# Using structured learning to improve model performace in WiC task

Yuhang Fang, Zirui Hong, Jiangqiong Liu

May-2023

# Introduction

**Background**

- Includes structure information as rich features
- Add tree structure on sequence model for intuitive evaluation

**Objectives**

- Child-Sum Tree-LSTM (Dependency Tree-LSTM)
- N-ary Tree-LSTMs (Constituency Tree-LSTM)

**Research Questions to Answer**

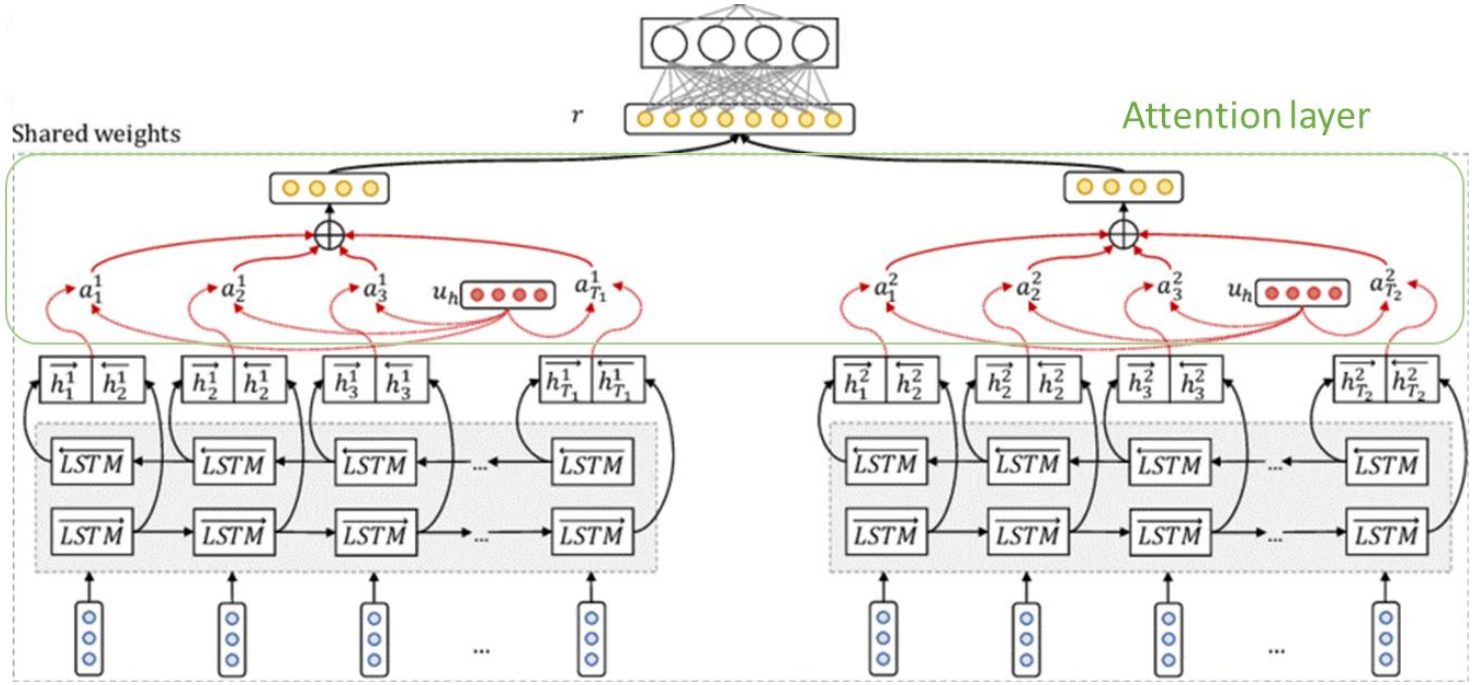
- Time Cost
  - Many loops for enumerating the tree nodes
- Dataset Quality
  - The sentence data sample in the WiC dataset is short and may hardly provide enough structure information

