Exploring Named Entity Recognition task using Random Forest Classifier, CRF and BiLSTM-CRF Models

CS – 562 Natural Language Processing
Sonam De

Abstract

Sequence tagging and Named Entity recognition is a challenging task in several domains, however recent studies shows great achievements in this area. NER is a subset of information extraction task. In this project I aim to explore and compare the performance of three models - Random Forest Classifier, Conditional Random Forest and Bi- directional LSTM - CRF model on named entity tasks. The experiments include different training procedures (use of oversampling algorithm). For this task, I use GMB (Groningen Meaning-Bank) corpus dataset. This dataset is an open source and is easily available from [1]. This data is highly imbalanced. To overcome the class imbalance, I have used SMOTE - Synthetic Minority Oversampling Technique. The evaluation reports include reports for both "all the classes(including 'O') " and "excluding class 'O'" to analyse the performance change. The result also includes top likely and unlikely transitions using CRF models.

1 Introduction

Modern era of internet, a vast amount of data is available and most of them are unstructured and unannotated. With increasing volume of unannotated text data, labeling data is crucial to make it suitable for different applications but it is a challenging task. There is minor amount of supervised training data available for modelling. Therefore, sequence tagging is very crucial task to tackle this issue. Sequence tagging can be performed in several ways, like Part of Speech tagging (POS), chunking and Named Entity Recognition (NER)[16] . The task of extracting information from a literature and tagging it to a relevant class using Recurrent Neural Networks is called NER. "Named Entity" phrase was first mentioned at the sixth Message Understanding Conference in 1996 when the information extraction from unstructured text became an important problem[11]. NER is an integral part of natural language processing (NLP) system. NER is fundamental key for several applications and has been used as a critical component of NLP applications such as information extraction, machine translation, question-answering, automatic text summarization, text clustering, opinion mining (sentiment analysis), semantic search and other application in medical field to detect adverse drag effect, disease diagnosis, identification of heart disease risk factor, and extraction of biomedical entities, etc. NER deals with two process, first to identify names of in text, and second to classify them into set of predefined categories such as person, organization, date, location etc. It is challenging for machines to classify into categories with dictionary as proper noun are evolving continuously. The rest of project report is organized as follows. In the following sections, I reviewed some related work on the project topic, described the dataset, models used in this study, results from Random Forest Classifier model and Bi-LSTM – CRF model and followed by conclusions.

2 Related work

Named Entity Recognition is a popular technique used in information extraction to identify and segment the named entities and classify or categorize them under various predefined classes. It can be applied to different domains. In recent years, automatic named entity recognition and extraction systems became very popular and significant advances has been made. NER is typically classified into three categories- rule-based NER, machine Learning-based NER and hybrid NER . Machine learning-based approaches can be classified into three categories- supervised learning, semi-supervised learning and unsupervised learning. Hy-

brid approaches are often advantageous because it combines the results of multiple machine learning techniques or handcrafted rules. Goyal et al provided a detailed review of application of NER and recent advances in NER approaches and identified numerous issues and challenges of different approaches[8]. Perera et al shows the use of NER in Biomedical Information Extraction and reviewed the work related to Biomedical Named Entity Recognition (BioNER)[13]. They discussed about "Rule-based models" that uses user identified rules to capture named-entities and classify them based on their orthographic and morphological features and "Dictionary-based models" that apply large databases of named-entities and classify into different categories as a reference to locate and tag entities. Previous studies adopted unsupervised learning along with hand-engineered features and specialized knowledge resources. However , Peters et al utilized neural architectures for NER without any language specific resources such as gazetteers[14]. They utilized character-based word representations derived from supervised corpus and unsupervised word representations derived from unannotated corpora. They used character-based word representation model to represent orthographic sensitivity and dropout to avoid dependency on one representation class. Huang et al applied different types of LSTM-based models[19]. They applied a bidirectional LSTM CRF model for NLP benchmarking. They showed that BI-LSTMCRF model can efficiently use both past and future input features. This model is robust and does not depend on word embedding. Primary research includes Hidden Markov Models (HMM), Maximum entropy Markov models (MEMMs), and Conditional Random Fields (CRF). Several previous works discussed on LSTM network for several NER applications[6][7][18]. For example, Plank et al investigated sensitivity of bi-LSTMs models to the amount of training data and labeling noise. They introduce a novel bi-LSTM model that trained with auxiliary loss. Their model predicts the POS and the log frequency of the word [15]. My contribution in this project is to explore and evaluate the accuracy of sequence tagging using vanilla LSTM and bidirectional LSTM – CRF model[19].

