Sentences Generation

陈云川

2016.05.11

Outline

- Hongyu Guo. Generating Text with Deep Reinforcement Learning. arXiv 2015.
- Alex Graves *et al*. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. ICML 2006.
    - Alex Graves. Supervised sequence labelling. Springer Berlin Heidelberg, 2012.

Motivation

- Lots of applications rely on text generation
  - speech recognition
  - machine translation
  - text rephrasing
  - question answering

Sequence Generation Model

$$p(y_1, y_2, \ldots, y_m \mid x_1, x_2, \ldots, x_n) = \prod_{t=1}^{m} p(y_t \mid D_{t-1})$$

$$D_{t-1} = \{\boldsymbol{v}_{t-1}\}$$
$$\boldsymbol{u} = \boldsymbol{f}(x_1, x_2, \ldots, x_n)$$
$$\boldsymbol{v}_0 = \boldsymbol{I}(u)$$
$$\boldsymbol{v}_t = \boldsymbol{g}(\boldsymbol{v}_{t-1})$$

NIPS '14

Sequence to Sequence Learning with Neural Networks

Ilya Sutskever, Google, ilyasu@google.com
Oriol Vinyals, Google, vinyals@google.com
Quoc V. Le, Google, qvl@google.com

$$D_{t-1} = \{\boldsymbol{u}, \boldsymbol{v}_{t-1}\}$$
$$\boldsymbol{u} = \boldsymbol{f}(x_1, x_2, \ldots, x_n)$$
$$\boldsymbol{v}_t = \boldsymbol{g}(y_1, y_2, \ldots, y_t)$$

ICLR '15

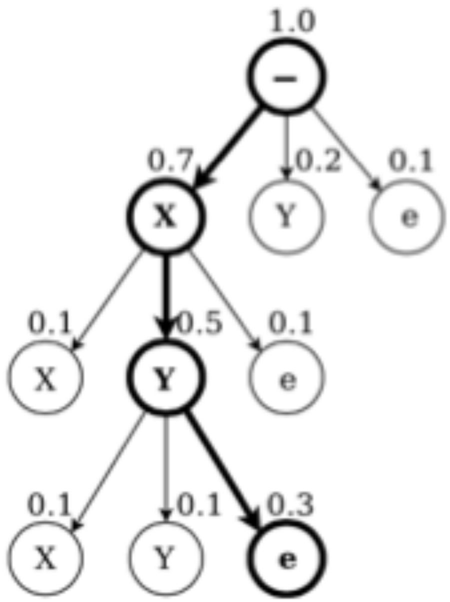
NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE

Dzmitry Bahdanau
Jacobs University Bremen, Germany

KyungHyun Cho, Yoshua Bengio*
Université de Montréal

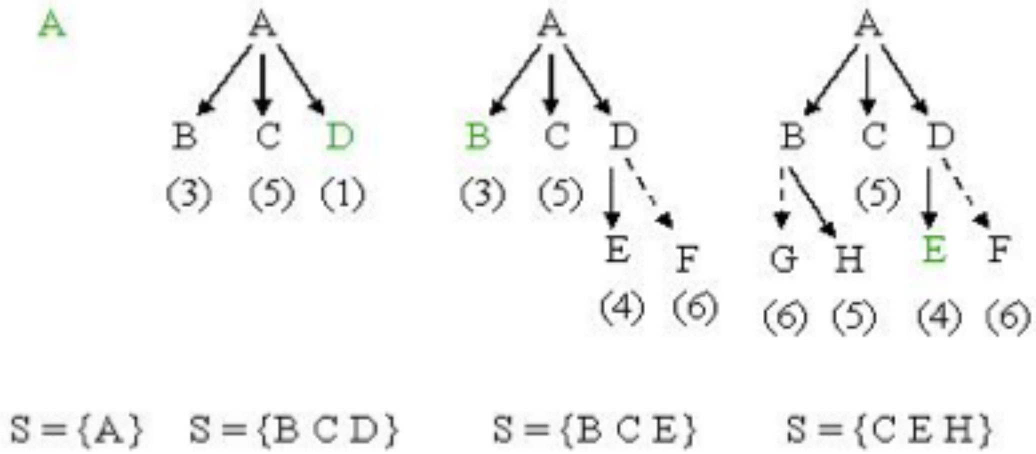
Search to Decode

- How to sample a sequence from $\prod_{t=1}^{m} p(y_t \mid D_{t-1})$?
  - Prefix search



  - ???

