Name Entity Recognition (NER)

By Team - Task Force X

...

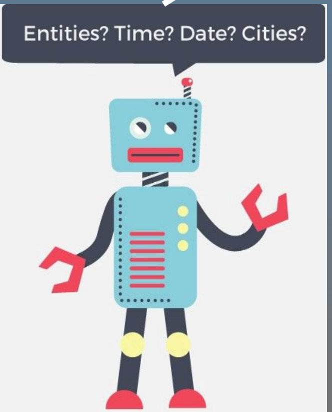
Team Members

Abhishek Yadav

Ishani Bari

Shruti Sachan

Tharun



Outline

- Introduction
- Methodology
- Results
- Conclusion
- Future Scope

Let's start with NER learning journey

Introduction -

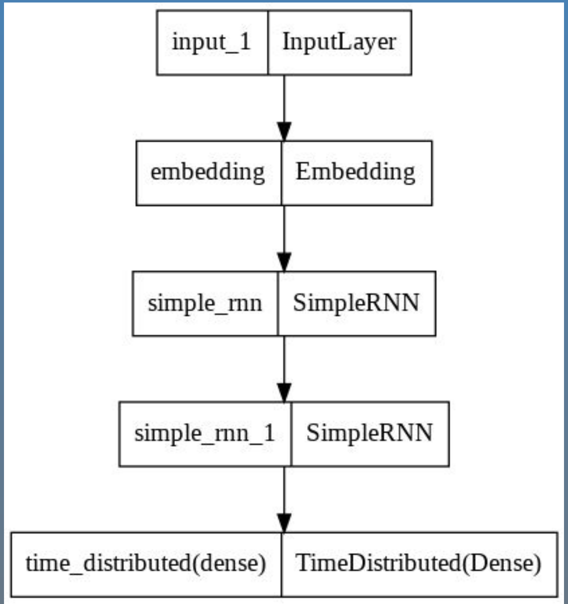
Problem Statement - Name Entity Recognition (NER) on WIESP (Workshop on Information Extraction from Scientific Publications) dataset to extract key information from scientific papers which can help search engines to better select and filter articles.

Approach towards the problem statement -

Data Pre-processing → Modelling → Evaluation and Prediction

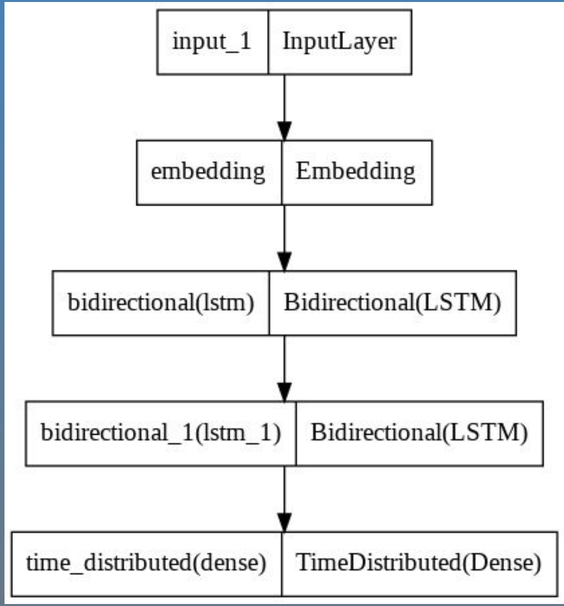
Introduction - Model Architectures

Baseline Model - Simple RNN layers



1. Embeddings. from scratch
2. Word2Vec Embeddings

Bidirectional LSTM Model



1. Embeddings. from scratch
2. Word2Vec Embeddings

Methodology

Dataset Description - Text fragments from astrophysics papers with manually tagged astronomical facilities and other entities of interest. The train data has 1753 sentences having NER tags and ids for each sentence tokens. While the test dataset has 1366 sentences.

Data Preprocessing done :

- Removing punctuation tokens having tag as 'O'.
- Removing outlier sentences (having length <10 or >600)
- Tokenization
- Creating tensorflow dataset.