3 Dataset

Citation of dataset: GMB (Groningen Meaning Bank) corpus[2]

Description: The Groningen Meaning Bank (GMB)[2] consists of public domain English texts with corresponding syntactic and semantic representations. This dataset is developed at the University of Groningen. It comprises thousands of texts in raw and tokenized format. The semantic annotations in the GMB are labelled using the BIO scheme, where each entity label is prefixed with either B or I letter. B- denotes the beginning and I- inside of an entity. The words which are not of interest is labelled with O [1].
Size and Characteristics: The GMB is very large corpus. This dataset contains 4 columns Sentence, Word, Pos, Tag Total Words Count = 1354149 Target Data Column: "tag" The classes in the dataset are:geo = Geographical Entity, org = Organization, per = Person, gpe = Geopolitical Entity, tim = Time indicator, art = Artifact, eve = Event, nat = Natural Phenomenon. One of the main reasons to use this dataset is its availability and accessibility. Also, the dataset is very large which makes it very useful in training neural networks. This dataset comes with labels and is preprocessed and tokenized which reduces the effort of data cleaning. The dataset is available online. It is accessible and not restricted with permission. The processed version is available in [1]. It can be downloaded from "Annotated Corpus for Named Entity Recognition" [2][1]
Glimpse of GMB dataset: Table 1 shows the glipmse of GMB dataset. All the words in individual sentences are tagged with 'POS' and 'Tag'.

Index	Sentence	Word	POS	Tag
0.0	1.0	Thousands	NNS	O
1.0	1.0	of	IN	O
2.0	1.0	demonstrators	NNS	O
3.0	1.0	have	VBP	O
4.0	1.0	marched	VBN	O

Table 1: GMB dataset

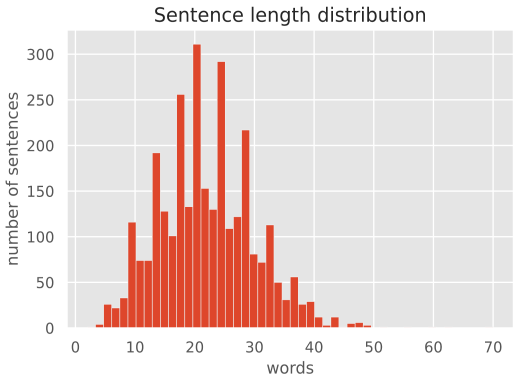

Figure 1: Sentence length distribution of GMB corpus[4]