  - Beam search



Other Way to Decode?

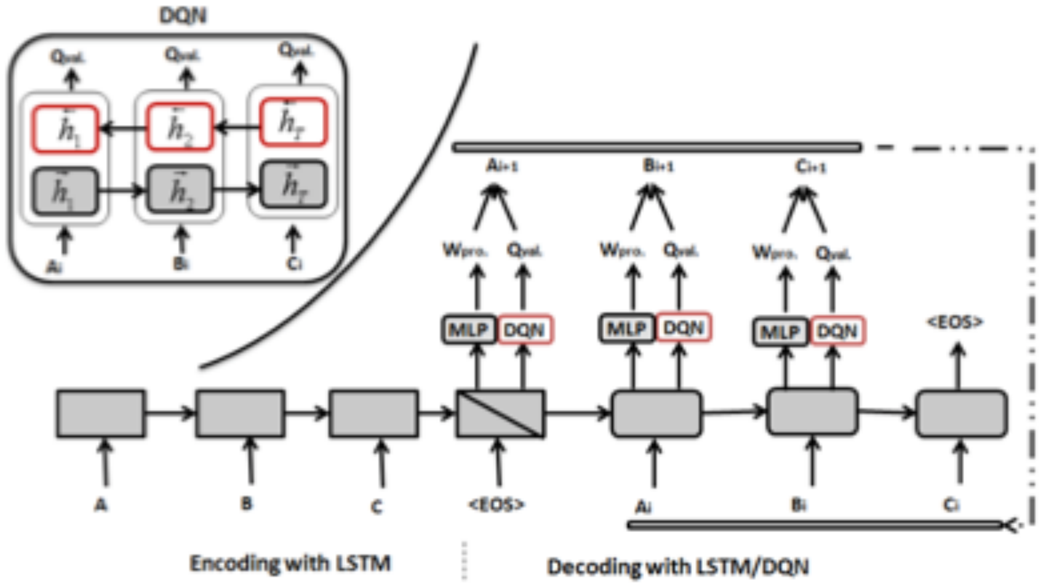
- Prefix searching is too expensive
- Not pleased with the approximate beam searching
- Is it possible to generate a coarse sequence first and then refine it iteratively?

Click here to read more than the New York Times

Click here to read more from the New York Times

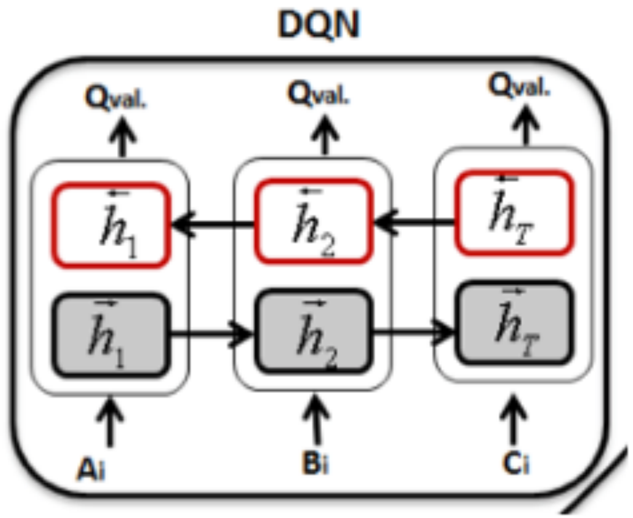
Generating Sequence with Deep Q-Network: the Model

- Generating the state Representation with LSTM
  - Encode
  - Decode
- Iteratively Decoding Sequence with Deep Q-Network (DQN)



Decoding Sequence with Deep Q-Network (DQN)

- Markov Decision Process
  - States: (EnSent, DeSent)
  - Actions: words and their positions
  - State Transition Prob.: <Deterministic>
  - Reward: BLEU score



