is belonging to that particular entity or not and how the tagging of the word either is affected or enhanced by those features useful for determination.

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Fundamentals

Named Entity Recognition (NER) can be performed by various methods but amongst all of them Conditional Random Fields (CRF) has to proven to be the most effective method. Conditional Random Fields are a type of Discriminative classifier, and as such, they model the decision boundary between the different classes.

$$\hat{y} = argmax_y P(y|x)$$

Fig 1.1: Conditional Distribution

In CRFs, our input data is sequential, and we have to take previous context into account when making predictions on a data point. To model this behavior, we will use Feature Functions, that will have multiple input values, which are going to be:

1. The set of input vectors, X
2. The position i of the data point we are predicting
3. The label of data point i-1 in X
4. The label of data point i in X

$$f(X, i, l_{i-1}, l_i)$$

Fig 1.2: Feature Function

The purpose of the feature function is to express some kind of characteristic of the sequence that the data point represents. Each feature function is based on the label of the previous word and the current word, and is either a 0 or a 1. To build the conditional field, we next assign each feature function a set of weights (lambda values), which the algorithm is going to learn.

$$P(y, X, \lambda) = \frac{1}{Z(X)} exp\{\sum_{i=1}^{n} \sum_{j} \lambda_j f_i(X, i, y_{i-1}, y_i)\}$$

$$\text{Where: } Z(x) = \sum_{y' \in y} \sum_{i=1}^{n} \sum_{j} \lambda_j f_i(X, i, y'_{i-1}, y'_i)$$

Fig 1.3: Probability Distribution

1.2 Objectives

The objective of this work is as follows:

1. To study the Named Entity Recognition systems which would help with the approach for creating one for clinical prescription that could also overcome drawbacks of existing methods if any.
2. To understand the method of tagging an entity that would help in its detection with high accuracy.
3. To identify evaluation metrics used for performance analysis of different Named Entity Recognition systems.

1.3 Scope

This scale of this project is of clinic level prescriptions obtained from a doctor under normal circumstances which usually mention the medicines along with their dosages and duration upto which to be consumed. This system can understand and process the existing clinical prescriptions and is not scaled up to hospital reports.

1.4 Organization of the Report

The report is organized as follows:

The introduction is given in Chapter 1. It describes the fundamental terms used in this project. It motivates to study and understand the different techniques used in this work. This chapter also presents the outline of the objective of the report. The Chapter 2 describes the review of the relevant various techniques in the literature systems. It describes the pros and cons of each technique. The Chapter 3 presents the Theory and proposed work. It describes the major approaches used in this work. The societal and technical applications are mentioned in Chapter 4. The summary of the report is presented in Chapter 5.

Chapter 2
Literature Survey

2.1 Introduction

The topic along with the algorithms and the techniques that we have chosen is studied and researched thoroughly by referring to the research papers listed down below.

2.2 Summary of Literature Survey

Table 2.1: Summary of Research Papers

SN	Techniques	Author & Year of Publication	Advantages and Disadvantages
1.	Machine learning based biomedical named entity recognition	PaIET Chennai Fourth International Conference on Sustainable Energy and Intelligent Systems (SEISCON 2013)	Advantages: The current trends of Machine Learning based recognition of Named Entities for Biomedical Document was analysed and a comparative analysis using 4 systems. Disadvantages: Entities extracted from biomedical documents can't be used to identify the multiple (coreferring) mentions of the same entity in the text identified. And the associations between the entities and Event Extraction which the current system cant do .
2.	Design Challenges and Misconceptions in Named Entity Recognition* †	Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL), Boulder, Colorado, June 2009.	Advantages: Presented a simple model for NER that uses expressive features to achieve new state of the art performance on the Named Entity recognition task. Consistent performance gains across several domains, most interestingly in Webpages, where the named entities had less context and were different in nature from the named entities in the training set Disadvantages: This model doesn't cater to the medical domain or specific domain in general.

3.	Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data	Proceedings of the 18th International Conference on Machine Learning 2001 (ICML 2001),	Advantages: Able to present iterative parameter estimation algorithms for conditional random fields and provides an alternative approach to HMMs and MEMMs on synthetic and natural-language data. Disadvantage: Their main current limitation is the slow convergence of the training algorithm relative to MEMMs, let alone to HMMs, for which training on fully observed data is very efficient.
4.	Biomedical Named Entities Recognition Using Conditional Random Fields Model	Conference: Fuzzy Systems and Knowledge Discovery, Third International Conference, FSKD 2006, Xi'an, China, September 24-28, 2006, Proceedings	Advantages: Shallow syntactic features greatly improve the model's performance. ¾ Show the effect of POS features and shallow syntactic features in detail; conclude that large granule linguistic knowledge can prompt the CRFs model's generalization, which can afford valuable reference to other researchers. ¾ Achieve a biomedical NRE system with an F-measure of 71.2% in JNLPBA test data and which is better than most state-of-the-art systems. The system has strong adaptability because it does not need any dictionary resources and post-processing. Disadvantage: Biomedical NER is a challenging problem. There are many different aspects to deal with. In general, biomedical NEs do not follow any nomenclature and can com-prise long compound words and short abbreviations.
5.	On the limited memory BFGS method for large scale optimization	Anand Shanker Tewari, Abhay Kumar, and Asim Gopal Barman. 2014	Advantages: Tests indicate that a simple implementation of L-BFGS method performs better than code of buckley and LeNir(1985) Disadvantage: In case of large problems with expensive function CG is competitive with L-BFGS

Chapter 3
Implementation Details

3.1 Overview

The implementation has been carried out using Conditional Random Fields (CRFs) method and the implementation of this Natural Language Processing project has been carried out in Python Language since it is convenient for multiple tasks such as data pre-processing, Machine Learning and Analysis purpose and has very few compatibility issues. Also Jupyter Notebook has been used as the IDE for coding as it provides easy access and convenient for code snippets.