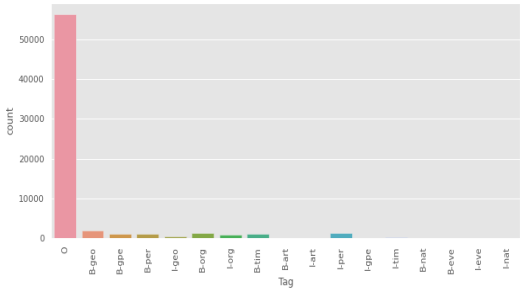

Figure 2: Word distribution across Tag

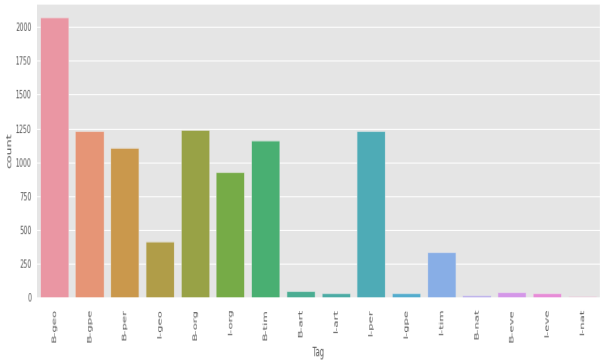

Figure 3: Word distribution across Tag removing O class

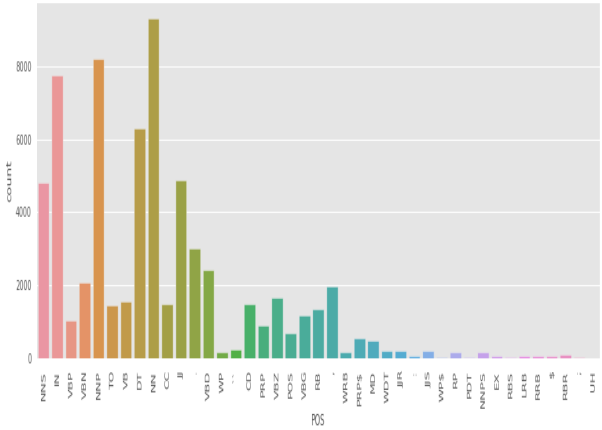

Figure 4: Word distribution across POS

In figure 1, the sentence length distribution is normal between the 10 to 40 mostly. Figure 2 shows the word distribution across tag for the dataset. The distribution is very skew as the data for class 'O' has more than 50,000 words. The number of samples for all other classes are very less as compared to class 'O'. This shows the dataset is highly imbalanced and does not have a normal distribution. This can results in class bias during model evaluation and may raise problems during classification task.It may happen that the model will not able to predict data from other classes with high precision and always predict class 'O' for most of the samples. To mitigate this, several approaches are there for under sampling majority class or oversampling minority class samples. From figure 3, it can be observed that after removing the class 'O', the data distribution across all other classes are quite similar.Figure 4 shows the data distribution across POS feature and it can be observed that the data from the first few bars are more in count that the later bars.

4 SMOTE - Oversampling Minority class data

SMOTE - Synthetic Minority Oversampling Technique over samples the minority class in the dataset. It will first selects a point randomly from a feature space and then it will look for its nearest neighbours from the same class. A point will be generated or placed on the vector joining the two randomly selecting points. The newly added point will be lamda percent way from the two points. Figure: 5 shows the new point x_{new} has been placed in between points $x_z i$ and x_i of minority classes to generate synthesized data samples for the minority class. Since, this technique works on random selection of points from the feature space, it may have some drawback of creating overlapping features.

5 Models

In this project I explored the Named Entity Recognition task using three models namely traditional Random Forest classifier, Conditional Random For-

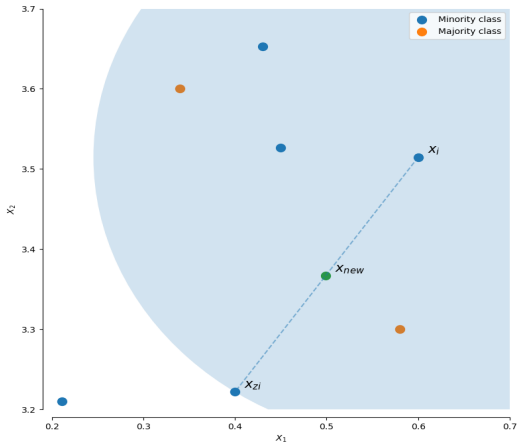

Figure 5: SMOTE - Synthetic Minority Oversampling Technique[5]

est and Bi-LSTM - CRF model.

5.1 Random Forest Classifier

Random Forest Classifier is a tree based ensemble algorithm mainly used for classification and regression tasks. The model's performance has been evaluated using the Precision, Recall and F1 score evaluation metrics. Since, the dataset has been very imbalanced, the results are evaluated on all the classes and then only on B and I labels excluding class 'O'. The experiments also includes oversampling algorithm. Splitting the dataset into train and set set. Then implementing the SMOTE to over-sample the minority class on only training dataset. The number of samples for each classes are same as shown in the figure 6. The model then trained on the over-sampled dataset and evaluated on the test set (without synthesized data).

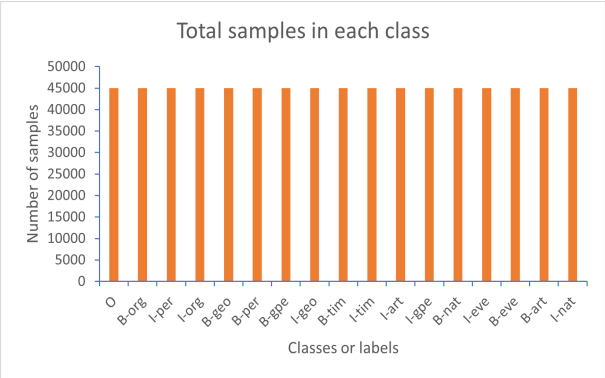

Figure 6: Number of samples in each class using SMOTE[5]

5.2 CRF Network

In year 2001 [10] proposed conditional random field model for text segmentation and sequence labelling tasks. The CRF model works on conditional probability $p(Y|X)$. CRF is discriminative model and uses supervised learning. CRF are represented as an undirected graphical model. The paper shows the probability of label sequence Y given sequence of observation X to be a normalized product of potential function in CRF models [17] (which is also called as global per sequence normalisation). Maximum entropy Markov models shows the problem of Label Bias Problem. Figure 7 shows going from state 0 to 3 will have same probability score. Since, MEMMs work on local normalization of a state, each state's forward transition probability will sum to 1. Figure 7 shows, from state 1 to 2 and 4 to 5 have same likelihood score. CRF models resolve the drawback of Maximum entropy Markov models (MEMMs)[12]. In this project CRF has been implemented using LGBFS algorithm (Gradient descent using the L-BFGS method). Gradient descent is beign used as the optimization function here. CRF models are always better as compared to the traditional tree based models.