# Attention-based LSTMs

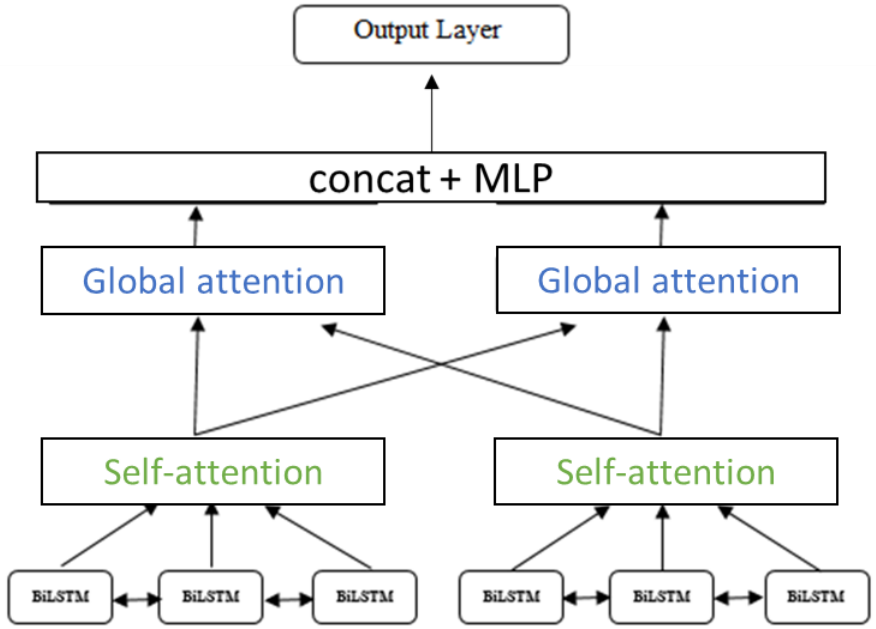
Use attention to get context vector:

1) Use the index of Target word
2) Use the last index of the sentence
3) Use the mean average of all

