Objective

Input: Words

Output: NER tags

Input: Words	Output: NER tags
Austin	B-Person
Hoag	I-Person

IOB tagging **format**[1] (Inside, outside, beginning)

I- prefix - tag is inside a chunk
B- prefix - tag is the beginning of a chunk that immediately follows another chunk without O tags between them
O tag - token belongs to no chunk

```
Examples from WIESP2022 dataset:
Word: NASA  NER Tag: B-Organization
Word: NNX13AP13G. NER Tag: B-Grant

General Examples
Alex I-PER, going O, Los I-LOC
Angeles I-LOC, California B-LOC
```

X: Input sentences → Tokenize and padding → Train test split

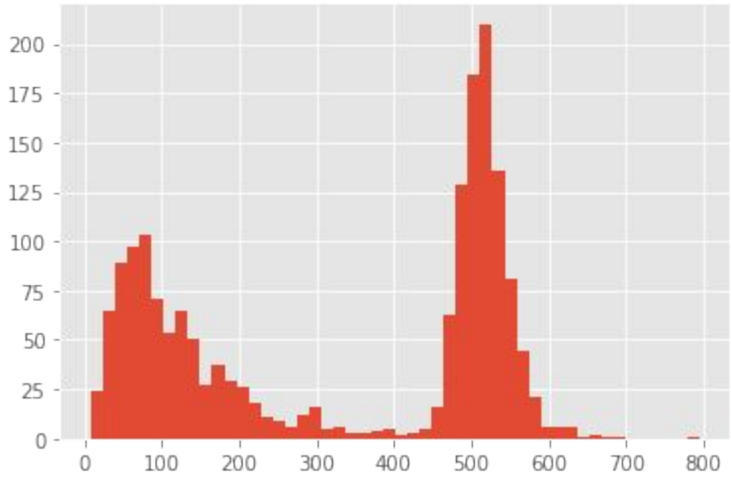
Y: Output NER tags → Map to indices and one-hot encoding → Train test split

Train model to predict NER tokens

Source
[1] https://en.wikipedia.org/wiki/Inside%E2%80%93outside%E2%80%93beginning_(tagging)#:~:text=The%20IOB%20format%20(short%20for,named%2Dentity%20recognition).

Data preprocessing

- list of tuples
- sentence lengths
- create word-to-index and index-to-word
- Padding
- Train_Test split
- GLOVE embedding



Model 1 (Base model):

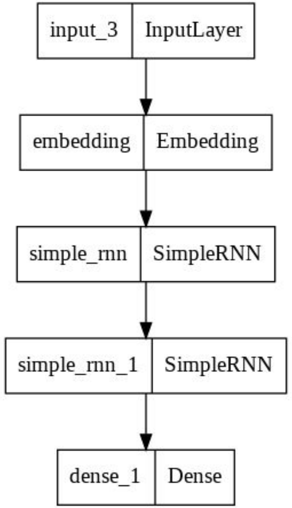


Figure 1: Model architeture

Glove embedding

Initialized using embedding matrix created from train vocabulary.

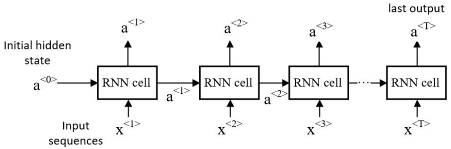


Figure 2: Simple RNN model example image[1]

Training

Training accuracy	97.9%
Validation accuracy	96.4%

RNN Prediction

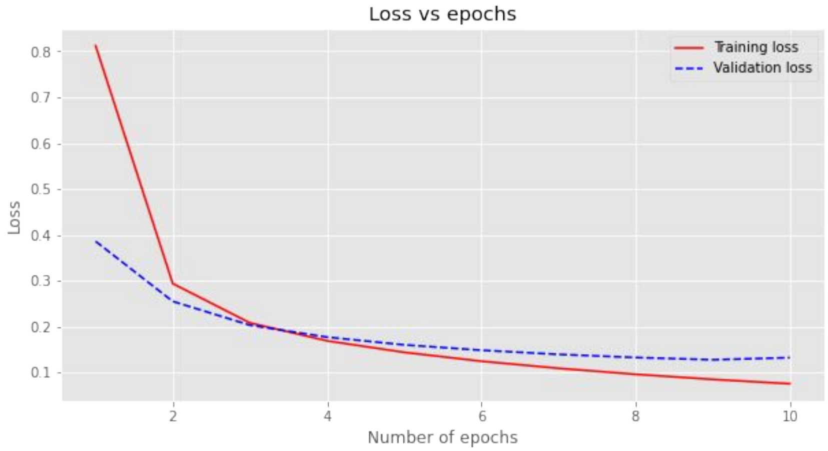
Hoag et al.	B-Citation I-Citation I-Citation	Correct
Harvard	B-Organization	Correct
HST-HF2-51413.001-A	B-Person	Incorrect
Peirce Fellowship	B-Person B-Fellowship	Incorrect