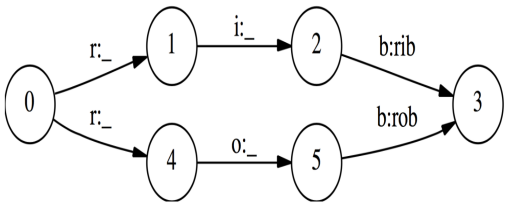

Figure 7: *Label bias problem in MEMMS[10]*

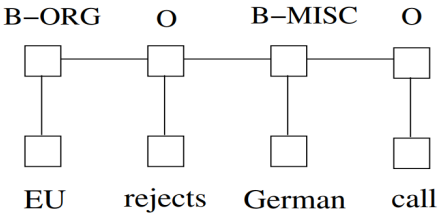

Figure 8: *A CRF network[19]*

5.3 Bi-LSTM-CRF Networks

LSTM is a recurrent neural network (RNN) and are proved as one of the powerful networks for NLP tasks. It maintains a memory based history information, which enables the model to predict

the current output conditioned on long distance features [19]. Figure 9 shows the LSTM network.

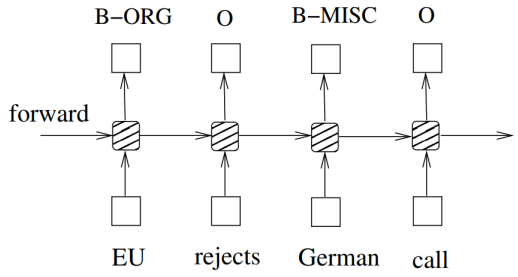

Figure 9: A LSTM network[19]

The Bi-LSTM network is an advanced version of vanilla LSTM. It can have access to both future and past input feature knowledge using both directional transition edges. The input to Bi-LSTM network will be word embedding vectors. The outputs from the Bi-LSTM layer will be the probability or emission scores generated from each Bi-LSTM nodes. On top of it, a CRF layer has been added. The Bi-LSTM layer passes the information to CRF layer. The CRF layer have the emission scores for all the words in an input sequence. Using these emission scores, it will compute the transition matrix for the entire input sequence. Based on the transition matrix, the highest transition probability of labels will be chosen for each word. The CRF layer has many constraints such as:

- The label of the first word in a sentence should start with "B-" or "O", not "I-"
- "B-label1 I-label2 I-label3 I-…", in this pattern, label1, label2, label3 … should be the same named entity label. For example, "B-Person I-Person" is valid, but "B-Person I-Organization" is invalid.
- "O I-label" is invalid. The first label of one named entity should start with "B-" not "I-", in other words, the valid pattern should be "O B-label"

Due to these constraints, Bi-LSTM-CRF architecture performs state of the art results for information extraction tasks. Bi-LSTM networks are different from LSTM with an additional back word arrow accessing the future and past both input features [9]. The forward and back word passes in all the time stamp will compute the probability of each word in a sentence based on the t+1 and t-1 words. This project follows the implementation details as mentioned in [9]. The paper used the batch implementation which enables multiple sentences to be processed at the same time. An additional CRF layer on top of Bi-directional network in Figure 10 shows the Bi-LSTM - CRF network.

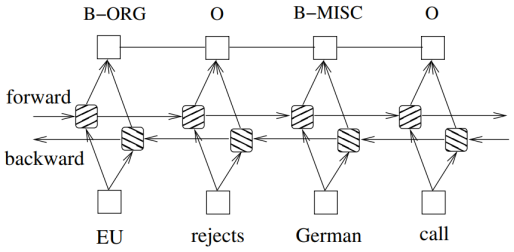

Figure 10: A bidirectional LSTM - CRF network[19]

6 Evaluation metrics

Information extraction tasks are most likely evaluated on classification report which includes Precision, Recall and f1 score metrics. The definition of each metrics are as follows: True Positives (TP) - These are the correctly predicted positive values which means that the value of actual class is yes and the value of predicted class is also yes. True Negatives (TN) - These are the correctly predicted negative values which means that the value of actual class is no and value of predicted class is also no. False Positives (FP) – When actual class is no and predicted class is yes. False Negatives (FN) – When actual class is yes but predicted class is no. Precision - Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.

$$Precision = \frac{TP}{TP + FP}$$

Recall (Sensitivity) - Recall is the ratio of correctly predicted positive observations to the all observations in actual class - yes.