**Self-attention layer**

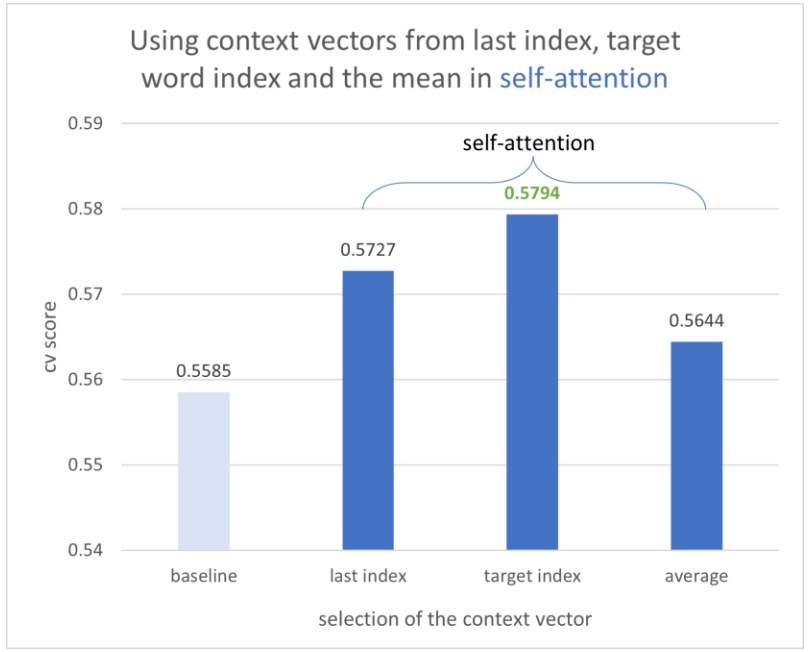


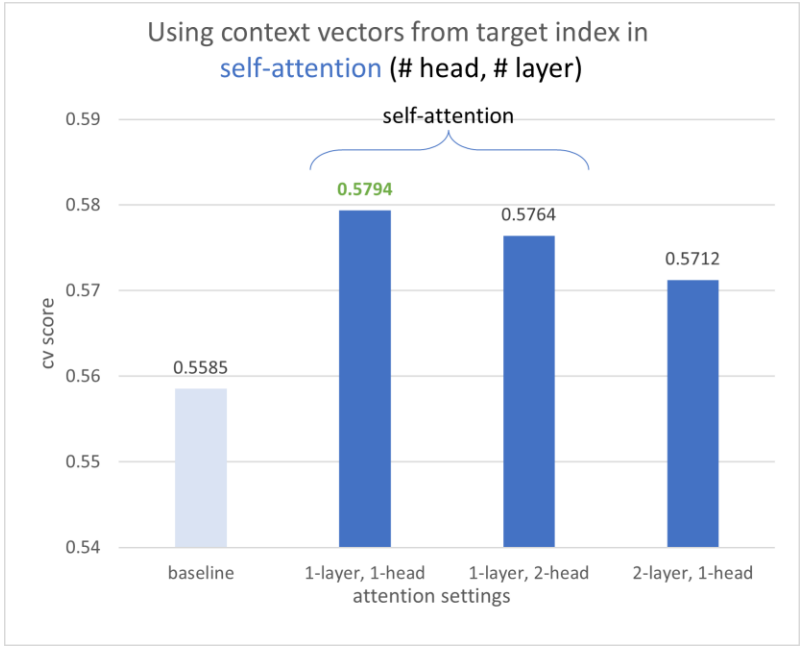

**Global attention layer**

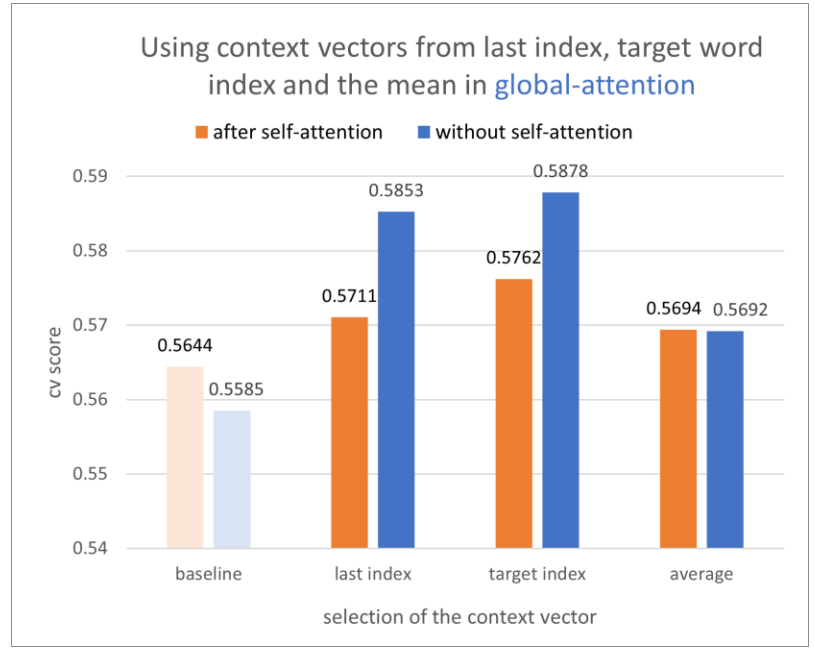


# Attention-based LSTMs

- Context source: Target index > last index > average of all
- Best self-attention: 1-head, 1-layer (2% improvement)
- Best global-attention: without self-attention (3% improvement)
- Best of all model: 0.5878 accuracy (lstm hidden states -> global attention -> target index)