3.1.1 Existing Methodology and Systems

There are a lot of systems and research work for NER in various domains such as Banking, Biomedical etc. but on the basis of our research, we found out that there is no NER system specifically for clinical prescriptions and with a good accuracy. One of the existing Systems involves NER using CRF on the popular GMB(Groningen Meaning Bank) corpus (CoNLL 2002) which is annotated with Part Of Speech (POS) and Inside Outside Beginning (IOB) tags. This annotated corpus has sentence wise list of words with POS and IOB tag for each. The IOB tagging system has suffixes before for e.g. ‘b-org’ or ‘i-org’ where ‘b’ tells whether the corresponding word is the beginning of a chunk and ‘i’ tells whether the word is inside the chunk. This corpus similar to ours has a set to tags defined like ‘org’ above which is short for organization. The system tags the words from sentences into one of their entities and this system is solely for a banking domain. Also, there are other methodologies used to perform NER tagging such as Perceptron, Naive-Bayes, SGD (Stochastic Gradient Descent), Passive-Aggressive Classifiers but none of them yield the results like CRFs. CRF method has proven to be better than all of these in terms of evaluation metrics such as Accuracy, Precision, F-Score and Recall values.

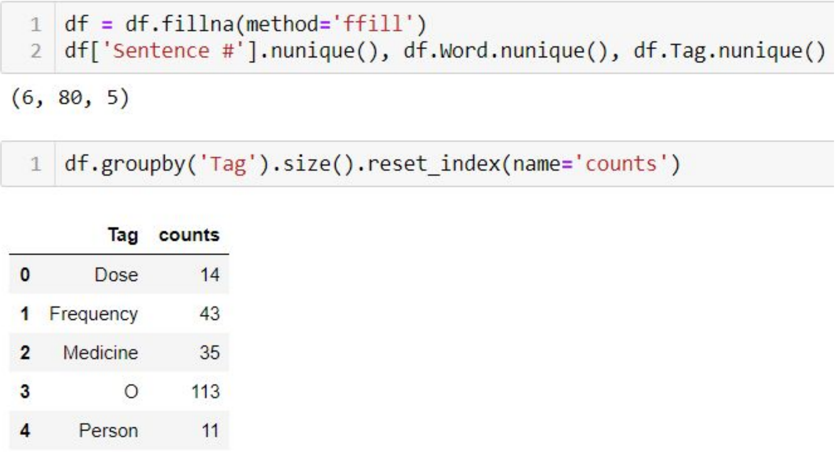
3.1.2 Proposed Methodology and System

Our proposed system on the basis of the conclusion derived from the existing systems as mentioned above is use of CRF. We are using the ‘L-BFGS’ (Limited-memory Broyden–Fletcher–Goldfarb–Shanno algorithm (BFGS)) algorithm to build the CRF model. This is an optimization algorithm as in the family of quasi-Newton methods which uses limited memory and is a popular algorithm for parameter estimation in machine learning. We train the model on a set of features (X) involving the current and neighbouring words and their POS tags. Their labels are fed into the other dataframe (Y). An 80:20 ratio is maintained for splitting the data into training and test sets. We are able to achieve considerable scores in evaluation and also can predict the likely transition of entities and weights of the features for prediction of an entity.

3.2 Implementation Details

Implementation steps are as follows:

1. Import the required packages and load the dataset from the working directory.
2. Preprocess the data i.e. check and remove 'null' values, format the input data and also check the tag distribution as shown in Figure 3.1.



```
df = df.fillna(method='ffill')
df['Sentence #'].nunique(), df.Word.nunique(), df.Tag.nunique()
```

```
(6, 80, 5)
```

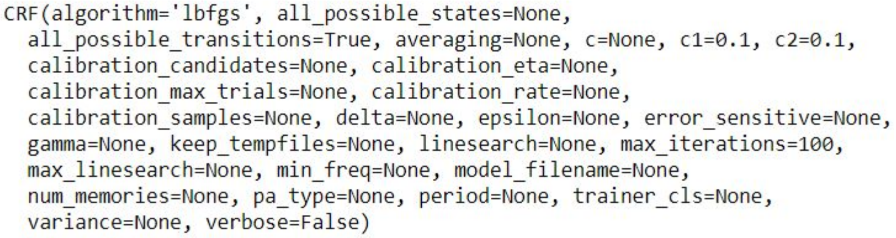
```
df.groupby('Tag').size().reset_index(name='counts')
```

	Tag	counts
0	Dose	14
1	Frequency	43
2	Medicine	35
3	O	113
4	Person	11

Fig 3.1: Preprocessing

3. Retrieve sentences with their POS and tags and create tuples in the format (word, POS, tag).
4. Next, we extract more features such as bias, word parts, simplified POS tags, lower/title/upper flags, features of nearby words etc. and convert them to required format i.e. each sentence should be converted to a list of dicts. Special features of next word (if the current word is not the last word) or of the previous word (if the current word is the last word) are also considered which includes POS tag, it's first 2 characters, lower/title/upper flags and 'BOS' if first word or 'EOS' for last word of the sentence.
5. Split train and test sets as mentioned in the above section.