$$Recall = \frac{TP}{TP + FN}$$

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 the harmonic mean of the both Precision and Recall

$$F1_{score} = 2 * \frac{Recall * Precision}{Recall + Precision}$$

7 Experiment Set-up and Model training

In this project I have trained three models - Random Forest Classifier, Conditional Random Forest and Bi-directional LSTM network. First I started with simple tree based algorithm RFC. The n_estimators for RFC has been set to 500. For CRF model, new features (like word.isupper(), word.islower(), word.isdigit(), word.istitle() has been added in the feature set to learn the context behind each word accurately. In Bi-LSTM-CRF network, the model has been trained for 20 epochs. The gradients of the model has been updated with a learning rate of 0.0005 and using adam optimizer. A dropout of 0.5 has been added to the model to decrease the over-fitting. The maximum length of an input sequence is 140 with a word embedding size of 150. The main activation function is 'relu'. For CRF and RFC implementation, "sklearn-crfsuite" and "RandomForestClassifier" has been used. For Bi-LSTM-CRF network, keras module has been used.

8 Results

8.1 Random Forest Classifier results

	precision	recall	f1-score	support
B-art	0.00	0.00	0.00	14
B-eve	0.00	0.00	0.00	8
B-geo	0.22	0.78	0.35	385
B-gpe	0.24	0.06	0.10	241
B-nat	0.00	0.00	0.00	7
B-org	0.62	0.15	0.24	232
B-per	1.00	0.14	0.25	207
B-tim	0.26	0.32	0.29	229
I-art	0.00	0.00	0.00	5
I-eve	0.00	0.00	0.00	3
I-geo	0.00	0.00	0.00	76
I-gpe	0.00	0.00	0.00	8
I-nat	0.00	0.00	0.00	4
I-org	0.35	0.04	0.07	185
I-per	0.44	0.02	0.03	253
I-tim	0.57	0.06	0.10	71
O	0.97	0.98	0.98	11305
accuracy			0.87	13233
macro avg	0.28	0.15	0.14	13233
weighted avg	0.89	0.87	0.86	13233

Table 2: Random Forest - Classification Report including all the classes

Table 2 shows the precision, recall and f1 score using Random Forest Classifier. The result shows high accuracy of 87. For class 'O' all three scores are more than 97. For all other classes the precision, recall and f1 scores are either 0 or below 50. This is due to the highly imbalanced classes in the GMB dataset. Since, most of the words are tagged to class 'O' the model is highly biased towards that class. For a set of test data, if 80 percent of it is tagged to class 'O' and 20 percent to all other classes, the model will produce a high accuracy even if the model predicts entire 20 percent data incorrectly. This shows, the model is not learning the context and only memorizing. This can be shown from table 3

	precision	recall	f1-score	support
B-art	0.00	0.00	0.00	14
B-eve	0.00	0.00	0.00	8
B-geo	0.22	0.78	0.35	385
B-gpe	0.24	0.06	0.10	241
B-nat	0.00	0.00	0.00	7
B-org	0.62	0.15	0.24	232
B-per	1.00	0.14	0.25	207
B-tim	0.26	0.32	0.29	229
I-art	0.00	0.00	0.00	5
I-eve	0.00	0.00	0.00	3
I-geo	0.00	0.00	0.00	76
I-gpe	0.00	0.00	0.00	8
I-nat	0.00	0.00	0.00	4
I-org	0.35	0.04	0.07	185
I-per	0.44	0.02	0.03	253
I-tim	0.57	0.06	0.10	71
micro avg	0.26	0.24	0.25	1928
macro avg	0.23	0.10	0.09	1928
weighted avg	0.40	0.24	0.19	1928

Table 3: Random Forest - Classification Report excluding class 'O'

Table 3 shows the result using Random Forest Classifier for all the classes except 'O'. Accuracy dropped significantly to 25 when class 'O' has been removed This shows that the accuracy is not be a perfect metrics for class imbalanced dataset.

Table 4 shows the result from oversampling the minority class in training set using SMOTE technique and testing on the original test set without including synthesized data. The overall accuracy of the model dropped from 87 to 75 however, the scores for most of the classes are still either 0 or less than 30. Training the model with over-sampled data did not performed well on test data set. This shows that the overall performance of Random Forest Classifier is not good on named entity task.

