Sequence Modeling via Segmentations

Presenter: Ceyer Wakilpoor

Chong Wang[1] Yining Wang*[2] Abdelrahman Mohamed*[3]
Dengyong Zhou[1] Li Deng*[4]

[1]Microsoft Research, [2]Carnegie Mellon University

[3]Amazon, [4]Citadel Securities LLC.

*Work performed when the authors were with Microsoft

ICML, 2017

Outline

1. Introduction
2. Mathematical Model
   - Mapping from non-sequence to sequence
   - Mapping from sequence to sequence
   - Carrying over information across segments
3. Forward, backward and decoding
   - Forward and backward propagations
4. Forward, backward and decoding
5. Experiments

Outline

1 Introduction

2 Mathematical Model
- Mapping from non-sequence to sequence
- Mapping from sequence to sequence
- Carrying over information across segments

3 Forward, backward and decoding
- Forward and backward propagations

4 Forward, backward and decoding

5 Experiments

Introduction

- Segmental structure can be found in many types of sequencing, examples:
  - Phase structure: "Machine Learning is part of artificial intelligence" $\rightarrow$ [Machine Learning] [is] [part of] [artificial intelligence]
  - Phonotactic rules: "thought" $\rightarrow$ [th][ou][ght]
- input can be sequence or not, but output is always a sequence
- Constructing condition probability $p(y|x)$
  - When non-sequence is given, sum over probabilities of all valid segmentations
  - When input is sequence, sum over all feasible segmentations

Outline

1 Introduction

2 Mathematical Model
- Mapping from non-sequence to sequence
- Mapping from sequence to sequence
- Carrying over information across segments

3 Forward, backward and decoding
- Forward and backward propagations

4 Forward, backward and decoding

5 Experiments

Mapping from non-sequence to sequence

- Some variables
  - for any a $_{1:\tau_\alpha} \in \mathbb{S}_y, \pi(a_{1:\tau_\alpha}) = y_{1:T}$
  - x: fixed length vector
  - T: number of outputs, length of whole sequence
  - τ: number of segments
  - \$: end of segment token
  - $\pi()$ is the concatenation operator
  - $\mathbb{S}_y$: all possible segmentations
- $|\mathbb{S}_y|$ is exponentially large

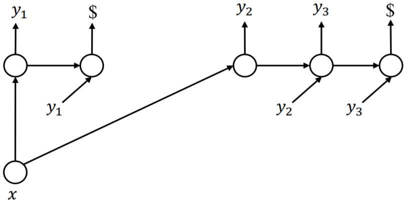


Figure 1. For Section 2.1. Given output $y_{1:3}$ and its segmentation $a_1 = \{y_1, \$\}$ and $a_2 = \{y_2, y_3, \$\}$, input x controls the initial states of both segments. Note that $\pi(a_{1:t-1})$ is omitted here.

Mapping from non-sequence to sequence

- Sum over all possible segementations, and multiply probability over previous segments

$$p(y_{1:T}|x) \triangleq \sum_{a_{1:\tau_a} \in \mathcal{S}_y} p(a_{1:\tau_a}|x)$$
$$= \sum_{a_{1:\tau_a} \in \mathcal{S}_y} \prod_{t=1}^{\tau_a} p(a_t|x, \pi(a_{1:t-1})),$$

- The segmentation probability ($p(a_t|x, \pi(a_{1:t-1})$) is modeled with an RNN with a softmax probability function
- Each segment's RNN will share the same weights, which means all timesteps are treated the same
- However, even with the segmentation probability, there is still an exponential number of possibilities, the RNN simply gives us the score for each possibility, this will come later

Mapping from sequence to sequence

- Monotonic alignment between input, which is now $x_{1:T'}$
- Each element x_t emits one segment, a_t, which is concatenated to $\pi(a_{1:T})$
- Unlike before, empty segments are allowed in the emission, which means y will always include T' segments, but there can be empty ones

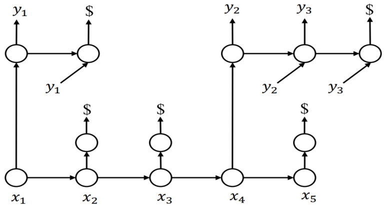


Figure 2. For Section 2.2. SWAN emits one particular segmentation of $y_{1:T}$ with x_1 waking (emits y_1) and x_4 waking (emits y_2 and y_3) while x_2, x_3 and x_5 sleeping. SWAN needs to consider all valid segmentations like this for $y_{1:T}$.