6. Train a CRF model with ‘lbfgs’ algorithm. Here, c1 & c2 are coefficients for L1 and L2 Regularization which are used for over-fitting and feature selection respectively and the maximum number of iterations for optimization algorithms is assigned as 100. All the parameters of the model are shown in below Figure 3.2.



```
CRF(algorithm='lbfgs', all_possible_states=None,
  all_possible_transitions=True, averaging=None, c=None, c1=0.1, c2=0.1,
  calibration_candidates=None, calibration_eta=None,
  calibration_max_trials=None, calibration_rate=None,
  calibration_samples=None, delta=None, epsilon=None, error_sensitive=None,
  gamma=None, keep_tempfiles=None, linesearch=None, max_iterations=100,
  max_linesearch=None, min_freq=None, model_filename=None,
  num_memories=None, pa_type=None, period=None, trainer_cls=None,
  variance=None, verbose=False)
```

Fig 3.2: Model Parameters

7. Create a copy of classes without ‘O’ class in it since it is not needed to be learned and detected.
8. Now comes Prediction of scores and Evaluation of the model in terms of learning.

3.2.1 Details of Packages and Data set

The corpus consists of the labels Sentence No., Word, POS and tag. We have identified 5 tags for the clinical prescription that are Person, Medicine, Dose, Frequency and Others (O). As the names suggest these tags tell that those tagged words are the particular entities or components of a sentence. ‘O’ represent all the words which are part of the sentences in the corpus but do not fall under the remaining tags. The dataset is annotated with the POS and the above mentioned tags.

Packages used in this system are :

1. ‘pandas’ - file handling and preprocessing
2. ‘numpy’ - vector operations
3. ‘DictVectorizer’ from ‘sklearn.feature_extraction’ - formatting of data and extraction.
4. ‘train_test_split’ from ‘sklearn.model_selection’ - splitting data for training and testing.
5. ‘classification_report’ and ‘flat_accuracy_score’ from ‘sklearn.metrics’ - evaluation
6. ‘sklearn_crfsuite’ - CRF model in Scikit-Learn Package that generates transition features that associate all of the possible Tags in the dataset.
7. ‘scorers’ and ‘metrics’ from ‘sklearn_crfsuite’ - CRF model evaluation
8. ‘Counter’ from ‘collections’ - aggregation
9. eli5 - Visual representation of analysis.

Chapter 4

Project Inputs and Outputs

4.1 Input Details

The input is a file named ‘NLP_dataset.csv’ which is the dataset described in the dataset section. The file is loaded through ‘pandas’ command ‘pd.read_csv’.

4.2 Evaluation Parameters Details

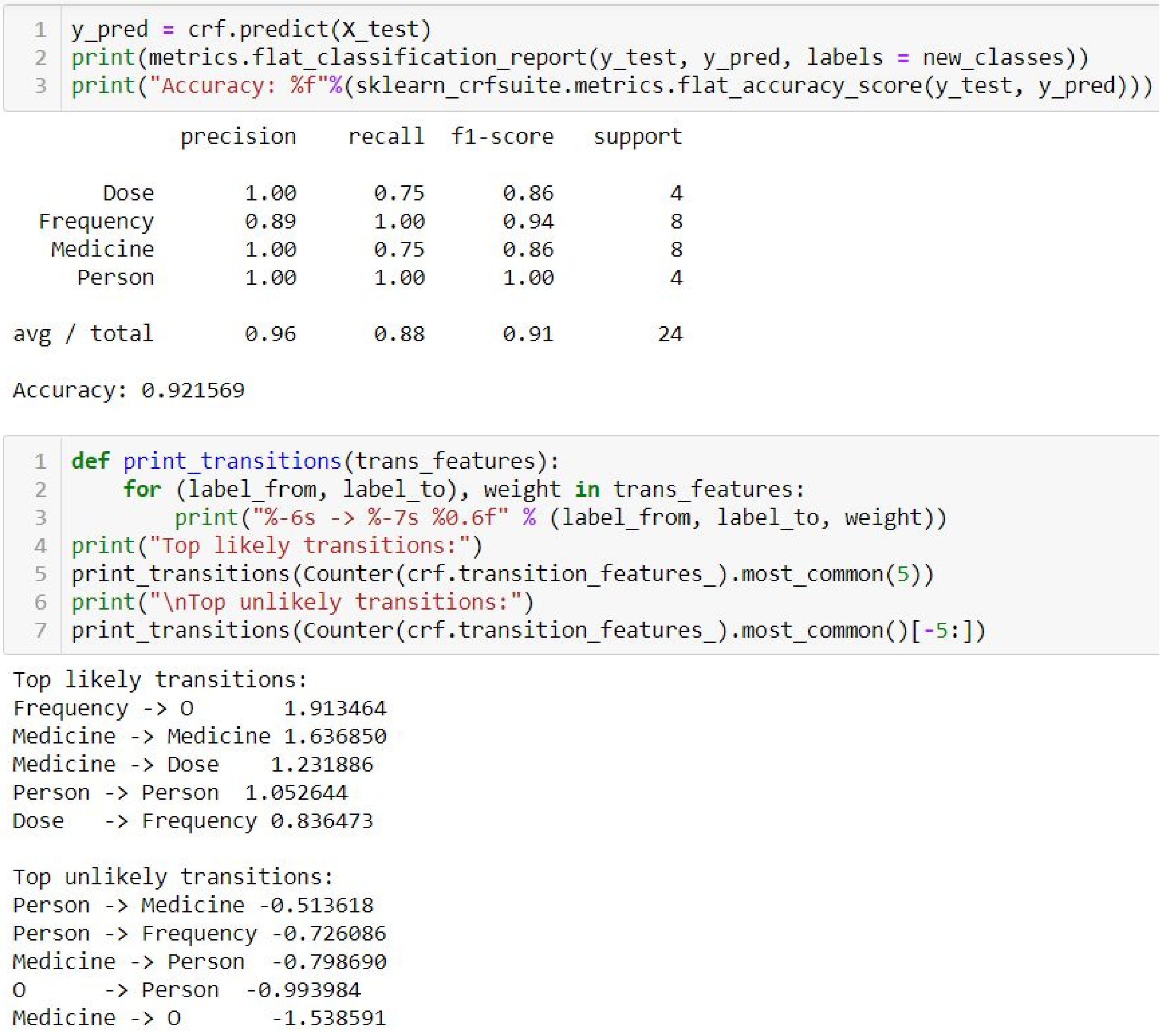
Evaluation Metrics are :