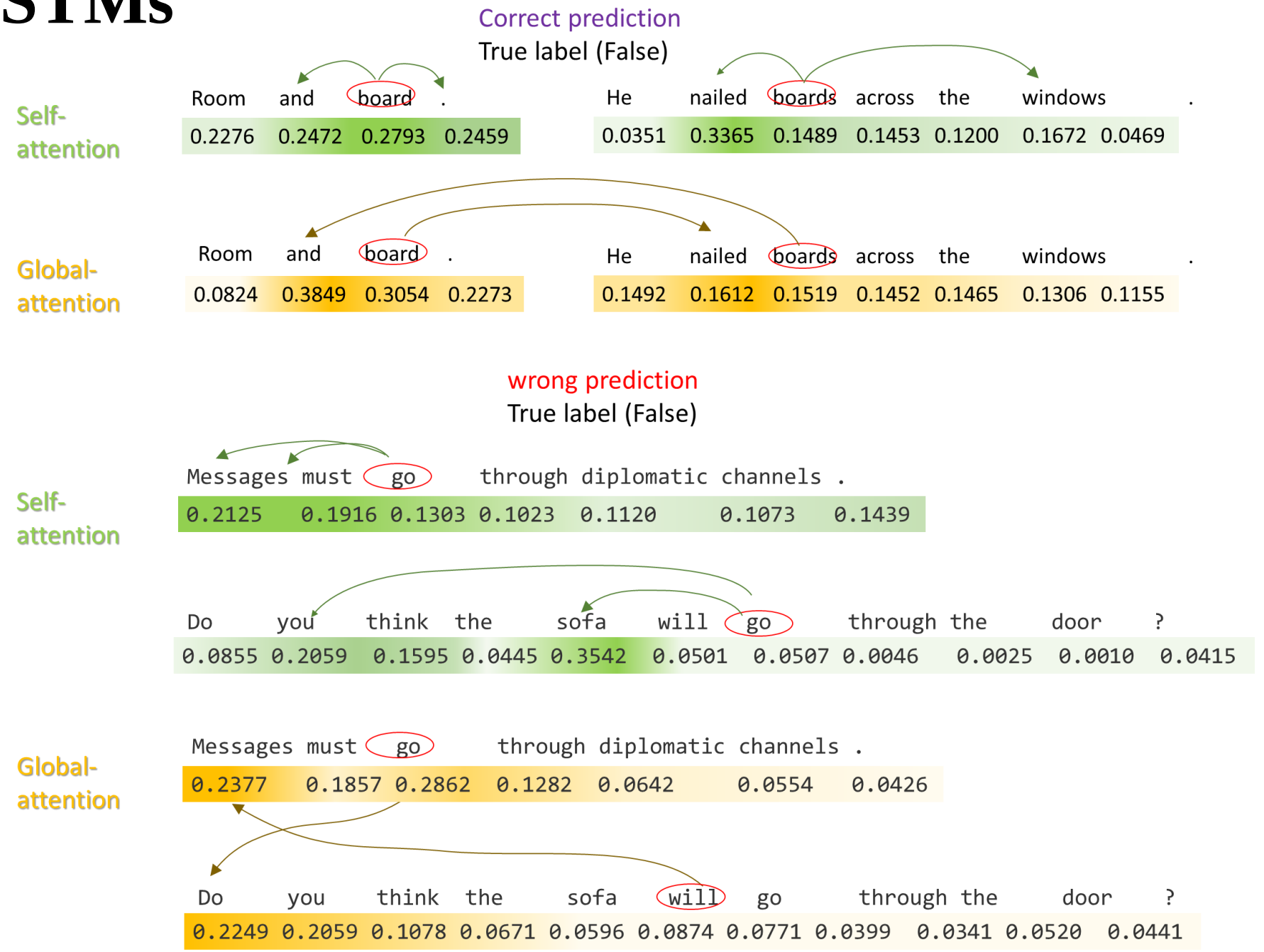


# Attention-based LSTMs

- **Self-attention** finds the dependency around target word inside a sentence
- **Global attention** finds the dependency of target word on words from the other sentence
- **The wrong prediction**
  - Inappropriate attention assignment
  - Global attention tends to get caught at starting words in a sentence?