Mapping from sequence to sequence

- We can model the same probability, except now conditional on a specific element of $x_{1LT'}$ as opposed to single input vector, x

$$p(y_{1:T}|x_{1:T'}) \triangleq \sum_{a_{1:T'} \in S_y} \prod_{t=1}^{T'} p(a_t|x_t, \pi(a_{1:t-1}))$$

- This condition on one element of x increases the number of possible segments, $|\mathbb{S}_y| \in O(T'T^2)$, again can't brute force the inference

Carrying over information across segments

- The previous RNN models how to construct each segment based on what was constructed before (can lead to greedy solution); we also need to model dependence between all segments
- Use sequence transducer approach to create separate RNN to model

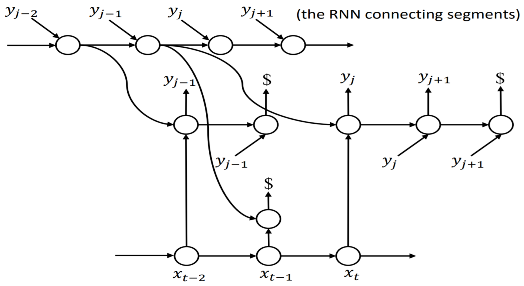


Figure 3. For Section 2.3. SWAN carries over information across segments using a separate RNN. Here the segments are $a_{t-2} = \{y_{j-1}, \$\}$, $a_{t-1} = \{\$\}$ and $a_t = \{y_j, y_{j+1}, \$\}$ emitted by input elements x_{t-2}, x_{t-1} and x_t respectively.

Outline

1 Introduction

2 Mathematical Model
- Mapping from non-sequence to sequence
- Mapping from sequence to sequence
- Carrying over information across segments

3 Forward, backward and decoding
- Forward and backward propagations

4 Forward, backward and decoding

5 Experiments

Forward and backward propagations

- Want to model forward and backward probabilities, like a bidirectional RNN
- Probability that input $x_{1:t}$ emits output $y_{1:j}$

$$\alpha_t(j) = p(y_{1:j}|x_{1:t}),$$

- Probability that $x_{t+1:T'}$ emits output $y_{j+1:T}$

$$\beta_t(j) = p(y_{j+1,T}|x_{t+1:T'}, y_{1:j})$$

- This allows the model to consider all segments in x, even those ahead of what's being considered at the time step:

$$p(y_{1:T}x_{1:T'}) = \sum_{j=0}^{T} \alpha_t(j)\beta_t(j)$$

Outline

1 Introduction

2 Mathematical Model
- Mapping from non-sequence to sequence
- Mapping from sequence to sequence
- Carrying over information across segments

3 Forward, backward and decoding
- Forward and backward propagations

4 Forward, backward and decoding

5 Experiments

Forward

- Dynamic Programming problem:

$$\alpha_t(j) = \sum_{j'=0}^{j} \alpha_{t-1}(j') p(y_{j'+1:j} | x_t),$$

$$\beta_t(j) = \sum_{j'=j}^{T} \beta_{t+1}(j') p(y_{j+1:j'} | x_{t+1})$$

Backward

- Gradient w.r.t x_t, similar derivation can be done for when x is not a sequence (x)

$$\frac{\partial logp(y_{1:T}|x_{1:T'})}{\partial x_t} = \sum_{j'=0}^{T}\sum_{j=0}^{j'} w_t(j,j')\frac{\partial logp(y_{j+1:j'}|x_t)}{\partial x_t}$$

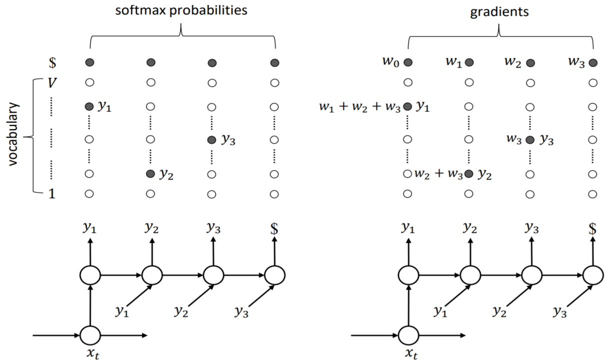
$$w_t(j,j') \triangleq \alpha_{t-1}(j)\beta_t(j')\frac{p(y_{j+1:j'}|x_t)}{p(y_{1:T}|x_{1:T'})}$$

Speed ups

- Limit maximum segment to be L, reduces complexity to $O(T'TL^2)$, since fewer dependencies have to be modeled
- The computation for the longer segments will usually compute terms that are needed for the shorter ones as well, just need to assign the proper weights from the shorter segments to apply to terms found by the longer segment

Beam search decoding

- Effectively performing a simple left-to right beam search for each x_t



(a) Forward pass (b) Backward pass

Outline

1 Introduction

2 Mathematical Model
- Mapping from non-sequence to sequence
- Mapping from sequence to sequence
- Carrying over information across segments

3 Forward, backward and decoding
- Forward and backward propagations

4 Forward, backward and decoding

5 Experiments

Experiments

- Automatically segment text
- θ controls the distribution of the words
- Stop words are removed and punctuation is a known segment boundary
- 2-layer feed-forward neural network with ReLU nonlinearity
- Two-layer GRU used to model $p(y_{1:T}|W\theta(\xi))$