8.2 Conditional Random Forest

Table 5 shows the classification report for all the classes using conditional random forest. The precision, recall and f1 score improved significantly as compared to Random Forest Classifier. The scores

	precision	recall	f1-score	support
B-art	0.00	0.00	0.00	12
B-eve	0.01	0.25	0.02	8
B-geo	0.07	0.78	0.35	431
B-gpe	0.32	0.09	0.13	247
B-nat	0.01	0.17	0.02	6
B-org	0.38	0.11	0.17	250
B-per	1.00	0.13	0.23	211
B-tim	0.23	0.25	0.24	226
I-art	0.00	0.00	0.00	5
I-eve	0.00	0.00	0.00	8
I-geo	0.00	0.00	0.00	80
I-gpe	0.01	0.57	0.01	7
I-nat	0.00	0.00	0.00	2
I-org	0.18	0.07	0.10	167
I-per	0.50	0.00	0.01	259
I-tim	0.04	0.53	0.07	66
O	0.99	0.87	0.93	11248
accuracy			0.75	13233
macro avg	0.22	0.18	0.11	13233
weighted avg	0.89	0.75	0.80	13233

Table 4: Random Forest - Classification Report- model trained on over-sampled data(SMOTE oversampling technique)

	precision	recall	f1-score	support
B-art	0.40	0.25	0.31	8
B-eve	0.25	0.20	0.22	5
B-geo	0.69	0.84	0.76	414
B-gpe	0.82	0.77	0.79	258
B-nat	1.00	0.14	0.25	7
B-org	0.70	0.54	0.61	281
B-per	0.76	0.80	0.78	236
B-tim	0.90	0.84	0.87	240
I-art	0.00	0.00	0.00	3
I-eve	0.50	0.17	0.25	6
I-geo	0.62	0.52	0.56	81
I-gpe	0.67	0.33	0.44	6
I-nat	1.00	0.20	0.33	5
I-org	0.71	0.64	0.67	204
I-per	0.78	0.89	0.83	267
I-tim	0.83	0.66	0.74	86
O	0.99	0.99	0.99	11612
accuracy			0.95	13719
macro avg	0.68	0.52	0.55	13719
weighted avg	0.95	0.95	0.95	13719

Table 5: Conditional Random Forest - Classification Report- including all the classes

	precision	recall	f1-score	support
B-art	0.40	0.25	0.31	8
B-eve	0.25	0.20	0.22	5
B-geo	0.69	0.84	0.76	414
B-gpe	0.82	0.77	0.79	258
B-nat	1.00	0.14	0.25	7
B-org	0.70	0.54	0.61	281
B-per	0.76	0.80	0.78	236
B-tim	0.90	0.84	0.87	240
I-art	0.00	0.00	0.00	3
I-eve	0.50	0.17	0.25	6
I-geo	0.62	0.52	0.56	81
I-gpe	0.67	0.33	0.44	6
I-nat	1.00	0.20	0.33	5
I-org	0.71	0.64	0.67	204
I-per	0.78	0.89	0.83	267
I-tim	0.83	0.66	0.74	86
micro avg	0.75	0.74	0.75	2107
macro avg	0.66	0.49	0.53	2107
weighted avg	0.75	0.74	0.74	2107

Table 6: Conditional Random Forest - Classification Report- excluding class 'O'

from		to	transition probability
B-geo	$->$	I-geo	5.682330
B-tim	$->$	I-tim	5.437136
B-org	$->$	I-org	5.314368
B-per	$->$	I-per	4.978456
I-org	$->$	I-org	4.973689
I-tim	$->$	I-tim	4.936380
I-per	$->$	I-per	4.891806
B-art	$->$	I-art	4.760881
I-art	$->$	I-art	4.305233
I-geo	$->$	I-geo	4.233342
B-eve	$->$	I-eve	4.155746
B-gpe	$->$	I-gpe	3.678515
O	$->$	O	3.535741
I-gpe	$->$	I-gpe	3.384389
I-eve	$->$	I-eve	3.036805
B-nat	$->$	I-nat	2.489609
O	$->$	B-per	1.692267
B-geo	$->$	B-tim	1.602308
O	$->$	B-tim	1.439387
O	$->$	B-org	1.423159

Table 7: Conditional Random Forest - Top likely transition

for most of the classes are above 60. If we remove the class 'O' , the micro avg dropped to 75 from 95 (Table 6). However, CRF performs still better than random forest algorithm.

Table 7 shows the top 20 likely transition using CRF model. The highest transition from B-geo to I-geo have 5.437136 probability score. This shows that, if a word is tagged with geo entity and it is the beginning of the sentence, then its following tag should be inside of the same entity (i.e. I-geo). Similarly, I-tim will follow B-tim and so on. If a word is tagged with class 'O', then its following word should be tagged with the beginning of any entity. It should not have a following tag of I.