*BiLSTM-CRF in our case*

T or F

*Task2: Learn word sense*

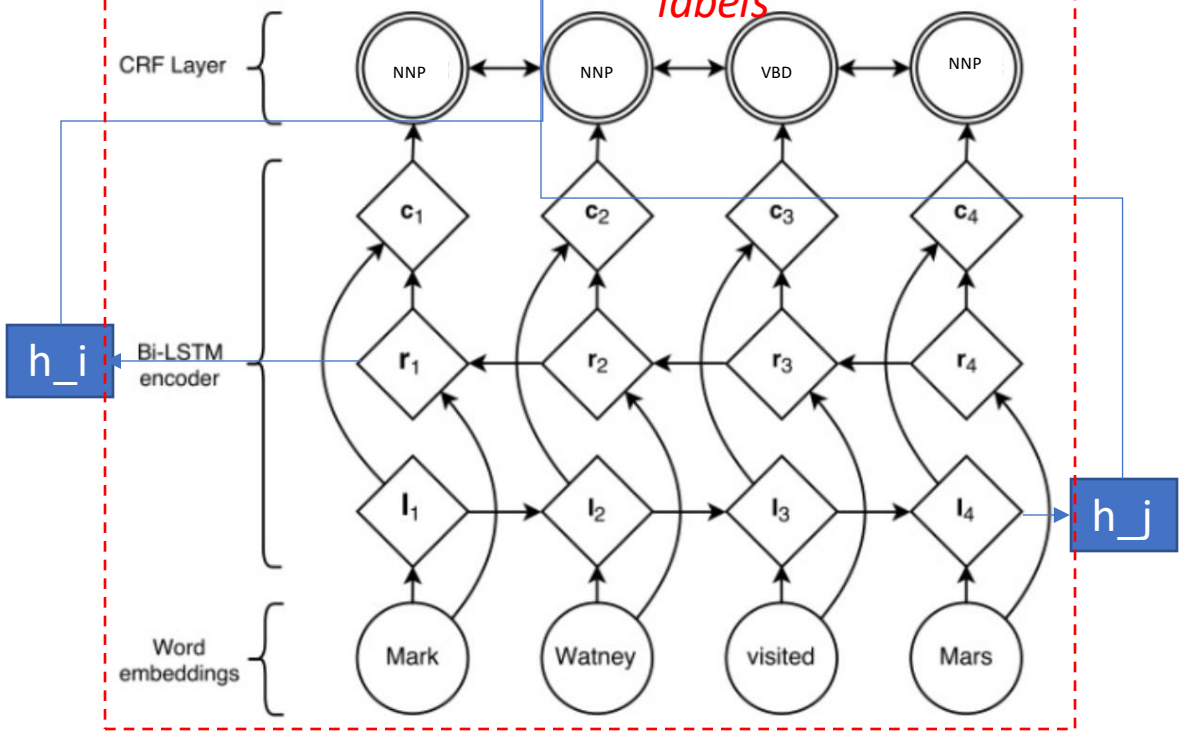
MLP

concat

concat

*Task1: Learn POS tag*

*Pos-tag labels*

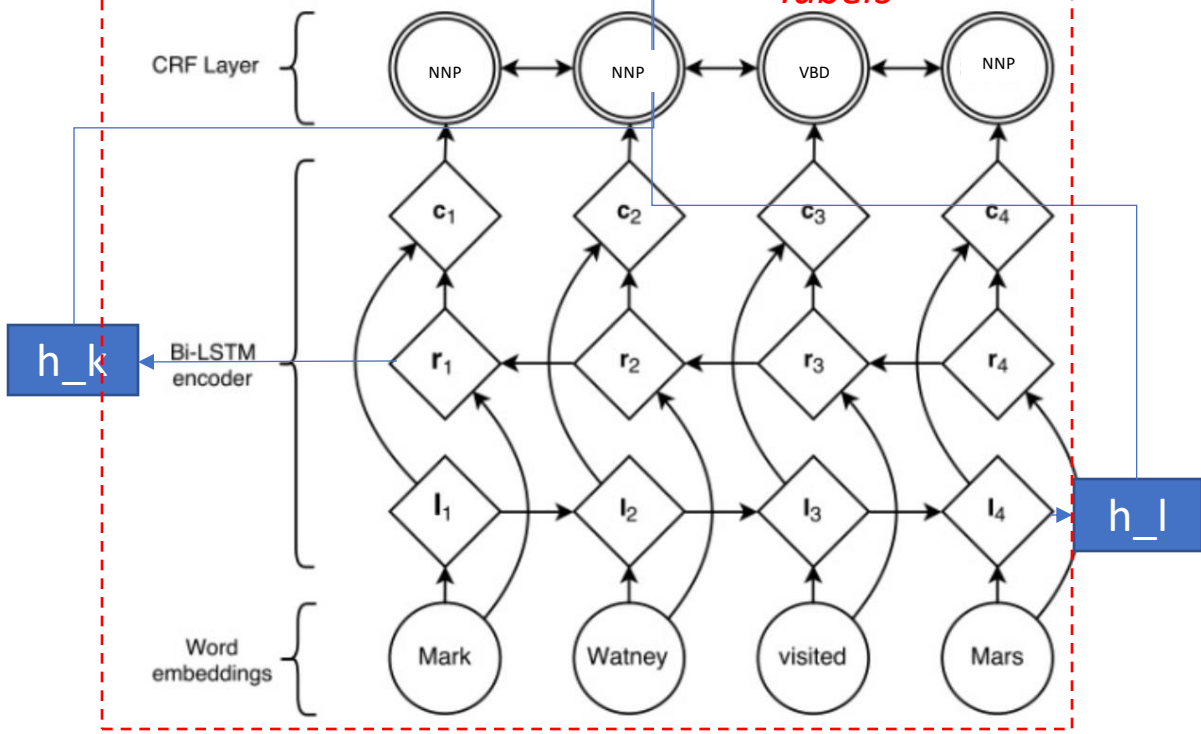



*Sentence 1*

concat

*Task1: Learn POS tag*

*Pos-tag labels*




*Sentence 2*

# LSTM-CRF

multitask learning approach with a bidirectional LSTM and CRF layer

- used pre-trained part-of-speech tagger model in NLTK to generate pos tags for input sentences
- used the hidden states from the bidirectional LSTM model to produce a sequence of feature vectors for each sentence
- fed the feature vectors into a CRF layer for training the part-of-speech tags

