Character-based Neural Networks for Sentence Pair Modeling

THE OHIO STATE UNIVERSITY

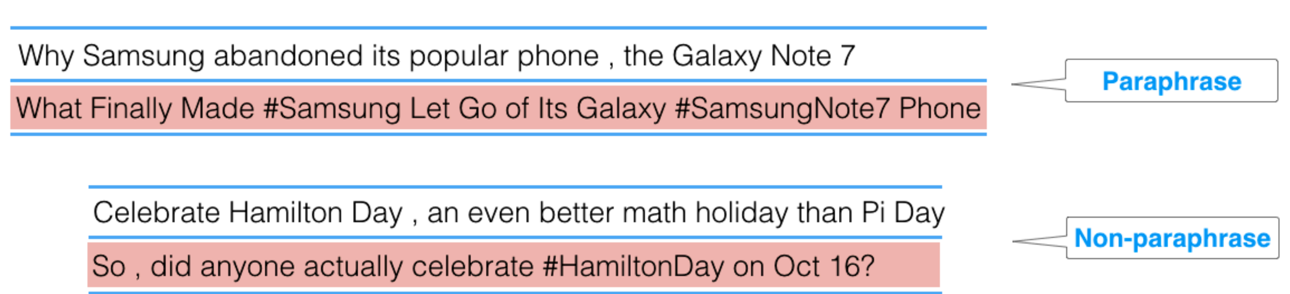
Wuwei Lan and **Wei Xu**

Introduction

- Sentence pair modeling is critical for paraphrase identification, question answering, natural language inference and etc.
- Various neural models achieved state-of-the-art performance by using pretrained word embeddings, however they have poor coverage in domain (e.g., social media) with high OOV ratio.
- We explored character-based neural networks for sentence pair modeling, which is more challenging than individual sentence modeling: similarly spelled words with completely different meanings could introduce error (e.g., *ware* and *war*).

Example for Sentence Pair Modeling

Paraphrase task: given a sentence pair, predict whether they imply the same meaning. Sample from **Twitter URL** [Lan et.al 2017]:



Pairwise Word Interaction Model (PWIM [He et.al 2016])

1. Context modeling:

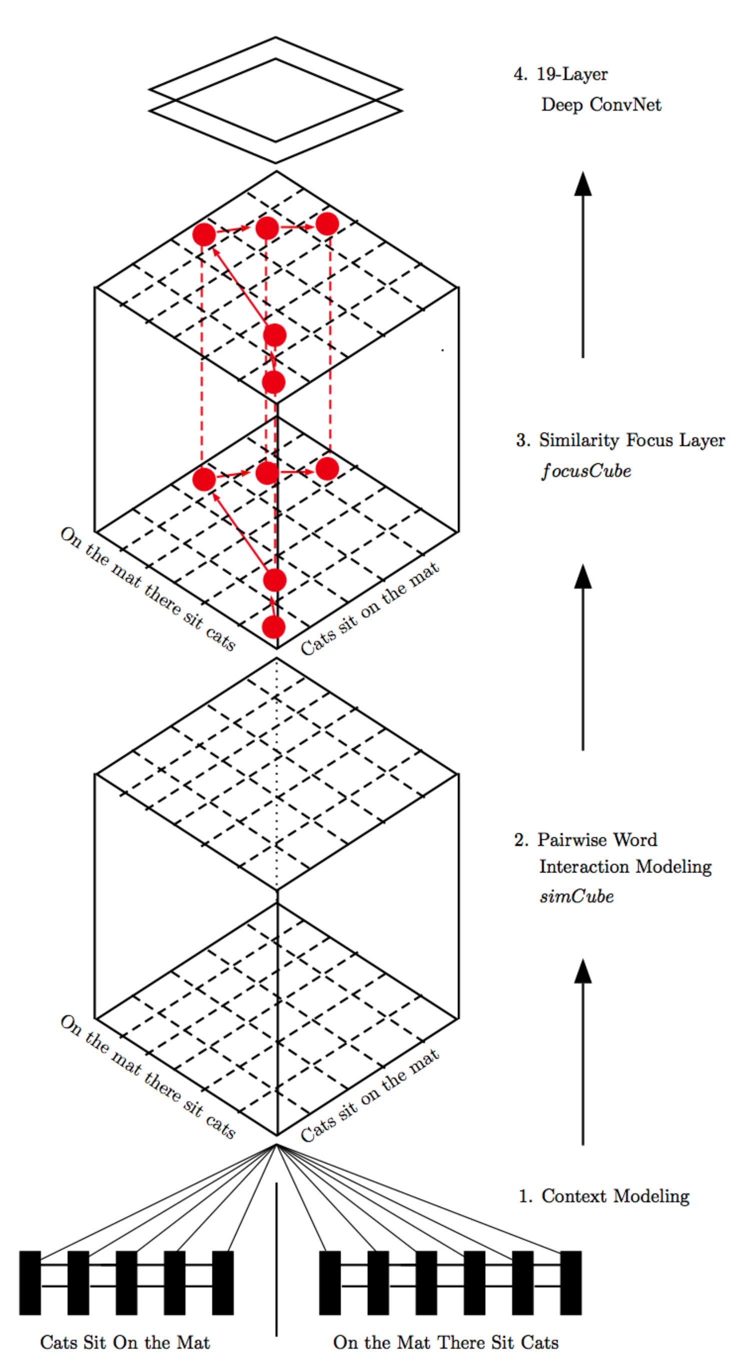
$$\overrightarrow{h}_i = LSTM^f(w_i, \overrightarrow{h}_{i-1})$$
$$\overleftarrow{h}_i = LSTM^b(w_i, \overleftarrow{h}_{i+1})$$
$$\overleftrightarrow{h}_i = [\overrightarrow{h}_i, \overleftarrow{h}_i]$$
$$h_i^+ = \overrightarrow{h}_i + \overleftarrow{h}_i$$