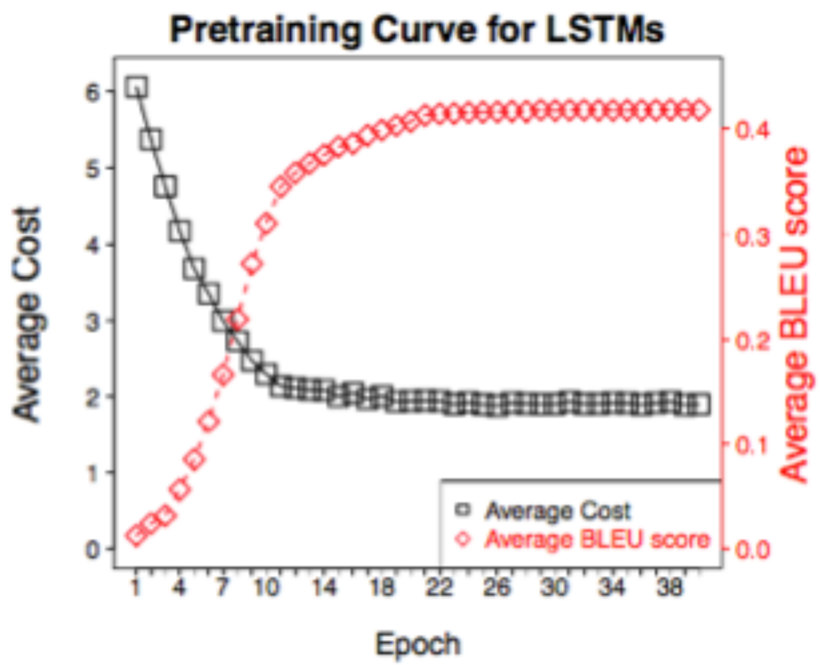
- Loss: $L_i(\boldsymbol{\theta}) = \mathbb{E}_{s,a}[(q_i - Q(s, a; \boldsymbol{\theta}_i))^2]$

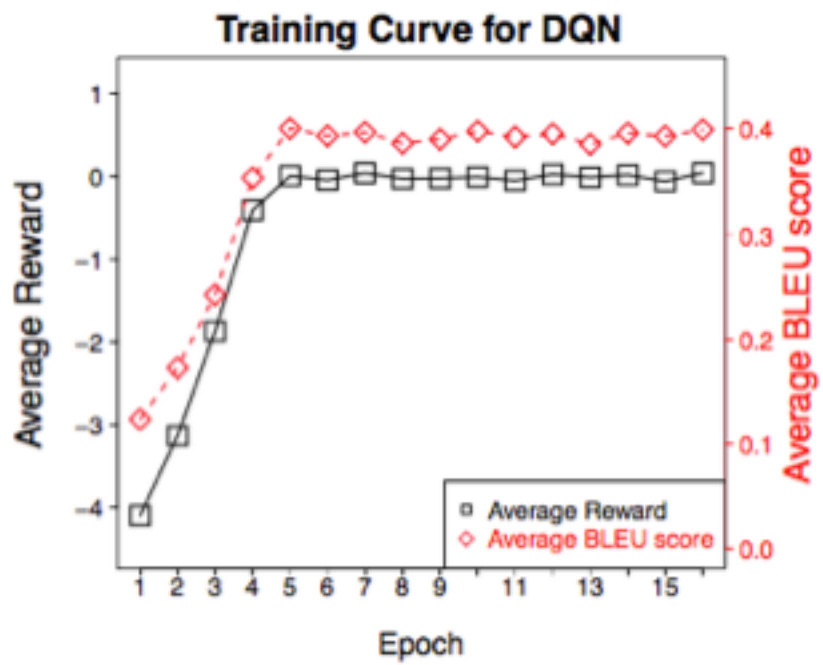
$$q_i = \mathbb{E}_{s,a}[r_i + \lambda \max_{a'} Q(s', a'; \boldsymbol{\theta}_{i-1}]$$

Empirical Observations on Model Design

- Separating Make State Generation Function from DQN
- Pre-training the State Generation Function
- Updating with Replay Memory Sampling
- Importance of Supervised Softmax Signal