Experiments

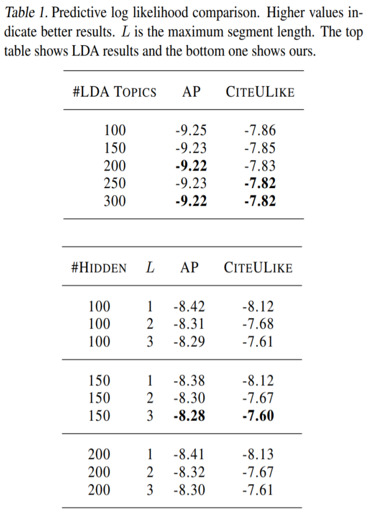


Table 1. Predictive log likelihood comparison. Higher values indicate better results. L is the maximum segment length. The top table shows LDA results and the bottom one shows ours.

#LDA TOPICS	AP	CITEULIKE
100	-9.25	-7.86
150	-9.23	-7.85
200	**-9.22**	-7.83
250	-9.23	**-7.82**
300	**-9.22**	**-7.82**

#HIDDEN	L	AP	CITEULIKE
100	1	-8.42	-8.12
100	2	-8.31	-7.68
100	3	-8.29	-7.61
150	1	-8.38	-8.12
150	2	-8.30	-7.67
150	3	**-8.28**	**-7.60**
200	1	-8.41	-8.13
200	2	-8.32	-7.67
200	3	-8.30	-7.61

Experiments

- Speech recognition : Character Error Rate of 30.5% compared to Baidu's CTC error rate of 31.8%

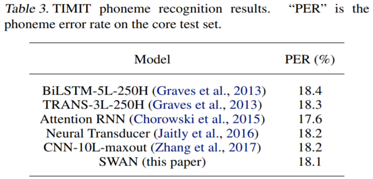


Table 3. TIMIT phoneme recognition results. "PER" is the phoneme error rate on the core test set.

Model	PER (%)
BiLSTM-5L-250H (Graves et al., 2013)	18.4
TRANS-3L-250H (Graves et al., 2013)	18.3
Attention RNN (Chorowski et al., 2015)	17.6
Neural Transducer (Jaitly et al., 2016)	18.2
CNN-10L-maxout (Zhang et al., 2017)	18.2
SWAN (this paper)	18.1

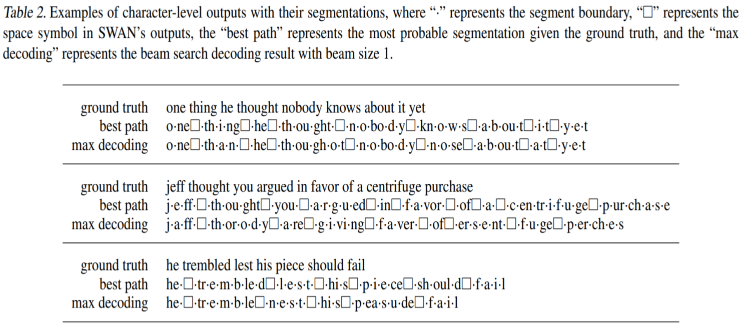


Table 2. Examples of character-level outputs with their segmentations, where "·" represents the segment boundary, "□" represents the space symbol in SWAN's outputs, the "best path" represents the most probable segmentation given the ground truth, and the "max decoding" represents the beam search decoding result with beam size 1.

ground truth	one thing he thought nobody knows about it yet
best path	o·ne□·th·i·ng□·he□·th·ou·ght·□·n·o·bo·d·y□·kn·o·w·s□·a·b·ou·t□·i·t□·y·e·t
max decoding	o·ne□·th·a·n·□·he□·th·ou·gh·o·t□·n·o·bo·d·y□·n·o·se□·a·b·ou·t□·a·t□·y·e·t
ground truth	jeff thought you argued in favor of a centrifuge purchase
best path	j·e·ff·□·th·ou·ght□·you·□·a·r·g·u·ed□·in□·f·a·vor·□·of□·a·□·c·en·tr·i·f·u·ge□·p·ur·ch·a·s·e
max decoding	j·a·ff·□·th·or·o·d·y□·a·re□·g·i·vi·ng□·f·a·ver·□·of□·er·s·e·nt·□·f·u·ge□·p·er·ch·e·s
ground truth	he trembled lest his piece should fail
best path	he·□·tr·e·m·b·le·d□·l·e·s·t·□·hi·s□·p·i·e·ce□·sh·oul·d□·f·a·i·l
max decoding	he·□·tr·e·m·b·le□·n·e·s·t·□·hi·s□·p·ea·s·u·de□·f·a·i·l

Future Work

They have presented a new probability distribution for sequence modeling in a very generalizable framework, and they tested it in two domains