2. Pairwise word interaction:

$$D(\overrightarrow{h}_i, \overrightarrow{h}_j) = [cos(\overrightarrow{h}_i, \overrightarrow{h}_j), L2Euclid(\overrightarrow{h}_i, \overrightarrow{h}_j), DotProduct(\overrightarrow{h}_i, \overrightarrow{h}_j)].$$

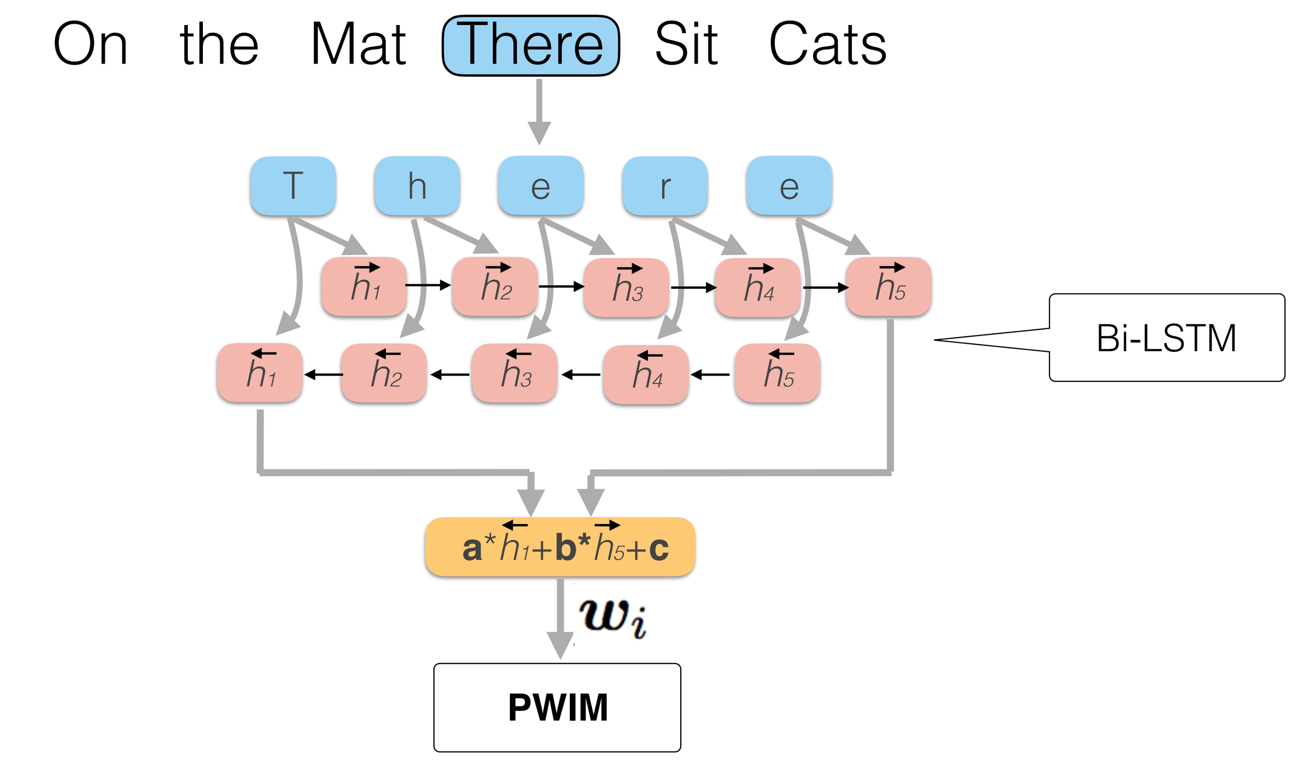
3. Similarity focus:
sorting the interaction values and selecting top ranked pairs