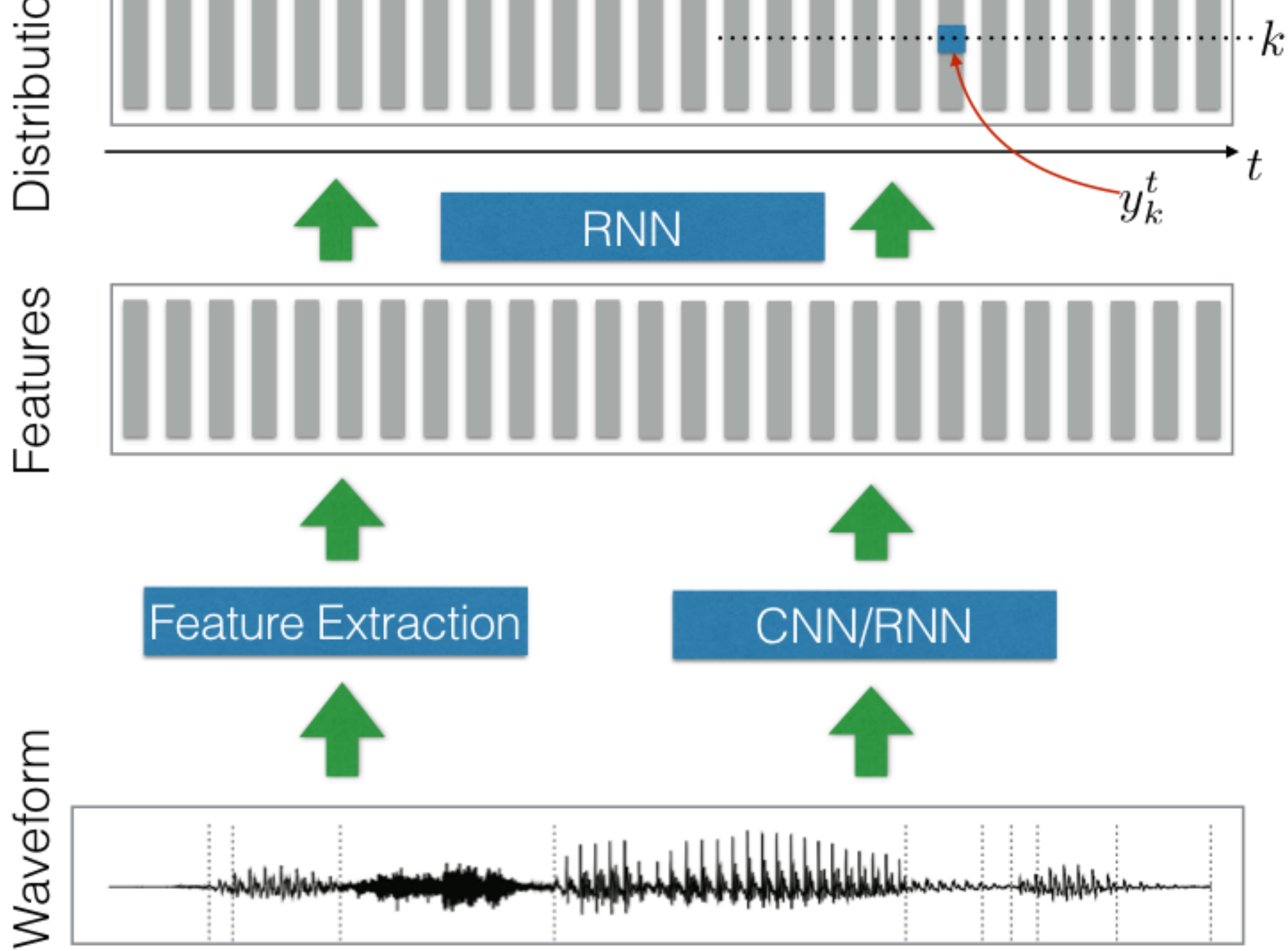
Experimental Result



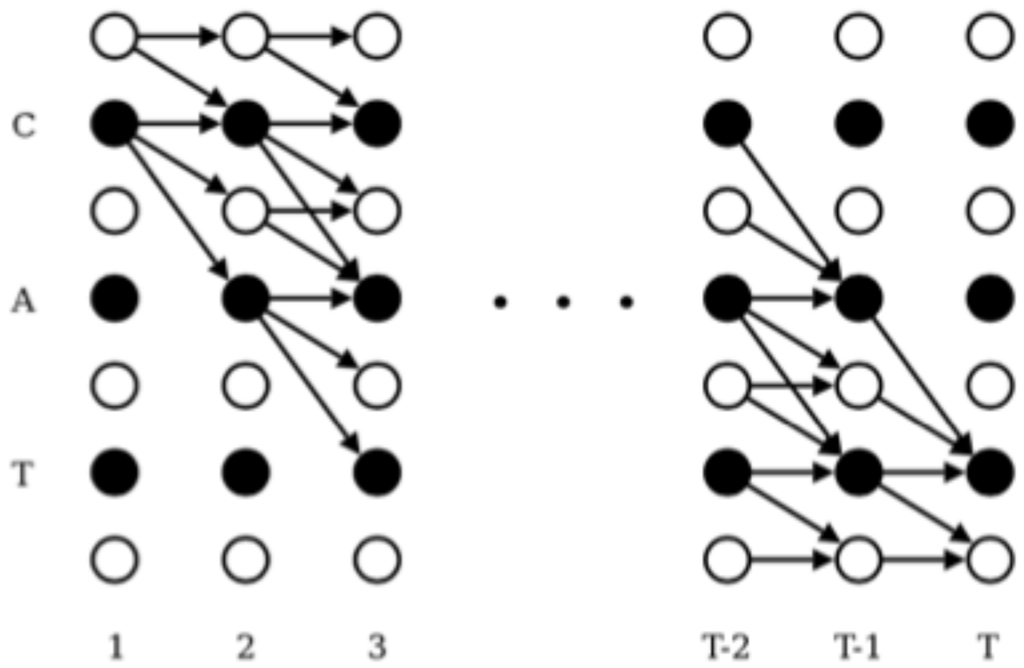


Testing systems	LSTM decoder	DQN
Average SmoothedBLEU on sentences IN the training set	0.425	0.494
Average SmoothedBLEU on sentences NOT in the training set	0.107	0.228

Connectionist temporal classification (CTC)