1. Accuracy - It is the ratio of correctly predicted observation to the total observations. It is given as: (TP+TN) / (TP+FP+FN+TN)
2. Precision - It is the ratio of correctly predicted positive observations to the total predicted positive observations. It is given as: (TP) / (TP+FP)
3. Recall (Sensitivity) - Recall is the ratio of correctly predicted positive observations to all the observations in actual class - yes. It is given as: (TP) / (TP+FN)
4. F1 score - F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. It is given as: 2*(Recall * Precision) / (Recall + Precision)
5. Support - It is the support is the number of occurrences of each class in y_test.
6. Weights - Contribution of a feature for the word to be tagged as an entity.
7. Transition Likelihood - Probability percentage of word of one entity coming after another.

4.3 Output Details and Screenshots

We have gained the following outputs as a result of the evaluation of the model and it’s analysis.

1. The below Figure 4.1 shows the precision, recall, f1-score, support and accuracy of our trained CRF model, this is followed by the top 5 and least 5 likely transitions of entities in the sentences as per the dataset.



```
y_pred = crf.predict(X_test)
print(metrics.flat_classification_report(y_test, y_pred, labels = new_classes))
print("Accuracy: %f"%(sklearn_crfsuite.metrics.flat_accuracy_score(y_test, y_pred)))
```

```
             precision    recall  f1-score   support

       Dose       1.00      0.75      0.86         4
  Frequency       0.89      1.00      0.94         8
   Medicine       1.00      0.75      0.86         8
     Person       1.00      1.00      1.00         4

avg / total       0.96      0.88      0.91        24

Accuracy: 0.921569
```