4. Aggregation and prediction:
19-layer deep ConvNet

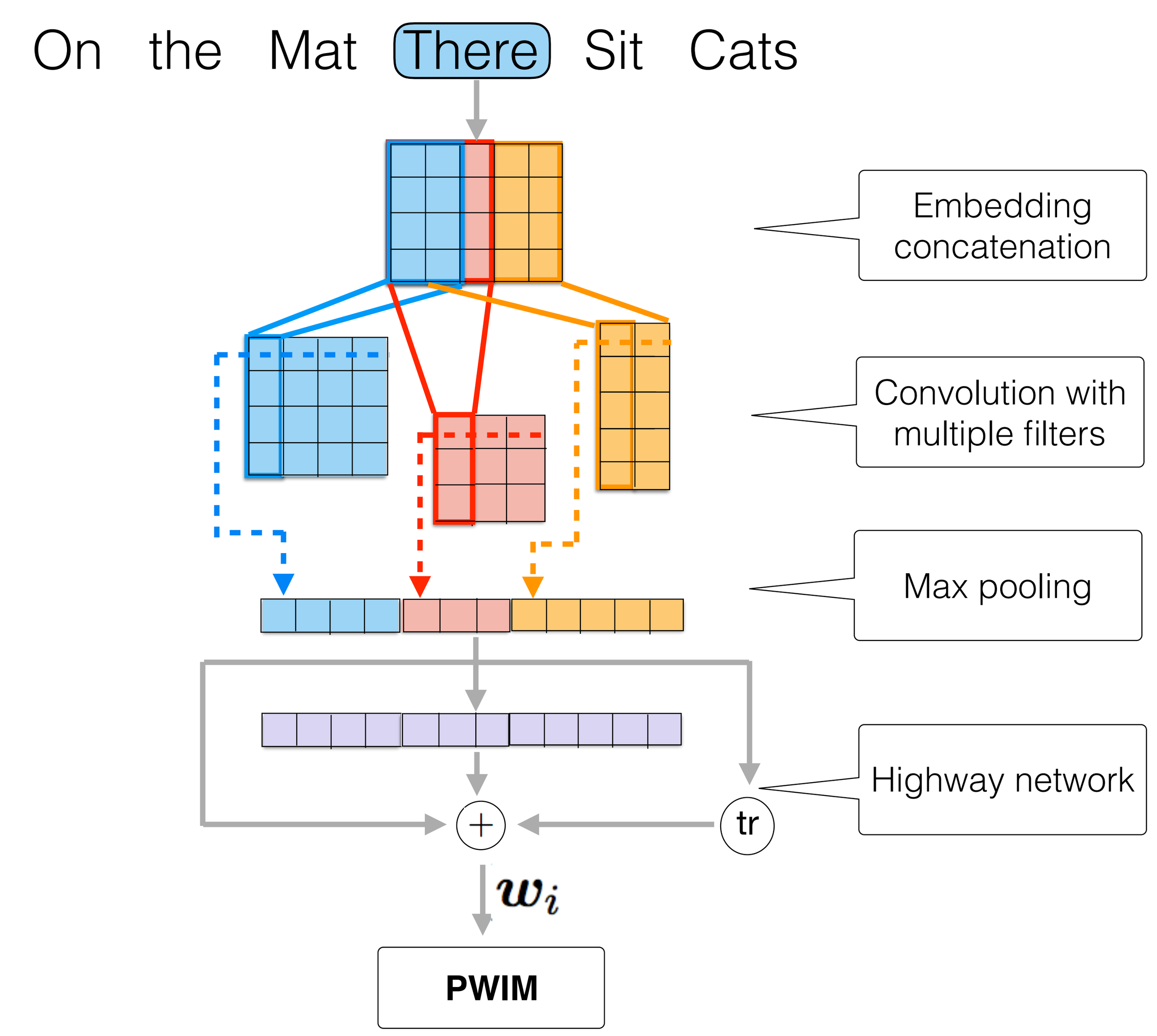


Embedding Subwords in PWIM

1. Char RNN [Ling et.al 2015]



2. Char CNN [Kim et.al 2016]



Conclusion

- ✓ Pretrained word embedding is not a necessity for sentence pair modeling.
- ✓ Subword models without any pretraining achieved new SOTA results in Twitter URL and PIT-2015.
- ✓ Multitask LM can improve subword embedding performance by injecting semantic information.

