CS 6501 Natural Language Processing

Seq2seq Models

Yangfeng Ji

October 31, 2018

Department of Computer Science
University of Virginia

UNIVERSITY of VIRGINIA | ENGINEERING

Overview

1. Variants of RNNs
2. Applications of RNNs
3. Seq2seq Models
4. Attention Mechanism
5. Discussion: RNNs and Linguistic Information

Variants of RNNs

Overview

- Bi-directional RNNs
- Stacked (*or* Multi-layer) LSTM
- Recurrent neural network grammars [Dyer et al., 2016]

Bi-directional RNNs

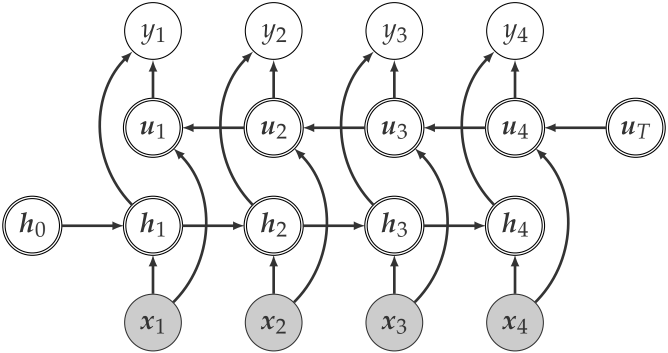
To construct a bi-directional RNN, we need another uni-directional RNN running from the end of the sequence to the beginning, as

$$\boldsymbol{u}_t = f(\boldsymbol{x}_t, \boldsymbol{u}_{t+1}). \tag{1}$$

where $\boldsymbol{u}_t$ is the hidden state at time t in this new model.

[Schuster and Paliwal, 1997]

Bi-directional RNNs (Cont.)



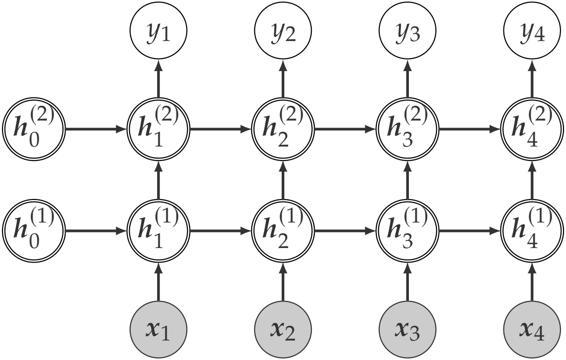
Stacked LSTM

Use the hidden state $\boldsymbol{h}_t^{(k)}$ from the current layer as input $\boldsymbol{x}_t^{(k+1)}$ to the next layer [Sutskever et al., 2014],

$$\boldsymbol{x}_t^{(k+1)} = \boldsymbol{h}_t^{(k)}. \tag{2}$$

[Sutskever et al., 2014]

Stacked LSTM (Cont.)



Applications of RNNs

Applications

- Language modeling
- POS tagging
- Named entity recognition
- Code switch
- Speech recognition
- …

NER

Example

`[Atlantis]`$_{\text{MSIC}}$ `touched down at [Kennedy Space Center]`$_{\text{LOC}}$

Category

- Person
- Location
- Organization
- Msic

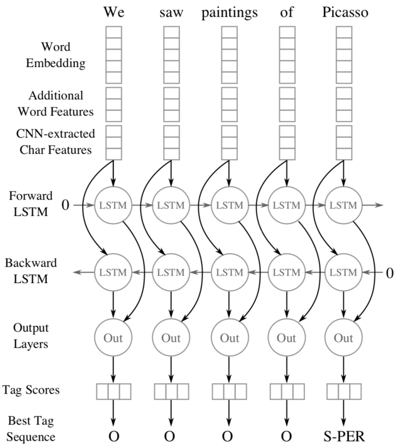
Tag set

- B: beginning
- I: inside
- O: outside

Atlantis	touched	down	at	Kennedy	Space	Center	.
B_{MSIC}	O	O	O	B_{LOC}	I_{LOC}	I_{LOC}	O

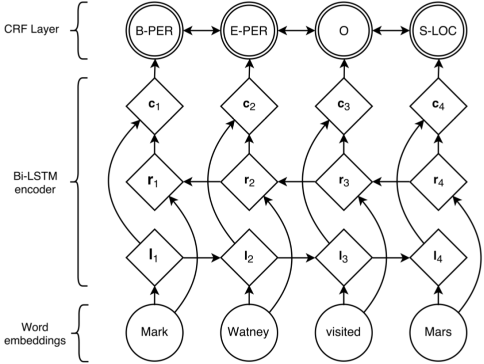
NER (Cont.)

