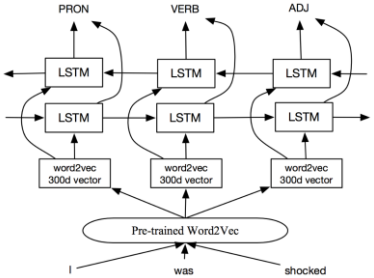
POS Tagging

Dong Liu (Donald) & Yunlin Peng (Linda)

Motivation and About the Project

POS tagging is the fundamental of many NLP tasks, for example, machine translation, QA, etc. We decided to realize the POS tagging in this project.

Below is an example how POS Tagging is realized by Bi-LSTM network.



Data and Labels

TreeBank Dataset from Upenn

The dataset contains

Number	Tag	Description
1.	CC	Coordinating conjunction
2.	CD	Cardinal number
3.	DT	Determiner
4.	EX	Existential *there*
5.	FW	Foreign word
6.	IN	Preposition or subordinating conjunction
7.	JJ	Adjective
8.	JJR	Adjective, comparative
9.	JJS	Adjective, superlative
10.	LS	List item marker
11.	MD	Modal

References

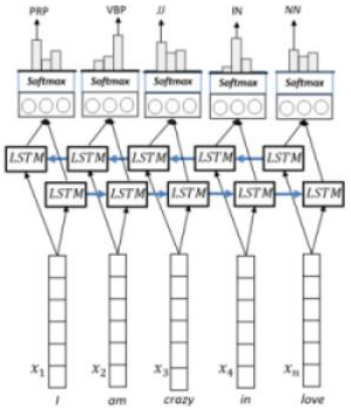
[1]1510.06168.pdf (arxiv.org)

[2]*https://ai.stackexchange.com/questions/28564/how-to-determine-the-embedding-size*.

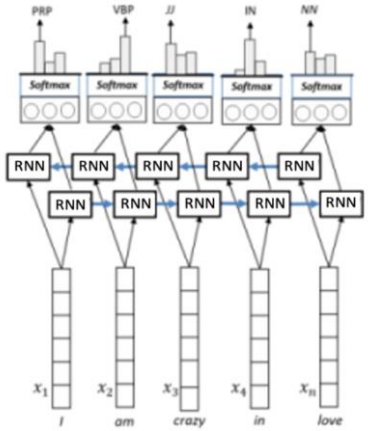
[3]*https://stats.stackexchange.com/questions/181/how-to-choose-the-number-of-hidden-layers-and-nodes-in-a-feedforward-neural-*

Model

Bi-LSTM



Bi-RNN



Results

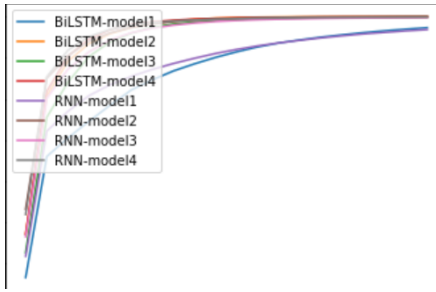


Fig.1 Accuracy Comparison

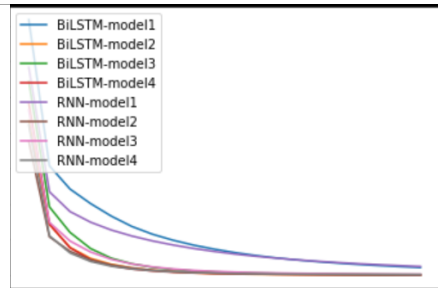


Fig.2 Loss Comparison

Model	Test Accuracy
Bi-LSTM (lr = 0.001, hidden_dim = 256, embedding_dim = 128)	88.40%
Bi-LSTM (lr = 0.005, hidden_dim = 256, embedding_dim = 128)	91.99%
Bi-LSTM (lr = 0.005, hidden_dim = 128, embedding_dim = 128)	92.14%
Bi-LSTM (lr = 0.005, hidden_dim = 128, embedding_dim = 256)	92.70%
Bi-RNN (lr = 0.001, hidden_dim = 256, , embedding_dim = 128)	88.72%
Bi-RNN (lr = 0.005, hidden_dim = 256, , embedding_dim = 128)	90.96%
Bi-RNN (lr = 0.005, hidden_dim = 128, , embedding_dim = 128)	90.85%
Bi-RNN (lr = 0.005, hidden_dim = 128, , embedding_dim = 256)	90.89%

Conclusion and Future Work

Conclusion:

We realize POS tagging by building Bi-LSTM and Bi-RNN network.

From the result, we can see that the Bi-LSTM model with (lr = 0.005, hidden_dim =128, embedding_dim = 256) has the best performance.

After analysing, we describe them by following reasons:

1. *A relative larger learning rate can solve the problem that a relative small learning rate will stuck to a local area.*
2. *A relative larger embedding size can get a model with more semantics meaning which will help to get a better result.*
3. *According to the reference [3], there is an experimental rule that "The number of hidden neurons should be between the size of the input layer and the size of the output layer." And the hidden dimension of the best model in our experiment obeys this rule.*

Future Work

1. *Using BERT + fine tuning to realize POS tagging.*
2. *Overfitting Problems: solutions*
   - *2.1. batch normalization*
   - *2.2. dropout*
   - *2.3. early stopping*
   - *2.4. data augmentation*