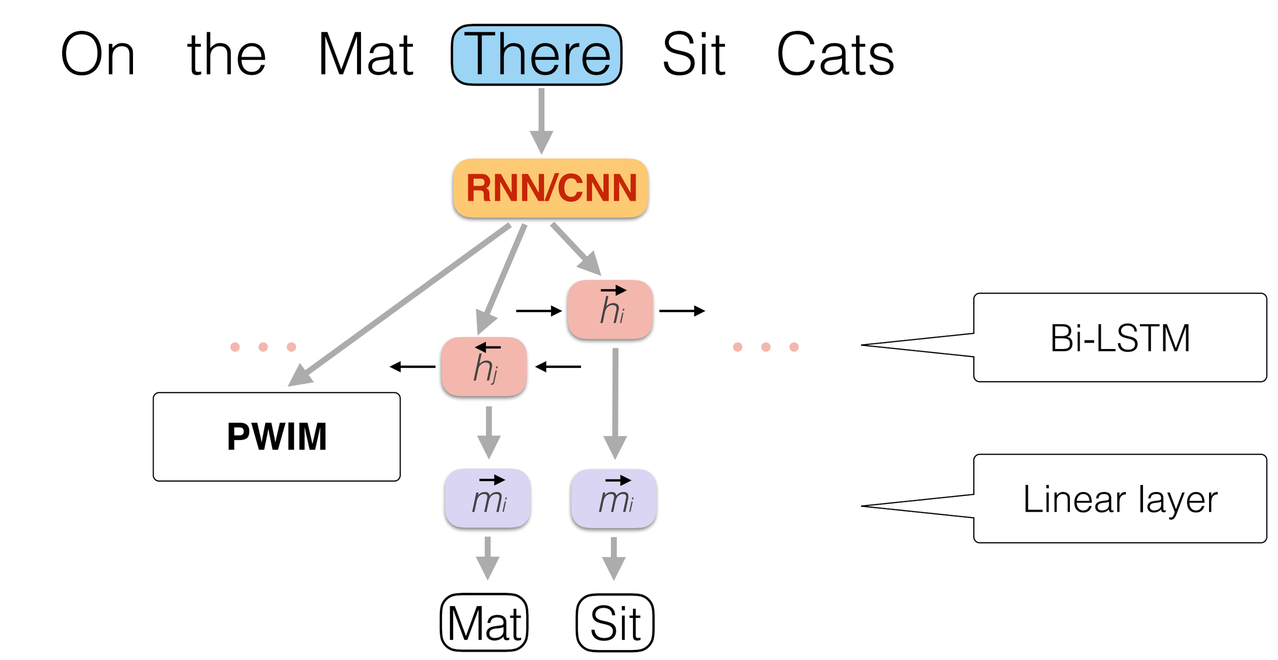
Source Code

https://github.com/lanwuwei/Subword-PWIM

See Also

- Wuwei Lan, Siyu Qiu, Hua He, and Wei Xu. A continuously growing dataset of sentential paraphrases. In EMNLP 2017
- Wuwei Lan and Wei Xu. Neural Network Models for Paraphrase Identification, Semantic Textual Similarity, Natural Language Inference, and Question Answering. In COLING 2018

Language Modeling Objective (Multitask)