As classification



NER (Cont.)

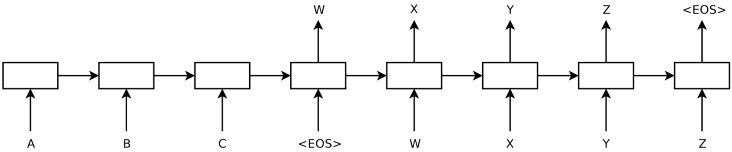
As sequence labeling



Seq2seq Models

Seq2seq Models

- Input: `ABC`
- Output: `WXYZ`



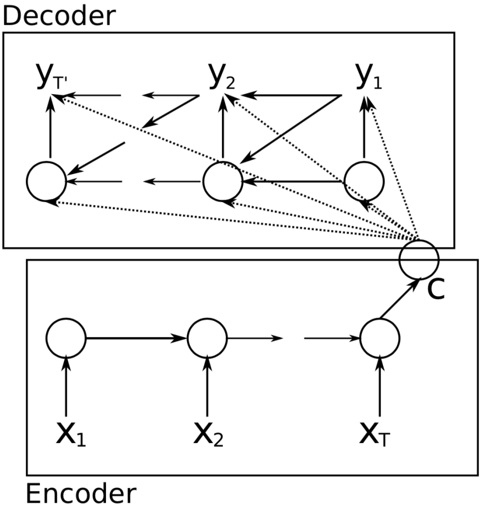
[Sutskever et al., 2014]

Tricks

- two different LSTMs: one for input sequence and the other for output sequence
- stacked (*or* deep) LSTM with four layers, which has much more potential than single-layer LSTMs
- it can greatly improve the performance on the output side, by reversing the order of input sequence

[Sutskever et al., 2014]

RNN Encoder-decoder Models



[Cho et al., 2014]

RNN Encoder-decoder Models (Cont.)

The hidden states on the output side (the decoder, as defined in [Cho et al., 2014]) are computed as

$$\boldsymbol{h}_t^{(o)} = f(\boldsymbol{h}_{t-1}^{(o)}, \boldsymbol{y}_t, \boldsymbol{c}) \tag{3}$$

where $\boldsymbol{y}_t$ is the input to the decoder size and f could be any nonlinear transition function *similar* to LSTM and GRU.

[Cho et al., 2014]

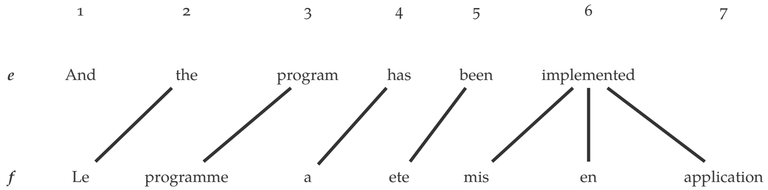
RNNs for Machine Translation

- Use seq2seq model evaluation scores to re-rank the k-best list [Sutskever et al., 2014]

RNNs for Machine Translation

- Use seq2seq model evaluation scores to re-rank the k-best list [Sutskever et al., 2014]
- Use encoder-decoder evaluation scores as additional features in the translation model [Cho et al., 2014]

Alignment in Statistical MT



$$
\begin{aligned}
P(\boldsymbol{f} \mid \boldsymbol{a}, \boldsymbol{e}) = & \, t(\text{Le} \mid \text{the}) \cdot t(\text{programme} \mid \text{program}) \cdot \\
& t(\text{a} \mid \text{has}) \cdot t(\text{ete} \mid \text{been}) \cdot \\
& t(\text{mis} \mid \text{implemented}) \cdot t(\text{en} \mid \text{implemented}) \cdot \\
& t(\text{application} \mid \text{implemented})
\end{aligned}
$$

Attention Mechanism

Attention Mechanism

With the attention mechanism [Bahdanau et al., 2015], the hidden states in the decoder are computed as

$$\boldsymbol{h}_t^{(o)} = f(\boldsymbol{h}_{t-1}^{(o)}, \boldsymbol{y}_t, \boldsymbol{c}_t). \tag{4}$$

The only difference between Eq. 4 and Eq. 3 is here $\boldsymbol{c}_t$ is changing over time.