Table 8 shows the top 20 unlikely transition using CRF model. The transitions from class 'O' to any 'I-' has the lowest transition probability. This shows, if a word has been tagged with 'O' , then the following word can not be tagged with Inside of an entity. The next word should tagged with

from		to	transition probability
O	– >	I-eve	-0.97888
B-geo	– >	I-org	-0.980901
B-org	– >	I-per	-0.981992
B-gpe	– >	I-org	-1.007779
B-geo	– >	I-gpe	-1.019925
B-geo	– >	I-per	-1.087746
B-tim	– >	B-tim	-1.135647
B-org	– >	B-org	-1.141812
I-org	– >	I-per	-1.177864
B-geo	– >	B-org	-1.254548
O	– >	I-art	-1.443963
B-geo	– >	B-per	-1.697911
B-tim	– >	B-gpe	-1.900895
I-per	– >	B-per	-2.016075
O	– >	I-per	-2.364394
B-gpe	– >	B-gpe	-2.425464
B-per	– >	B-per	-2.549516
O	– >	I-tim	-2.973031
O	– >	I-org	-3.157237
O	– >	I-geo	-3.281237

Table 8: Conditional Random Forest - Top unlikely transition

beginning of an entity.

8.3 Bi-LSTM - CRF

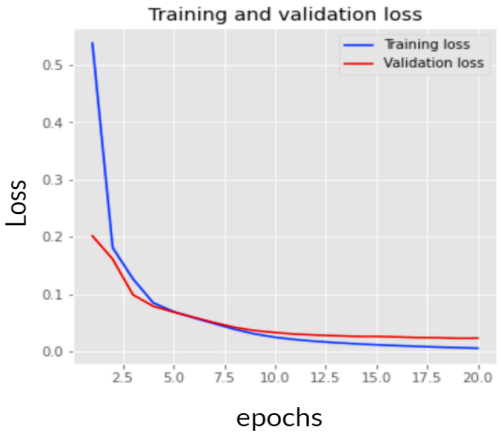

Figure 11: Bi-LSTM-CRF - Training and Validation loss

Figure 11 shows the Training and Validation loss. Loss minimizes quickly after 2.5 epoch and converged after 10 epochs during training the model. On Validation set, the starting loss is significantly low as compared to the training and converged after 15 epochs. The loss curve is a perfect elbow curve shows that the model has learned to tag the words under correct label.

Figure 12 shows the accuracy curve for training and validation set for 20 epochs. After 7 epoch, the accuracy has increased significantly from 96 to 99 for both training and validation. This shows that the

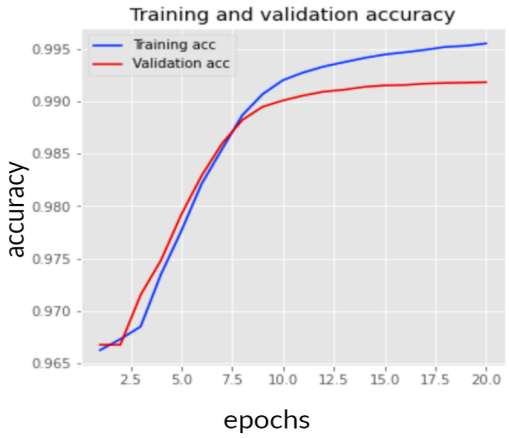

Figure 12: Bi-LSTM-CRF - Training and Validation accuracy

model has performed well and predicted the labels on validation set with 99 percent of accuracy.

	precision	recall	f1-score	support
B-art	0	0	0	92
B-eve	0.64	0.21	0.31	67
B-geo	0.83	0.87	0.85	7167
B-gpe	0.94	0.92	0.93	3202
B-nat	0	0	0	59
B-org	0.79	0.67	0.72	4129
B-per	0.83	0.76	0.79	3490
B-tim	0.91	0.86	0.88	3982
I-art	0	0	0	61
I-eve	0	0	0	52
I-geo	0.77	0.76	0.77	1362
I-gpe	0	0	0	52
I-nat	0	0	0	11
I-org	0.79	0.7	0.74	3252
I-per	0.85	0.77	0.81	3539
I-tim	0.82	0.68	0.74	1210
O	1	1	1	953313
accuracy			0.99	985040
macro avg	0.54	0.48	0.5	985040
weighted avg	0.99	0.99	0.99	985040

Table 9: Bi-LSTM- CRF - Classification Report

Table 9 shows the classification report using Bi-LSTM-CRF model. The precision, recall and f1 score is very high as compared to the random forest classifier and conditional random forest model above, for all the classes. Bi-LSTM with CRF layer model shows state of the art result with an accuracy of 99. For most of the classes, the Precision, recall and f1-scores are above 70. This shows, that the model has learned the context and predicted the correct labels with maximum probability score. This can be shown from true and predicted results for a given sentence in figure 13.