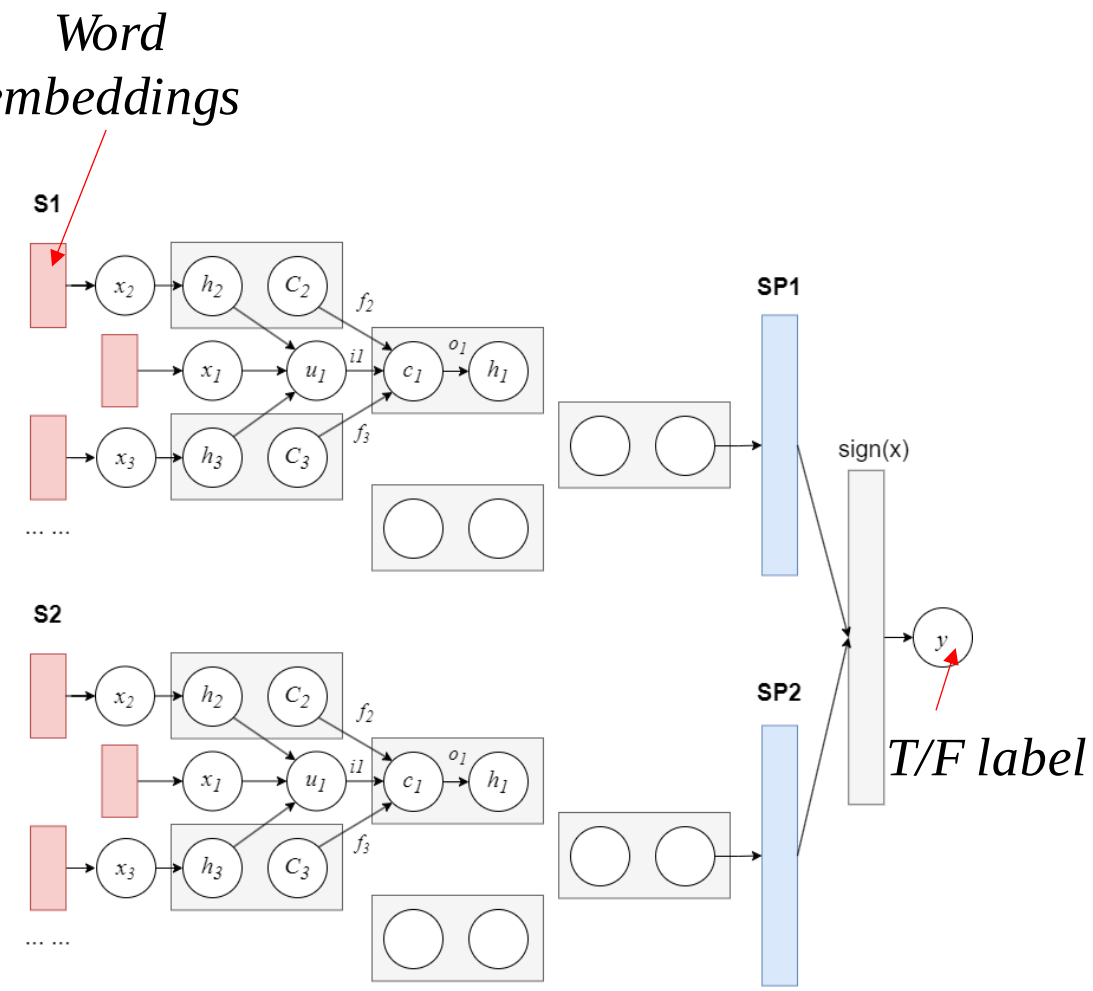
Challenges

- multitask learning increases the complexity of the model, which can make it more difficult to train and optimize because of large amounts of computation
- balancing the learning of the shared representations across tasks through determining the optimal weighting of the different tasks during training

Experiment

- the task of training part of speech tagging using a CRF layer is able to improve the accuracy of sentence comparison
- learn how to balance the learning of two tasks during training through tuning the weights

# Tree-LSTM (Exploration)




*Basic Architecture of Applied Tree-LSTMs*

**Intuition and Justifications**

- Includes structure information as rich features
- Add tree structure on sequence model for intuitive evaluation

**Tree-LSTM Variants**

- Child-Sum Tree-LSTM (Dependency Tree-LSTM)
- N-ary Tree-LSTMs (Constituency Tree-LSTM)

**Challenges**

- Time Cost
  - Many loops for enumerating the tree nodes
- Dataset Quality
  - The sentence data sample in the WiC dataset is short and may hardly provide enough structure information

*Future experiments are needed*

**Experiment Plan**

•Baseline: Bi-LSTM

# Discussion

- Self-attention layer is able to catch the dependency of target word on its context words
  - generating a context vector with more condensed information out of the highly attentive context words.
- Global attention shows how target word attend to the other sentence's specific part that can help to make the T/F decision,
  - can also serve as an explicit explanation to T/F label.
- However, stacking a global attention out of self-attention is not always good, as some information might get lost among the transition.

- Using a multitask learning approach with a bidirectional LSTM and CRF layer is able to improve the overall accuracy of tasks.
  - If the weight ratio between the training of pos tags and the task of sentence comparison is more than one or close to one, the accuracy for the sentence comparison task is low, and sometimes the accuracy could be lower than the baseline model.
- If time permits, we could use a machine learning algorithm like a grid search or a Bayesian optimization over a range of values of each task. This would on the other hand require more complexity and computational power.