Distributio
k
t
y_k^t
RNN
Features
Feature Extraction
CNN/RNN
Waveform

Connectionist Temporal Classification (CTC)



$$p(\pi|\boldsymbol{x}) = \prod_{t=1}^{T} y_{\pi_t}^{t}$$

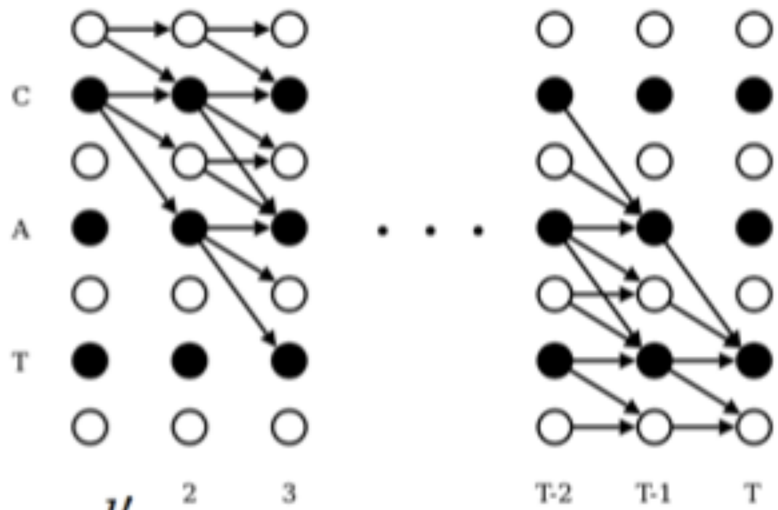
$$p(\boldsymbol{l}|\boldsymbol{x}) = \sum_{\pi \in \mathcal{B}^{-1}(\boldsymbol{l})} p(\pi|\boldsymbol{x})$$