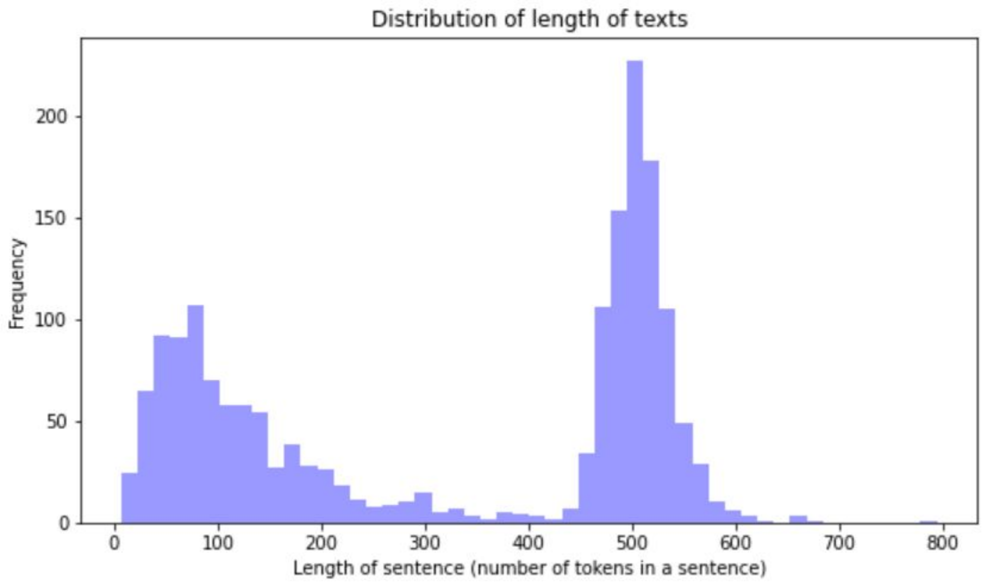


Fig 1. - Distribution of length of texts

Methodology

Modules Used

TensorFlow

GENSIM

scikit learn

NumPy

pandas

matplotlib

Distribution of NER tags

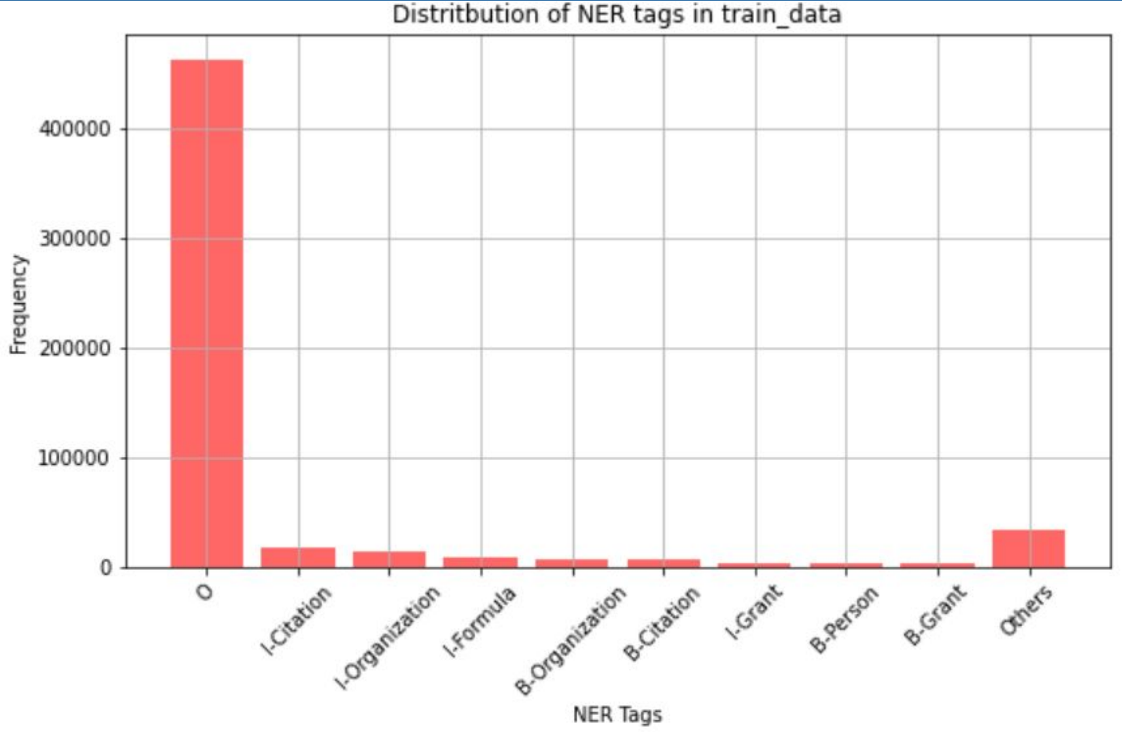


Fig 2. - Distribution of NER tags in train dataset

Results

Comparison of models

Model	Train Acc.	Val. Acc.	Train Loss	Val. Loss
Baseline Model (2 Simple RNN layers)	0.936	0.916	0.149	0.192
Bidirectional LSTM model	0.954	0.936	0.101	0.145
Baseline Model (Using Word2Vec Embeddings)	0.885	0.873	0.319	0.317
Bidirectional LSTM (Using Word2Vec Embeddings) model	0.908	0.895	0.208	0.222

Results

Actual tags for an example input sentence :