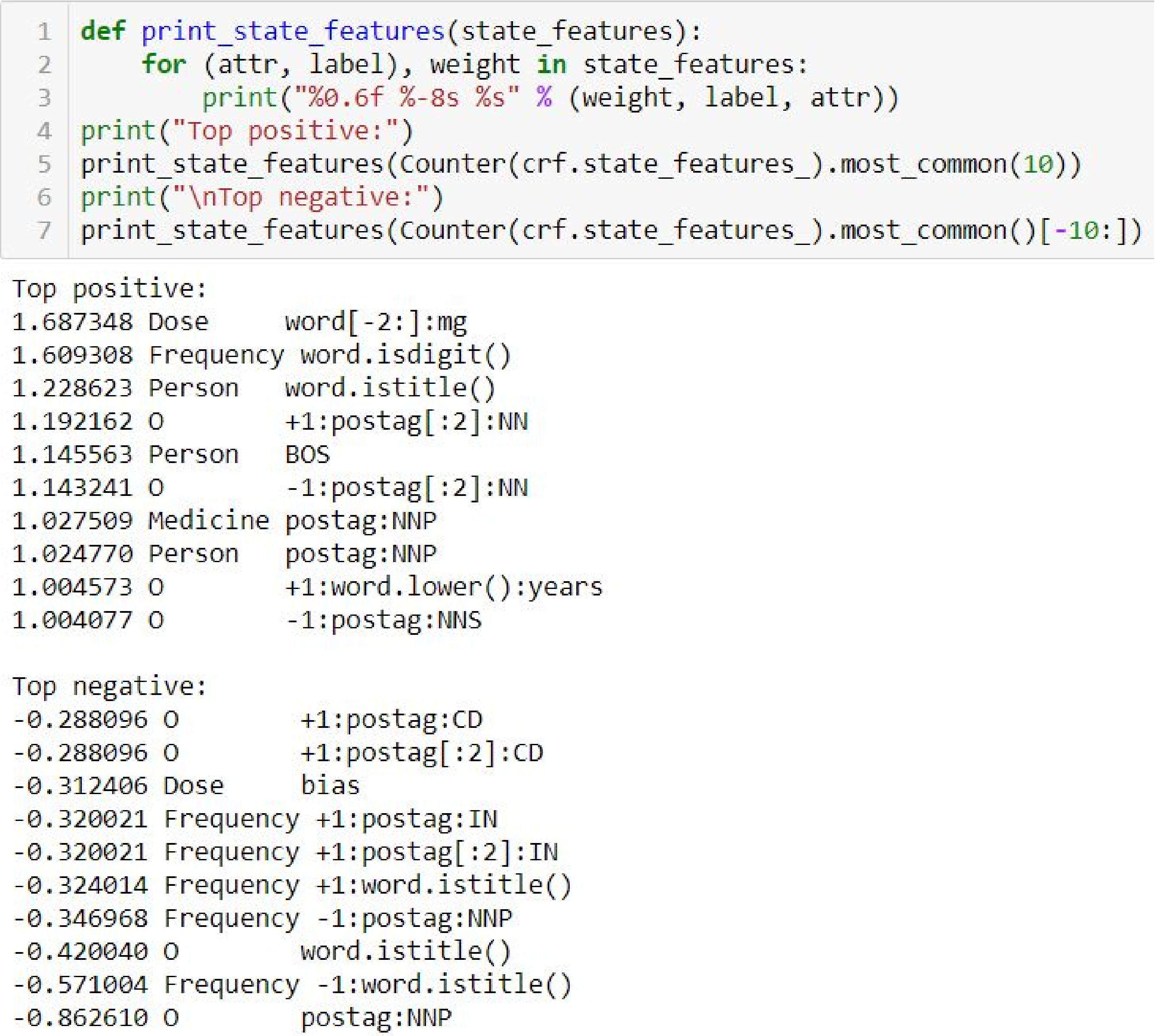
```
def print_transitions(trans_features):
    for (label_from, label_to), weight in trans_features:
        print("%-6s -> %-7s %0.6f" % (label_from, label_to, weight))
print("Top likely transitions:")
print_transitions(Counter(crf.transition_features_).most_common(5))
print("\nTop unlikely transitions:")
print_transitions(Counter(crf.transition_features_).most_common()[-5:])
```

```
Top likely transitions:
Frequency -> O       1.913464
Medicine -> Medicine 1.636850
Medicine -> Dose    1.231886
Person -> Person  1.052644
Dose   -> Frequency 0.836473

Top unlikely transitions:
Person -> Medicine -0.513618
Person -> Frequency -0.726086
Medicine -> Person  -0.798690
O      -> Person  -0.993984
Medicine -> O       -1.538591
```

Fig 4.1 : Evaluation Metrics & Transition Likelihood

2. The Figure 4.2 below displays the top 10 and the least 10 weights of features i.e the features contributing most or least for a particular entity.



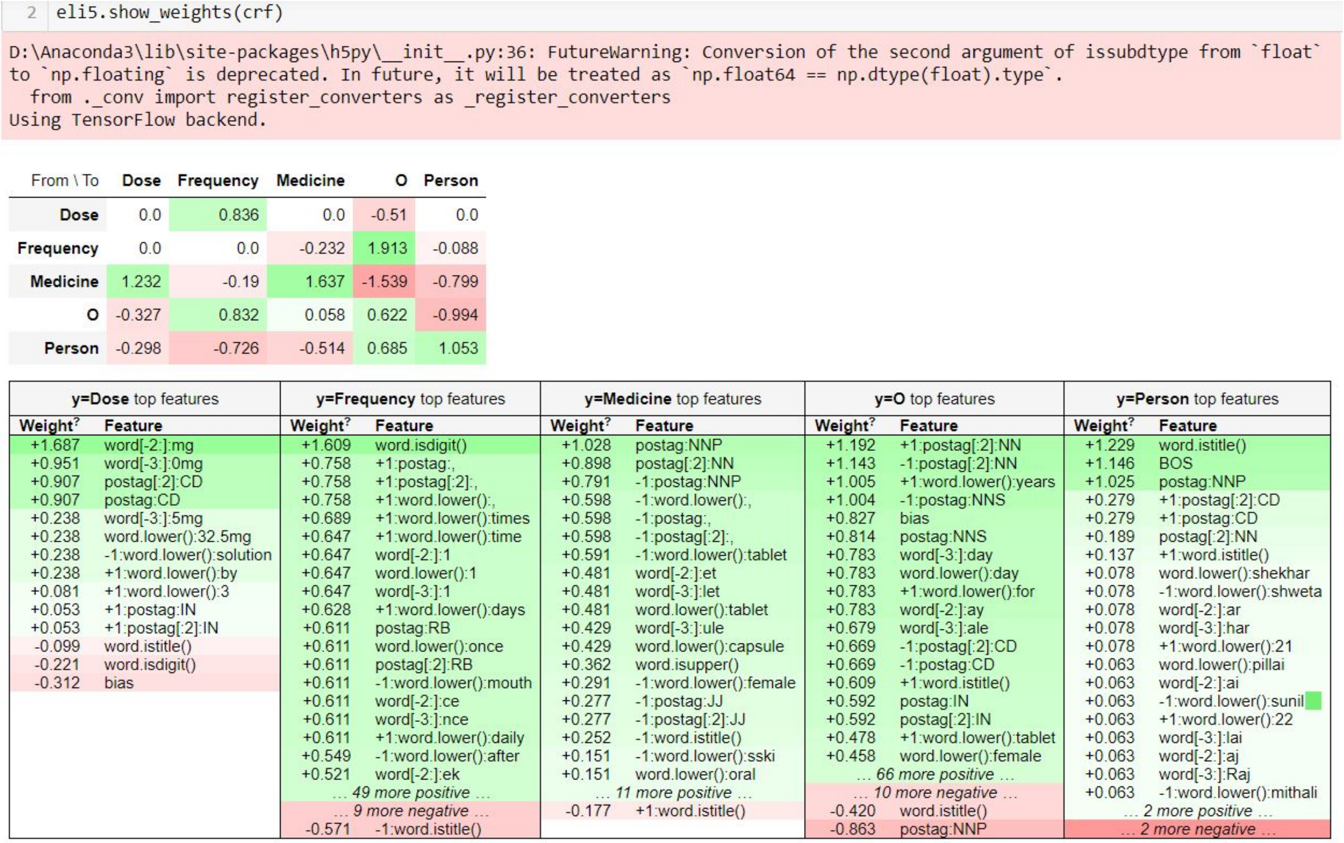
```
def print_state_features(state_features):
    for (attr, label), weight in state_features:
        print("%0.6f %-8s %s" % (weight, label, attr))
print("Top positive:")
print_state_features(Counter(crf.state_features_).most_common(10))
print("\nTop negative:")
print_state_features(Counter(crf.state_features_).most_common()[-10:])
```

```
Top positive:
1.687348 Dose     word[-2:]:mg
1.609308 Frequency word.isdigit()
1.228623 Person   word.istitle()
1.192162 O        +1:postag[:2]:NN
1.145563 Person   BOS
1.143241 O        -1:postag[:2]:NN
1.027509 Medicine postag:NNP
1.024770 Person   postag:NNP
1.004573 O        +1:word.lower():years
1.004077 O        -1:postag:NNS

Top negative:
-0.288096 O        +1:postag:CD
-0.288096 O        +1:postag[:2]:CD
-0.312406 Dose     bias
-0.320021 Frequency +1:postag:IN
-0.320021 Frequency +1:postag[:2]:IN
-0.324014 Frequency +1:word.istitle()
-0.346968 Frequency -1:postag:NNP
-0.420040 O        word.istitle()
-0.571004 Frequency -1:word.istitle()
-0.862610 O        postag:NNP
```