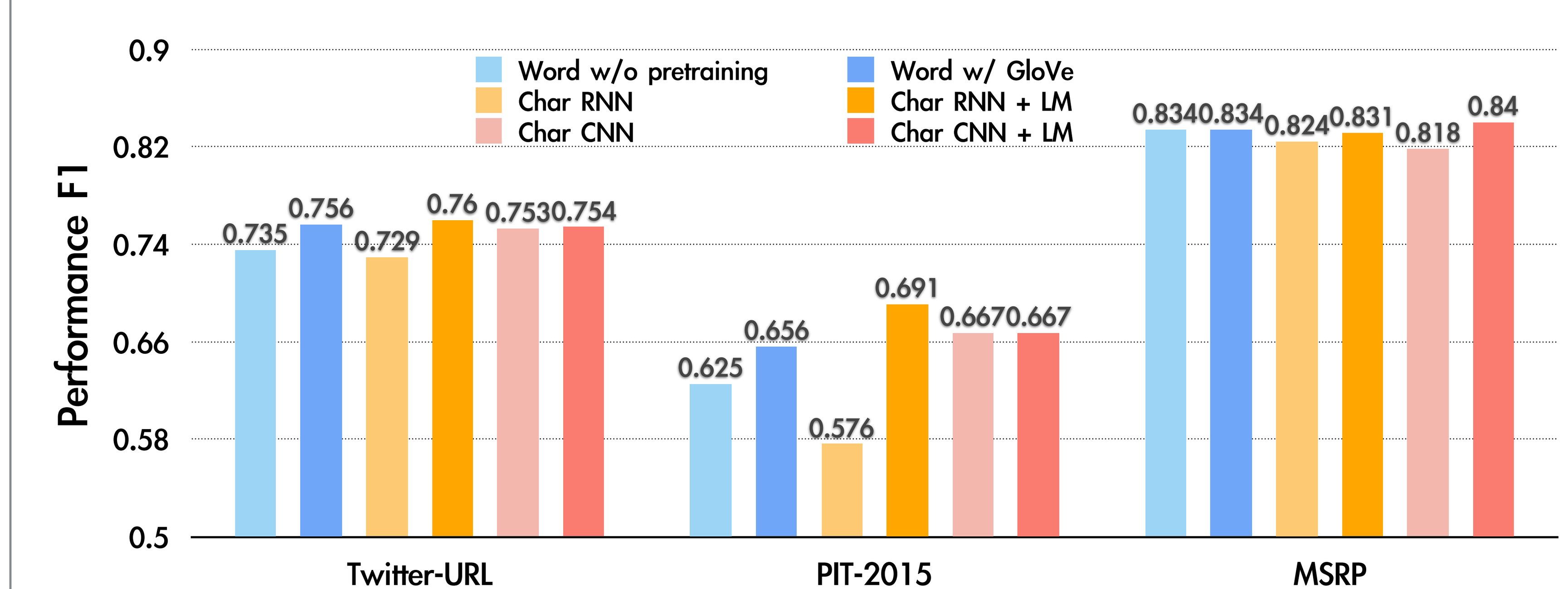


The forward LSTM predicts the **next word** and the backword LSTM predicts the **previous word.** The log-likelihood loss for both language models is added to the training objective:

$$\widetilde{E} = E + \gamma(\overrightarrow{E} + \overleftarrow{E})$$

$$\overrightarrow{E} = -\sum_{t=1}^{T-1} log(P(w_{t+1}|\overrightarrow{m}_t)) \qquad \overleftarrow{E} = -\sum_{t=2}^{T} log(P(w_{t-1}|\overleftarrow{m}_t))$$

Experiments



We performed experiments on three benchmark datasets for paraphrase identification: **Twitter URL** (social media/news, OOV ratio 31.5%), **PIT-2015** (social media, OOV ratio 13.7%) and **MSRP** (news, OOV ratio 9.0%).

Model	INV Words		OOV Words	
	any	walking	#airport	brexit
Word	anything anyone other	walk running dead	salomon 363 #trumpdchotel	bollocks missynistic patriarchy
Subword	analogy nay away	waffling slagging scaling	@atlairport #dojreport #macbookpro	grexit bret juliet
Subword + LM	any1 many ang	warming wagging waging	airport #airports rapport	#brexit brit ofbrexit

Nearest neighbors of word (subword, subword + LM) vectors under cosine similarity in Twitter-URL dataset.