Attention Mechanism (Cont.)

At each timestep t on the decoder side, $\boldsymbol{c}_t$ is defined as

$$\boldsymbol{c}_t = \sum_{j=1}^{T_o} \alpha_{tj} \boldsymbol{h}_j^{(i)}, \tag{5}$$

where $\{\boldsymbol{h}_j^{(i)}\}_{j=1}^{T_i}$ are the hidden states from the encoder and α_{tj} is the attention weight between the t-th token from the decoder and the j-th token from the encoder

Attention Weights

$$\alpha_{tj} = \frac{\exp(a(\boldsymbol{h}^{(o)}_{t-1}, \boldsymbol{h}^{(i)}_{j}))}{\sum_{j'=1}^{T_i} \exp(a(\boldsymbol{h}^{(o)}_{t-1}, \boldsymbol{h}^{(i)}_{j'}))}. \tag{6}$$

Attention Weights

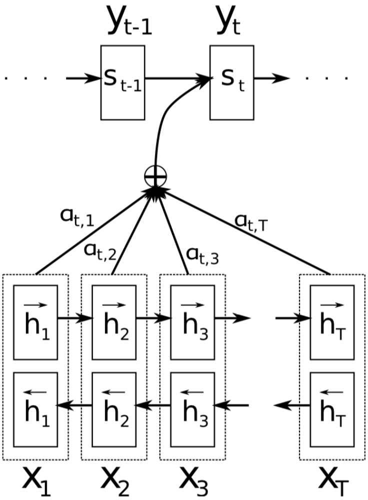
$$\alpha_{tj} = \frac{\exp(a(\boldsymbol{h}^{(o)}_{t-1}, \boldsymbol{h}^{(i)}_{j}))}{\sum_{j'=1}^{T_i} \exp(a(\boldsymbol{h}^{(o)}_{t-1}, \boldsymbol{h}^{(i)}_{j'}))}. \tag{6}$$

In [Bahdanau et al., 2015], $a(\boldsymbol{h}^{(o)}_{t-1}, \boldsymbol{h}^{(i)}_{j})$ is specifically defined as

$$a(\boldsymbol{h}^{(o)}_{t-1}, \boldsymbol{h}^{(i)}_{j}) = \boldsymbol{v}_a^{\top} \tanh(\mathbf{W}_{ao}\boldsymbol{h}^{(o)}_{t-1} + \mathbf{W}_{ai}\boldsymbol{h}^{(i)}_{j}) \tag{7}$$

with parameters $\mathbf{W}_{ao}, \mathbf{W}_{ai}$ and $\boldsymbol{v}_a$.

Network Architecture



[Bahdanau et al., 2015]

Another Option about Attention

$a(\boldsymbol{h}_{t-1}^{(o)}, \boldsymbol{h}_j^{(i)})$ can also be defined as bilinear product as

$$a(\boldsymbol{h}_{t-1}^{(o)}, \boldsymbol{h}_j^{(i)}) \propto \exp((\boldsymbol{h}_{t-1}^{(o)})^\top \mathbf{W}_a \boldsymbol{h}_j^{(i)}) \tag{8}$$

to capture the correlation between input and output, which is missed in Eq. 7.

For example [Ji et al., 2017]

Application: Sentence Summarization

AFL star blames vomiting cat for speeding

Adelaide Crows defender Daniel Talia has kept his driving license, telling a court he was speeding 36km over the limit because he was distracted by his sick cat.

The 22-year-old AFL star, who drove 96km/h in a 60km/h road works zone on the South Eastern expressway in February, said he didn't see the reduced speed sign because he was so distracted by his cat vomiting violently in the back seat of his car.

In the Adelaide magistrates court on Wednesday, Magistrate Bob Harrap fined Talia $824 for exceeding the speed limit by more than 30km/h.

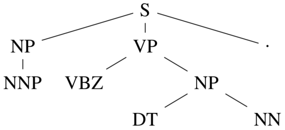
He lost four demerit points, instead of seven, because of his significant training commitments.