Fig 4.2 : Top Feature Weights

3. The Figure 4.3 below shows the matrix representation of transition likelihood and also all the top features’ weight contributions for all the entities.



```
eli5.show_weights(crf)
```

```
D:\Anaconda3\lib\site-packages\h5py\__init__.py:36: FutureWarning: Conversion of the second argument of issubdtype from `float`
to `np.floating` is deprecated. In future, it will be treated as `np.float64 == np.dtype(float).type`.
  from ._conv import register_converters as _register_converters
Using TensorFlow backend.
```

From \ To	Dose	Frequency	Medicine	O	Person
Dose	0.0	0.836	0.0	-0.51	0.0
Frequency	0.0	0.0	-0.232	1.913	-0.088
Medicine	1.232	-0.19	1.637	-1.539	-0.799
O	-0.327	0.832	0.058	0.622	-0.994
Person	-0.298	-0.726	-0.514	0.685	1.053

y=Dose top features

Weight?	Feature
+1.687	word[-2:]:mg
+0.951	word[-3:]:0mg
+0.907	postag[:2]:CD
+0.907	postag:CD
+0.238	word[-3:]:5mg
+0.238	word.lower():32.5mg
+0.238	-1:word.lower():solution
+0.238	+1:word.lower():by
+0.081	+1:word.lower():3
+0.053	+1:postag:IN
+0.053	+1:postag[:2]:IN
-0.099	word.istitle()
-0.221	word.isdigit()
-0.312	bias

y=Frequency top features

Weight?	Feature
+1.609	word.isdigit()
+0.758	+1:postag:,
+0.758	+1:postag[:2]:,
+0.758	+1:word.lower():,
+0.689	+1:word.lower():times
+0.647	+1:word.lower():time
+0.647	word[-2:]:1
+0.647	word.lower():1
+0.647	word[-3:]:1
+0.628	+1:word.lower():days
+0.611	postag:RB
+0.611	word.lower():once
+0.611	postag[:2]:RB
+0.611	-1:word.lower():mouth
+0.611	word[-2:]:ce
+0.611	word[-3:]:nce
+0.611	+1:word.lower():daily
+0.549	-1:word.lower():after
+0.521	word[-2:]:ek
	… *49 more positive* …
	… *9 more negative* …
-0.571	-1:word.istitle()

y=Medicine top features

Weight?	Feature
+1.028	postag:NNP
+0.898	postag[:2]:NN
+0.791	-1:postag:NNP
+0.598	-1:word.lower():,
+0.598	-1:postag:,
+0.598	-1:postag[:2]:,
+0.591	-1:word.lower():tablet
+0.481	word[-2:]:et
+0.481	word[-3:]:let
+0.481	word.lower():tablet
+0.429	word[-3:]:ule
+0.429	word.lower():capsule
+0.362	word.isupper()
+0.291	-1:word.lower():female
+0.277	-1:postag:JJ
+0.277	-1:postag[:2]:JJ
+0.252	-1:word.istitle()
+0.151	-1:word.lower():sski
+0.151	word.lower():oral
	… *11 more positive* …
-0.177	+1:word.istitle()