$\mathcal{B}$ is the many-to-one map that remove all repeated symbols and blanks

$$\mathcal{B}(\emptyset aa \emptyset \emptyset abb) = aab$$

Forward Variable

- the forward variable α is the summed probability of all length t paths that are mapped by $\mathcal{B}$ onto the length $u/2$ prefix of $\boldsymbol{l}$



$$\alpha(t,u) = \sum_{\pi \in V(t,u)} \prod_{i=1}^{t} y^i_{\pi_i}$$

$$\alpha(t,u) = y^t_{l'_u} \sum_{i=f(u)}^{u} \alpha(t-1,i)$$

$$f(u) = \begin{cases} u-1 & \text{if } l'_u = \text{blank or } l'_{u-2} = l'_u \\ u-2 & \text{otherwise} \end{cases}$$

$$V(t,u) = \{\pi \in L'^t : \mathcal{B}(\pi) = \boldsymbol{l}_{1:u/2}, \pi_t = l'_u\}$$

Backword Variable

- the backward variable $\beta(t,u)$ is defined as the summed probabilities of all paths starting at $t+1$ that complete $\boldsymbol{l}$ when append to any path contributing to $\alpha(t,u)$

$$\beta(t,u) = \sum_{\pi \in W(t,u)} \prod_{i=1}^{T-t} y_{\pi_i}^{t+i}$$

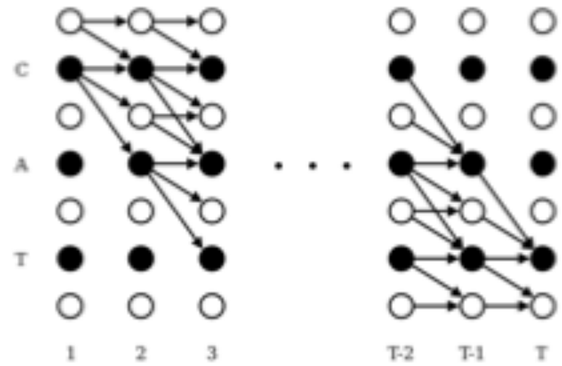
$$W(t,u) = \{\pi \in L'^{T-t} : \mathcal{B}(\hat{\pi} + \pi) = \boldsymbol{l}, \forall \hat{\pi} \in V(t,u)\}$$

Loss

$$\mathcal{L}(S) = -\ln \prod_{(\mathbf{x},\mathbf{z})\in S} p(\mathbf{z}|\mathbf{x}) = -\sum_{(\mathbf{x},\mathbf{z})\in S} \ln p(\mathbf{z}|\mathbf{x})$$

$$p(\mathbf{z}|\mathbf{x}) = \sum_{u=1}^{|\mathbf{z}'|} \alpha(t,u)\beta(t,u)$$

$$\mathcal{L}(\mathbf{x},\mathbf{z}) = -\ln \sum_{u=1}^{|\mathbf{z}'|} \alpha(t,u)\beta(t,u)$$



Loss Gradient

$$\frac{\partial \mathcal{L}(\mathbf{x},\mathbf{z})}{\partial y_k^t} = -\frac{\partial \ln p(\mathbf{z}|\mathbf{x})}{\partial y_k^t} = -\frac{1}{p(\mathbf{z}|\mathbf{x})}\frac{\partial p(\mathbf{z}|\mathbf{x})}{\partial y_k^t}$$

$$\frac{\partial p(\mathbf{z}|\mathbf{x})}{\partial y_k^t} = \frac{1}{y_k^t}\sum_{u\in B(\mathbf{z},k)} \alpha(t,u)\beta(t,u)$$

$$\because \; p(\mathbf{z}|\mathbf{x}) = \sum_{u=1}^{|\mathbf{z}'|}\alpha(t,u)\beta(t,u)$$

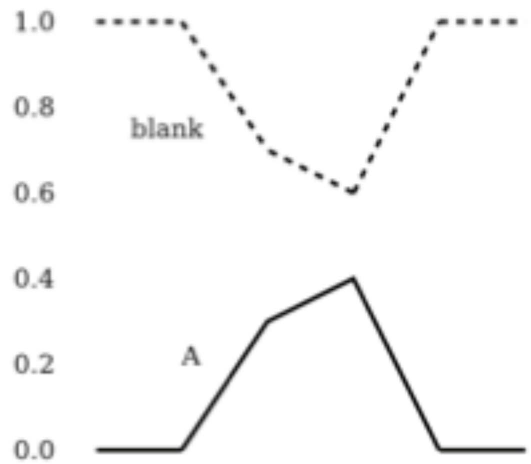
Decoding

$$\boldsymbol{l}^* = \arg\max_{\boldsymbol{l}} p(\boldsymbol{l}|\boldsymbol{x})$$