Word	True	Pred
Iran	B-geo	B-geo
says	O	O
the	O	O
F-4	B-org	B-geo
Phantom	I-org	O
jet	O	O
crashed	O	O
at	O	O
12.45	B-tim	B-geo
p.m.	I-tim	O
local	O	O
time	B-tim	O
(	O	O
915	O	B-tim
UTC	O	I-tim
)	O	O
Monday	B-tim	B-tim
in	O	O
waters	O	O
near	O	O
the	O	O
Iranian	B-gpe	B-gpe
port	O	O
city	O	O
of	O	O
Konarak	B-geo	O
.	O	O

Figure 13: Qualitative Analysis - Bi-LSTM-CRF - True and Predicted value

The label for the first word 'Iran' in the sentence has been predicted as "B-geo' same as the true. For most of the words, the model has predicted the correct value. However, there are few words like 'Konarak', '915', etc. for which the predicted labels are incorrect. This can be the result of data bias towards a particular class.

[1] has two sets of GMB data with size of 1354149 and 66641 words. For the Bi-LSTM CRF model I have used the larger dataset (1354149 - words).

9 Conclusion

The implementation for each model are inspired from kaggle implementation tutorial[3][4]. In this project, I compared the performance of Random forest classifier, Conditional random forest and Bi directional LSTM-CRF model on "Named entity recognition" task. The evaluation metrics precision, recall and f1 metrics for each model shows individual class score. The model Bi-LSTM with a CRF layer shows state of the art performance as compared to the other two models. CRF performed better in predicting labels with an avg accuracy of 75 (excluding class 'O'). After removing the biased class 'O', the accuracy dropped significantly from 87 to 25 using Random forest classifier. Using oversampling algorithm SMOTE to over-sample the minority class data, the performance of the random forest classifier did not improve. One of the reasons for this result may be due to the randomness in the process of generating synthesized data in SMOTE algorithm. Choosing a balanced dataset during model training is one of the best solutions to have a better predictive model.

10 Ethical Considerations

This project maintains the ethical codes. The data set used in this project is a GMB corpus and is cited properly. It is easily available and accessible without any restrictions. This corpus is in English language. Since, the project's main focus is on "Named Entity Task", it does not includes any human Intervention and therefore no sentiments has been harmed during the research. The reference from other literature and articles are cited properly.

References

[1] Annotated corpus for named entity recognition. https://www.kaggle.com/abhinavwalia95/entity-annotated-corpus.

[2] Groningen meaning bank data. https://gmb.let.rug.nl/data.php.

[3] Ner using bi-lstm. https://www.kaggle.com/williamroe/bi-lstm-with-crf-for-ner.

[4] Ner using random forest and crf. https://www.kaggle.com/shoumikgoswami/ner-using-random-forest-and-crf.

[5] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[6] Jason PC Chiu and Eric Nichols. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357–370, 2016.

[7] Chuanhai Dong, Jiajun Zhang, Chengqing Zong, Masanori Hattori, and Hui Di. Character-based lstm-crf with radical-level features for chinese named entity recognition. In *Natural Language Understanding and Intelligent Applications*, pages 239–250. Springer, 2016.

[8] Archana Goyal, Vishal Gupta, and Manish Kumar. Recent named entity recognition and classification techniques: a systematic review. *Computer Science Review*, 29:21–43, 2018.

[9] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. Ieee, 2013.

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

[11] Alireza Mansouri, Lilly Suriani Affendey, and Ali Mamat. Named entity recognition approaches. *International Journal of Computer Science and Network Security*, 8(2):339–344, 2008.

[12] Andrew McCallum, Dayne Freitag, and Fernando CN Pereira. Maximum entropy markov models for information extraction and segmentation. In *Icml*, volume 17, pages 591–598, 2000.

[13] Nadeesha Perera, Matthias Dehmer, and Frank Emmert-Streib. Named entity recognition and relation detection for biomedical information extraction. *Frontiers in Cell and Developmental Biology*, 8:673, 2020.

[14] Matthew E Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. Semi-supervised sequence tagging with bidirectional language models. *arXiv preprint arXiv:1705.00108*, 2017.

[15] Barbara Plank, Anders Søgaard, and Yoav Goldberg. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. *arXiv preprint arXiv:1604.05529*, 2016.

[16] Yanyao Shen, Hyokun Yun, Zachary C Lipton, Yakov Kronrod, and Animashree Anandkumar. Deep active learning for named entity recognition. *arXiv preprint arXiv:1707.05928*, 2017.

[17] Hanna M Wallach. Conditional random fields: An introduction. *Technical Reports (CIS)*, page 22, 2004.

[18] Donghuo Zeng, Chengjie Sun, Lei Lin, and Bingquan Liu. Lstm-crf for drug-named entity recognition. *Entropy*, 19(6):283, 2017.

[19] Kai Yu Zhiheng Huang, Wei Xu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.