y=O top features

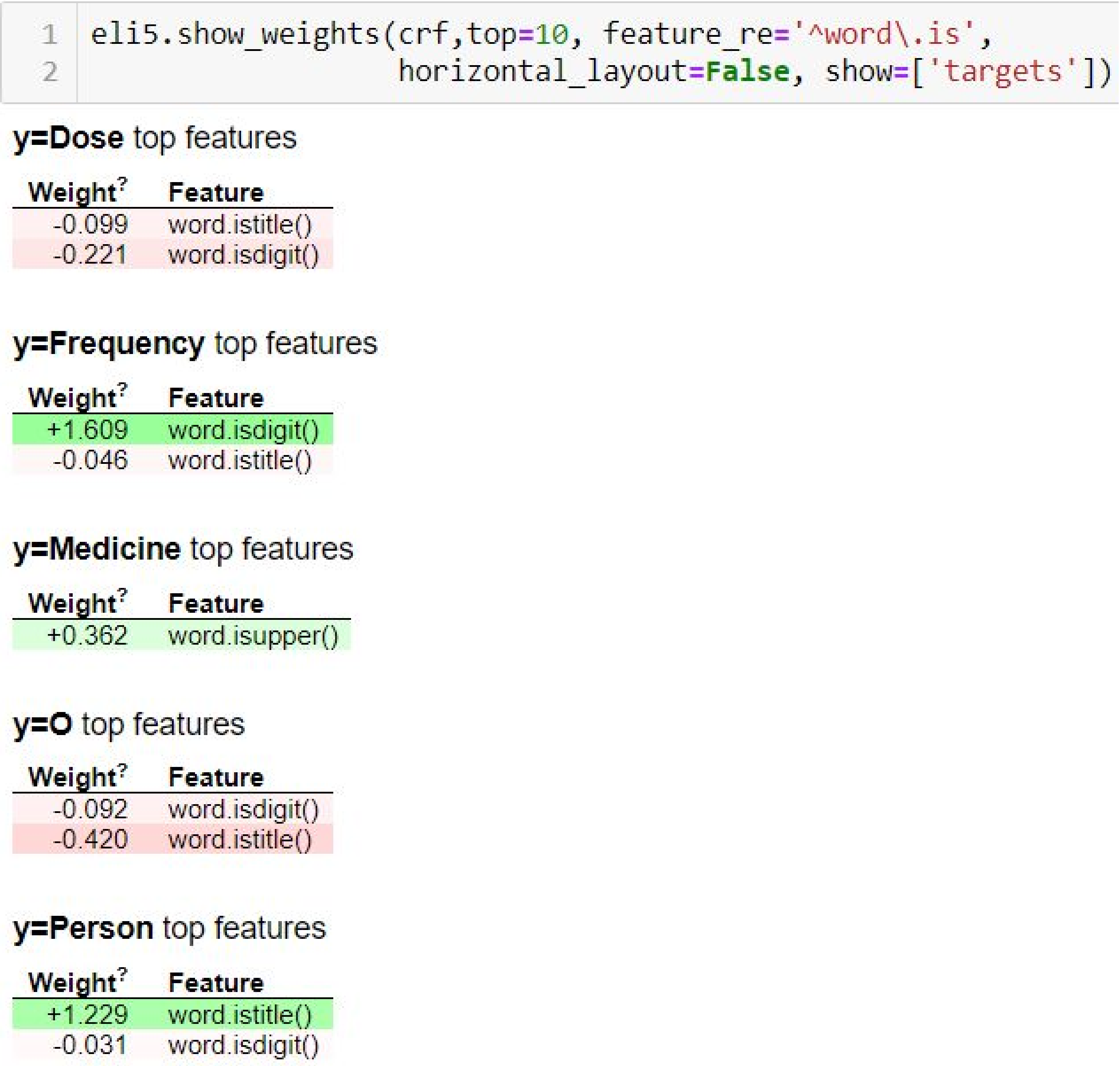
Weight?	Feature
+1.192	+1:postag[:2]:NN
+1.143	-1:postag[:2]:NN
+1.005	+1:word.lower():years
+1.004	-1:postag:NNS
+0.827	bias
+0.814	postag:NNS
+0.783	word[-3:]:day
+0.783	word.lower():day
+0.783	+1:word.lower():for
+0.783	word[-2:]:ay
+0.679	word[-3:]:ale
+0.669	-1:postag[:2]:CD
+0.669	-1:postag:CD
+0.609	+1:word.istitle()
+0.592	postag:IN
+0.592	postag[:2]:IN
+0.478	+1:word.lower():tablet
+0.458	word.lower():female
	… *66 more positive* …
	… *10 more negative* …
-0.420	word.istitle()
-0.863	postag:NNP

y=Person top features

Weight?	Feature
+1.229	word.istitle()
+1.146	BOS
+1.025	postag:NNP
+0.279	+1:postag[:2]:CD
+0.279	+1:postag:CD
+0.189	postag[:2]:NN
+0.137	+1:word.istitle()
+0.078	word.lower():shekhar
+0.078	-1:word.lower():shweta
+0.078	word[-2:]:ar
+0.078	word[-3:]:har
+0.078	+1:word.lower():21
+0.063	word.lower():pillai
+0.063	word[-2:]:ai
+0.063	-1:word.lower():sunil
+0.063	+1:word.lower():22
+0.063	word[-3:]:lai
+0.063	word[-2:]:aj
+0.063	word[-3:]:Raj
+0.063	-1:word.lower():mithali
	… *2 more positive* …
	… *2 more negative* …