- *Adelaide Crows defender Daniel Talia admits to speeding but says he didn't see road signs because his cat was vomiting in his car.*
- *22-year-old Talia was fined $824 and four demerit points, instead of seven, because of his 'significant' training commitments.*

[Cheng and Lapata, 2016]

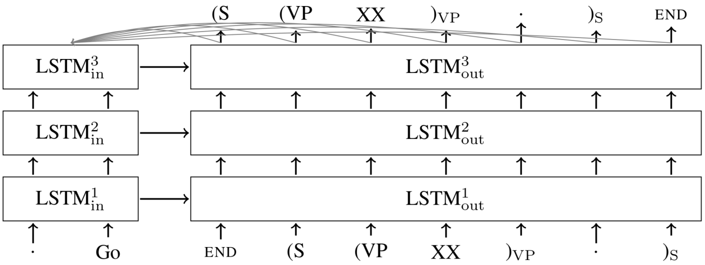
Application: Syntactic Parsing

John has a dog . $\rightarrow$



John has a dog . $\rightarrow$ (S (NP NNP $)_{\mathrm{NP}}$ (VP VBZ (NP DT NN $)_{\mathrm{NP}}$ $)_{\mathrm{VP}}$. $)_{\mathrm{S}}$

Application: Syntactic Parsing (Cont.)



Discussion: RNNs and Linguistic Information

Questions

- to what extent, RNNs can capture long-term contextual information in texts?
- how much syntactic information RNN can learn?

About Long-term Dependency

Experiments:

- shuffle, replace, or drop words in preceding text
- increase on perplexity
- only at test time

Observations

- LSTMs can capture long-term dependency upto 200 tokens
- beyond about 50 tokens, LSTMs are not sensitive to order information anymore

[?]

About learning syntax

A task of testing subject-verb agreement is called the *number prediction* task. As described in [], for a given half sentence,

```
The keys to the cabinet _______
```

the model is asked to guess the number of the following verb (not the verb itself). This is a binary choice, the answer could be either PLURAL or SINGULAR.

[Linzen et al., 2016]

Examples

Easy cases:

(a) The **key is** on the table.

(b) The **keys are** on the table.

Harder cases:

(c) The **keys** to the cabinet **are** on the table.

(d) The **building** on the far right that's quite old and run down is the Kilgore Bank Building.

[Linzen et al., 2016]

Examples

Tricky cases:

(e) Alluvial **soils** carried in the *floodwaters* **add** nutrients to the floodplains.

(f) Yet the **ratio** of men who survive to the women and children who survive **is** not clear in this story.

[Linzen et al., 2016]

Observations

- [Linzen et al., 2016]
  - LSTM achieve a really high overall accuracy
  - performance will drop when sequential and structural information conflicted
  - *stronger* architectures may be required to further reduce errors

Observations

- [Linzen et al., 2016]
  - LSTM achieve a really high overall accuracy
  - performance will drop when sequential and structural information conflicted
  - *stronger* architectures may be required to further reduce errors
- [Kuncoro et al., 2018]

**LSTMs Can Learn Syntax-Sensitive Dependencies Well,
But Modeling Structure Makes Them Better**

Summary

1. Variants of RNNs
2. Applications of RNNs
3. Seq2seq Models
4. Attention Mechanism
5. Discussion: RNNs and Linguistic Information

Reference

Bahdanau, D., Cho, K., and Bengio, Y. (2015).
Neural machine translation by jointly learning to align and translate.
In *ICLR*.

Cheng, J. and Lapata, M. (2016).
Neural summarization by extracting sentences and words.
arXiv preprint arXiv:1603.07252.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014).
Learning phrase representations using rnn encoder-decoder for statistical machine translation.
arXiv preprint arXiv:1406.1078.

Dyer, C., Kuncoro, A., Ballesteros, M., and Smith, N. A. (2016).
Recurrent neural network grammars.
arXiv preprint arXiv:1602.07776.

Ji, Y., Tan, C., Martschat, S., Choi, Y., and Smith, N. A. (2017).
Dynamic entity representations in neural language models.
arXiv preprint arXiv:1708.00781.

Kuncoro, A., Dyer, C., Hale, J., Yogatama, D., Clark, S., and Blunsom, P. (2018).
Lstms can learn syntax-sensitive dependencies well, but modeling structure makes them better.
In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1426–1436.

Linzen, T., Dupoux, E., and Goldberg, Y. (2016).
Assessing the ability of lstms to learn syntax-sensitive dependencies.
arXiv preprint arXiv:1611.01368.

Schuster, M. and Paliwal, K. K. (1997).
Bidirectional recurrent neural networks.
IEEE Transactions on Signal Processing