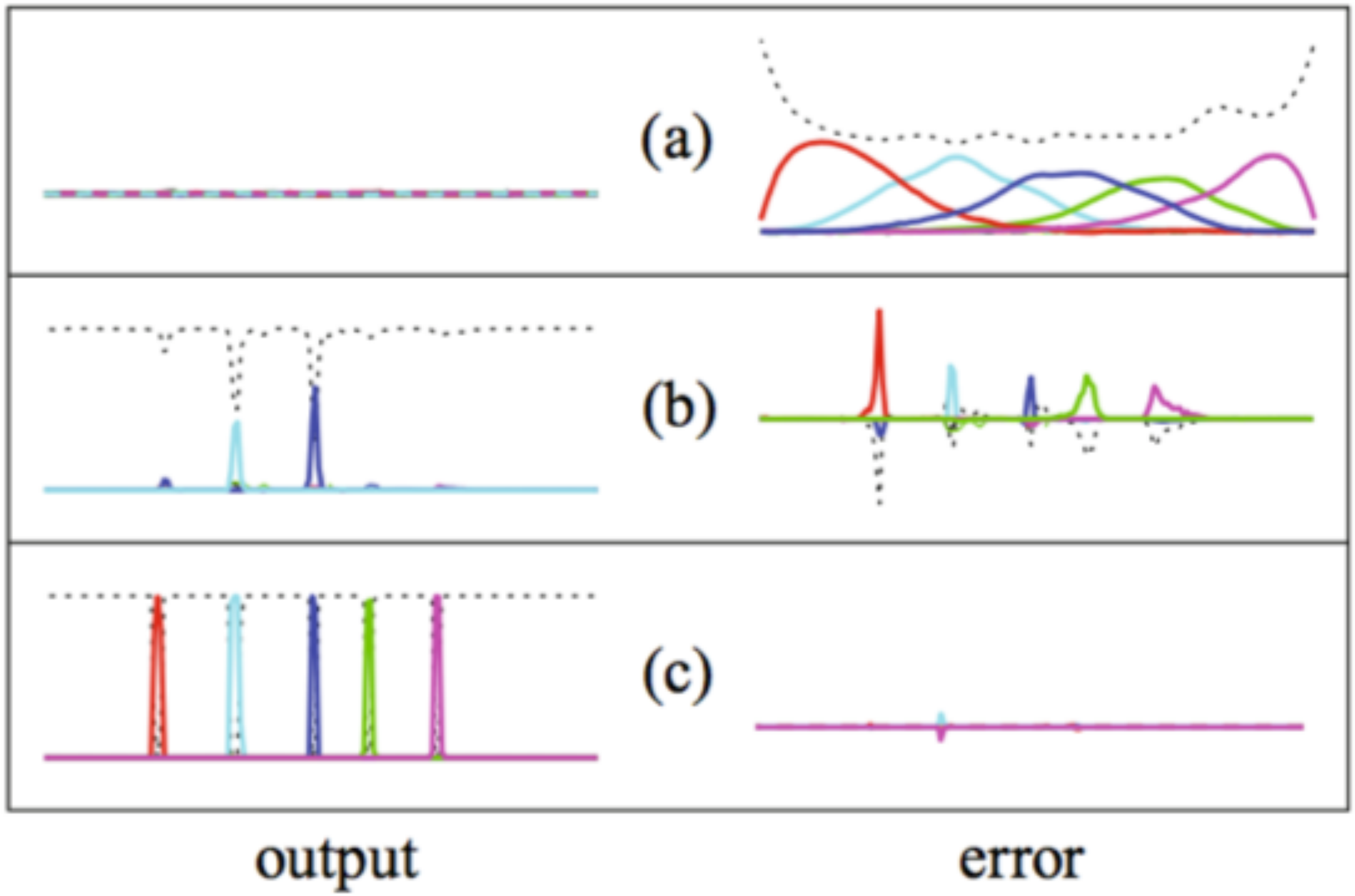
- Best Path Decoding

$$\boldsymbol{l}^* \approx \mathcal{B}(\pi^*)$$