Fig 4.3 : Transition Matrix & Feature Distribution

4. The Figure 4.4 below shows that, on an average, every word which is described by/or dependent on the topmost features are tagged into that particular entity. The current word is described as **word.is** using regular expression.



```
eli5.show_weights(crf,top=10, feature_re='^word\.is',
                  horizontal_layout=False, show=['targets'])
```

y=Dose top features

Weight?	Feature
-0.099	word.istitle()
-0.221	word.isdigit()

y=Frequency top features

Weight?	Feature
+1.609	word.isdigit()
-0.046	word.istitle()

y=Medicine top features

Weight?	Feature
+0.362	word.isupper()

y=O top features

Weight?	Feature
-0.092	word.isdigit()
-0.420	word.istitle()

y=Person top features

Weight?	Feature
+1.229	word.istitle()
-0.031	word.isdigit()

Fig 4.4: Final O/p of a tagging of word with topmost features

Chapter 5

Summary and Future Scope

5.1 Summary

This topic NER using CRF on clinical prescription can be summarised as follows NERC is nothing but named entity recognition classification is used for recognising our information units such as person,medicine,dose,frequency and others and detect the named entity with high accuracy of 96.02% and further CRF was used for predicting the sequences that use the contextual information such as(the tags) to add information(features) which will be used by the model to make a correct prediction by making use of L-BFGS(limited BFGS Algo) that uses limited amount of computer memory and is used for parameter estimation to give the desired output sequence.The output sequence is modeled as the normalized product of the feature function.The ELI5 package of python was used to explain weights and transition between each entity and understood the text processing utilities and highlight the text properly with color coding .The importance of color coding shows for every transition whether it has more important or positively determining the transition or whether it has negatively or less important features determining the transition.The final Output of our project mainly involves the top features of a particular word belonging to that entity or not where we are making use of Regular Expression to find out for the current word ie word.is.It tells us the tagging of the words depend on the more or less important feature determining it.

5.2 Future Scope

Currently our existing NER system is able to predict for sentences within a clinical prescription containing the Patient details, medicines along with their doses and frequency for consuming that particular medicine and the rest of the words in the sentence as others with high accuracy. Our future work mainly will involve improving our NER system for prediction of an entire prescription/report instead of sentences using various tools or machine learning algorithms with high accuracy as possible. This could be done vice-versa by providing only the words belonging to the particular tags i.e. (person, medicine, dose ,frequency and remaining as others) and generating the entire prescription from it, there is scope for including more tags to make the prescription generation more accurate, relevant,detailed and easier to understand.

References

[1] Luo Zhenghua, “Realization of Individualized Recommendation System on Books Sale”, IEEE 2012 International Conference on Management of e-Commerce and e-Government, pp.10-13.

[2] Tewari, A.S. Kumar, and Barman, A.G, “Book recommendation system based on combine features of content based filtering, collaborative filtering and association rule mining”, International Advance Computing Conference (IACC), IEEE, pp 500 – 503, April 2014.

[3] N. Kanya and T. Ravi, "Machine learning based biomedical named entity recognition," IET Chennai Fourth International Conference on Sustainable Energy and Intelligent Systems (SEISCON 2013), Chennai, 2013, pp. 380-384.

[4] Ratinov, Lev, and Dan Roth. "Design challenges and misconceptions in named entity recognition." Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009). 2009.

[5] Lafferty, John, Andrew McCallum, and Fernando CN Pereira. "Conditional random fields: Probabilistic models for segmenting and labeling sequence data." (2001).

[6] Sun, Chengjie, et al. "Biomedical named entities recognition using conditional random fields model." International Conference on Fuzzy Systems and Knowledge Discovery. Springer, Berlin, Heidelberg, 2006.

[7] Liu, Dong C., and Jorge Nocedal. "On the limited memory BFGS method for large scale optimization." Mathematical programming 45.1-3 (1989): 503-528.

[8] https://towardsdatascience.com/named-entity-recognition-and-classification-with-scikit-learn-f05372f07ba2

[9] https://en.wikipedia.org/wiki/Limited-memory_BFGS

[10] https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/

[11] https://sklearn-crfsuite.readthedocs.io/en/latest/tutorial.html

[12] https://towardsdatascience.com/beyond-accuracy-precision-and-recall-3da06bea9f6c

Acknowledgement

We would like to express our deepest appreciation to our guide Prof. Pranita Mahajan who provided us the possibility to complete this report and the project titled “**NER using CRF in Clinical Prescription**”. A special gratitude to our guide for helping us out whose contribution in stimulating suggestions and encouragement, helped us to coordinate our project especially in writing this report.

Furthermore we would also like to acknowledge and appreciate the crucial role of our HOD Dr. Aparna Bannore who gave the permission to use all required resources and the necessary materials to complete the task.

A special thanks goes to our I/C Principal Dr. Atul Khemkar who motivated us at all times to complete this report. Last but not the least, we have to appreciate the guidance given by other faculty members and staff in the Computer Engineering department for their constant support. We truly acknowledge their valuable comments and advice.

G. Saiprasad
Sunil Pillai
Shweta Shekhar