['We[O] would[O] like[O] to[O] acknowledge[O] the[O] pioneering[O] work[O] of[O] Adam[B-Person] Burrows,[I-Person] Hans-Thomas[B-Person] Janka,[I-Person] and[O] Matthias[B-Person] Liebendörfer,[I-Person] whose[O] contributions[O] to[O] CCSN[B-Model] theory[I-Model] were[O] directly[O] inspirational[O] to[O] the[O] work[O] presented[O] here.[O] This[O] material[O] is[O] based[O] upon[O] work[O] supported[O] by[O] the[O] National[B-Organization] Science[I-Organization] Foundation[I-Organization] under[O] Grant[B-Grant] No.[I-Grant] 1313036.[I-Grant]']

Predicted tags for an example input sentence :

['We[O] would[O] like[O] to[O] acknowledge[O] the[O] pioneering[O] work[O] of[O] Adam[O] Burrows,[O] Hans-Thomas[B-Person] Janka,[O] and[O] Matthias[O] Liebendörfer,[O] whose[O] contributions[O] to[O] CCSN[O] theory[O] were[O] directly[O] inspirational[O] to[O] the[O] work[O] presented[O] here.[O] This[O] material[O] is[O] based[O] upon[O] work[O] supported[O] by[O] the[O] National[B-Organization] Science[I-Organization] Foundation[I-Organization] under[O] Grant[O] No.[O] 1313036.[B-Grant]']

Results

Accuracy Plot and Classification Report of Best Model

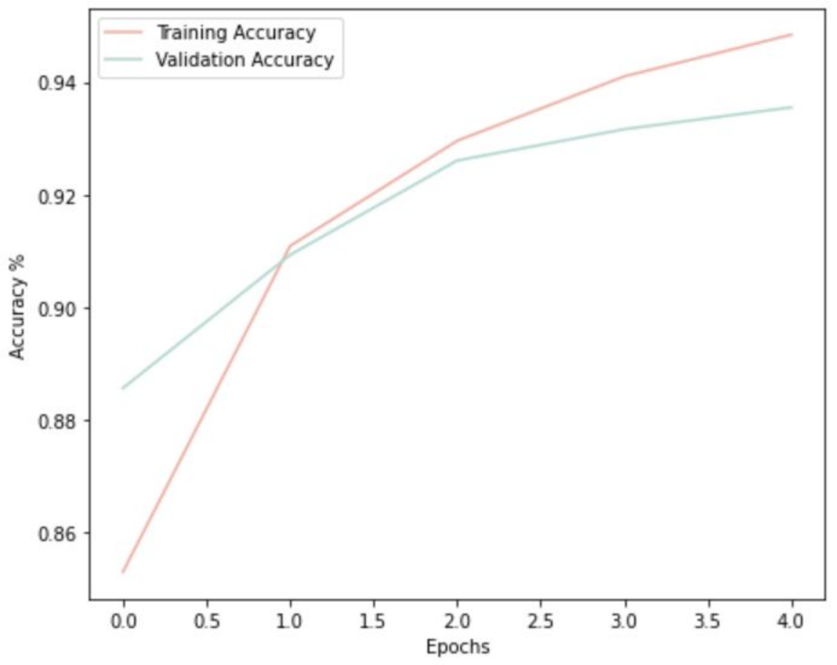


Fig 3. - Accuracy Plots of best model

	precision	recall	f1-score	support
I-Organization	0.87	0.89	0.88	3236
I-Person	0.92	0.79	0.85	545
I-Proposal	0.00	0.00	0.00	19
I-Software	0.67	0.11	0.18	113
I-Survey	0.63	0.64	0.64	104
I-Tag	0.00	0.00	0.00	2
I-Telescope	0.61	0.66	0.64	125
I-TextGarbage	0.00	0.00	0.00	9
I-URL	0.00	0.00	0.00	6
I-Wavelength	0.47	0.54	0.50	323
O	0.96	0.99	0.98	85234
accuracy			0.94	105574
macro avg	0.44	0.35	0.38	105574
weighted avg	0.92	0.94	0.93	105574

Fig 4 - Classification Report of best model (few tags)

Conclusion

- Bidirectional LSTM model having train accuracy of 95.4% and validation accuracy of 93.6%.
- As there is class imbalance in our dataset, we chose F1-Score as evaluation metric.
- There is dominance of 'O' tag in model due to which its F1-score is the highest. But, few of the other tags which are higher in counts after 'O' also have high F1-scores indicating that our model would predict entities well rather than just predicting 'O' tag.

Future Work

- Improving the F1_scores of rest of the tags by getting more data (tokens) of the tags which are less in the current dataset.
- Use Transformers: Using BERT like models for Fine-Tuning.
- We can also make changes to the model hyperparameters like epochs, number of LSTM Units, Activation Function used etc