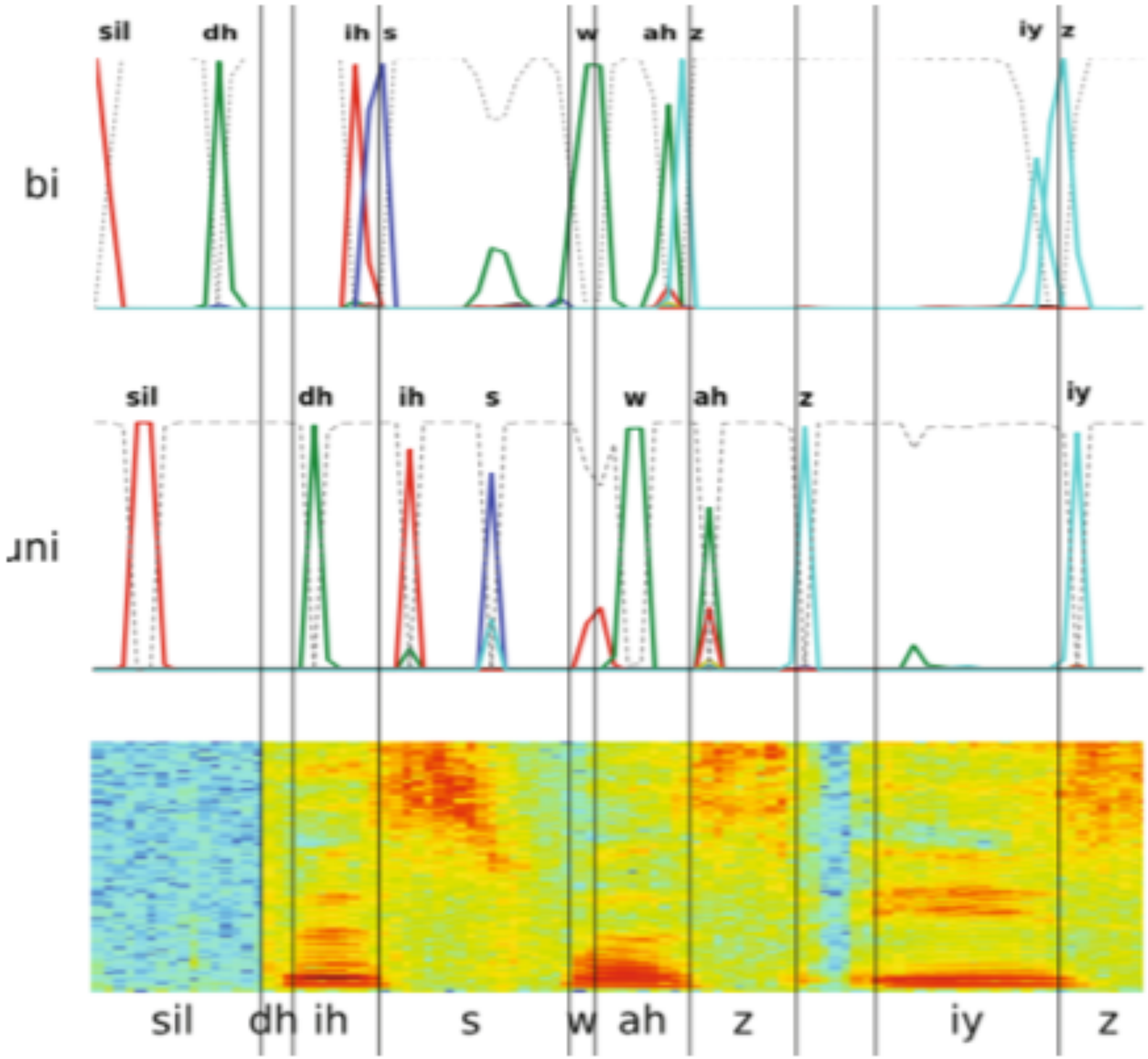
- Prefix Search Decoding



Experimental Results



Experimental Results



Tanks!