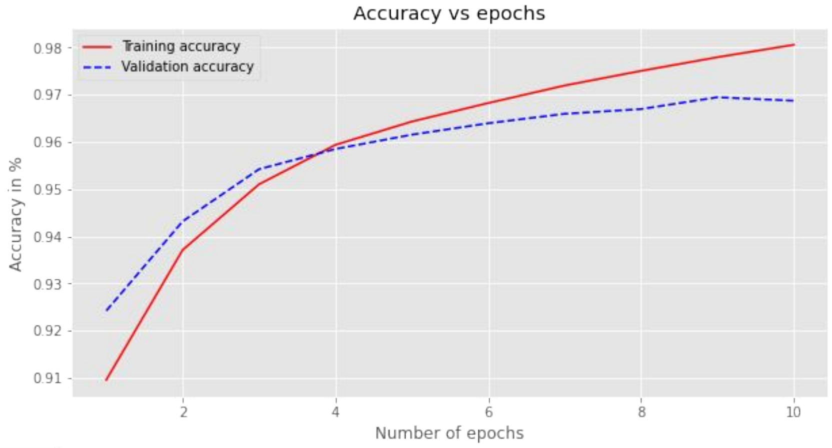
Source
[1] https://www.dlology.com/blog/how-to-use-return_state-or-return_sequences-in-keras/

Model 2: Bi-Directional LSTM

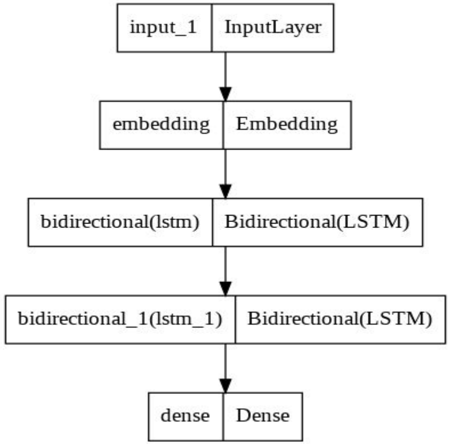
- As before, we merge individual tokens corresponding to the unique IDs to form sentences.
- The only difference this time around is that we utilize the Bidirectional variant of the Long-Short Term Memory model instead of a vanilla RNN with the goal of capturing richer context from the sentences. The reasoning behind this is that looking at a sentence from both directions allows us to capture richer past and future context that can prove to be crucial to label complex entities in a sentence.
- Since we had an appreciable performance with a two-layered base model, we create our model with two layers of BiLSTMs. Adding more layers would add unnecessary complexity in the model without a proportionate improvement in performance.

Model Architecture and Performance





Hoag et al.	B-Citation I-Citation I-Citation	Correct
Harvard	B-Organization	Correct
HST-HF2-514 13.001-A	B-Person	Incorrect
Peirce Fellowship	B-Fellowship I-Fellowship	Correct



Training accuracy	98.04%
Validation accuracy	96.87%

Future Work

- To further improve performance of the BiLSTM model we could input the logits obtained from the BiLSTM model to a Conditional Random Field (CRF).
- The CRF allows us to capture the relationship between the labels of two successive entities, much like a Hidden Markov Model (the current state is dependent on the predecessor state).
- Eg: If we have the label “B-Citation”, using a CRF will allow us to predict the label of the next entity as “I-Citation” with great likelihood.

Concluding Remarks

- In this project we develop a model that effectively tags the text fragments from an astrophysics dataset.
- We see that our model architecture outperforms the base model.
- The reason is that BiLSTMs capture more context.
- Example: For words like Ashford fellowship, the second word indicates that it is a fellowship. In our first model, where we only used forward RNNs we lost that context and tagged it incorrectly as O. In using the BiLSTMs, we were able to tag is as a fellowship.