Diving into the Training for the Sequence-to-sequence Model

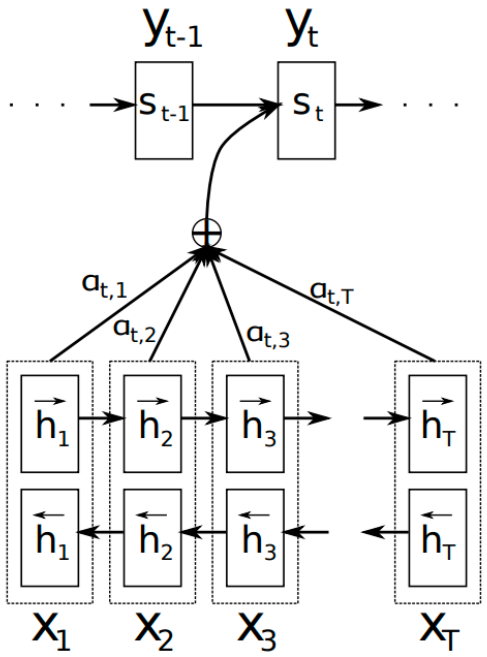
Yang Feng
Institute of Computing Technology
Chinese Academy of Sciences
2019.10.12

中国科学院计算技术研究所
INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES

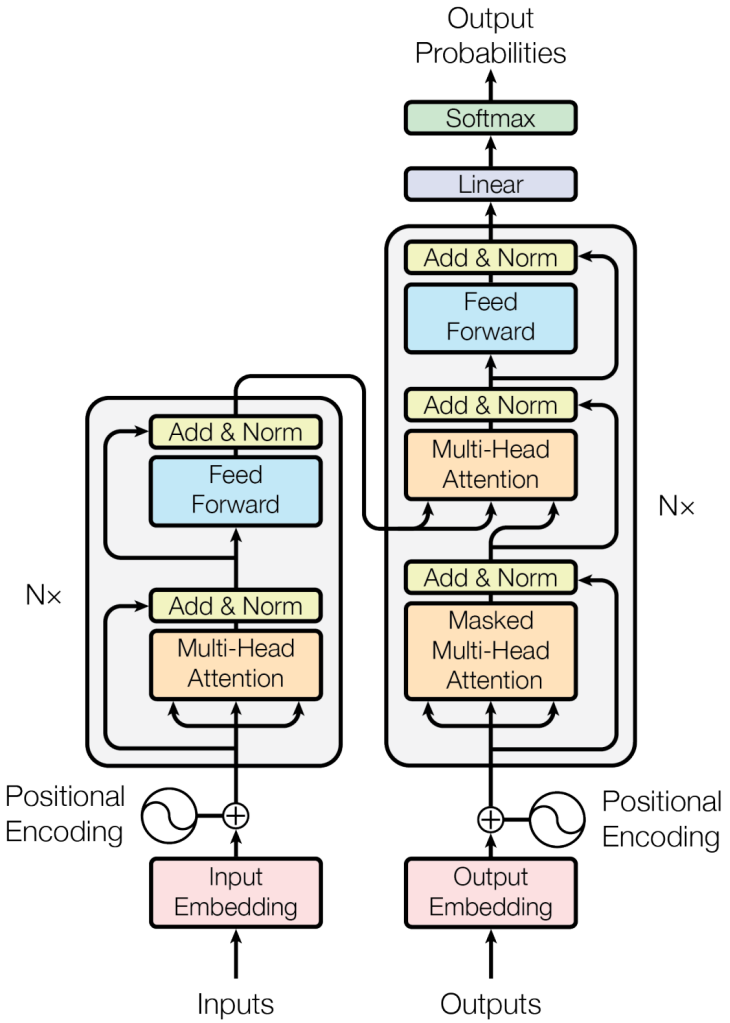
The Sequence-to-sequence Model

- Encoder-Decoder Framework
  - Encoder: encode the source into representations
  - Decoder: decode the representation into the target

Main Models



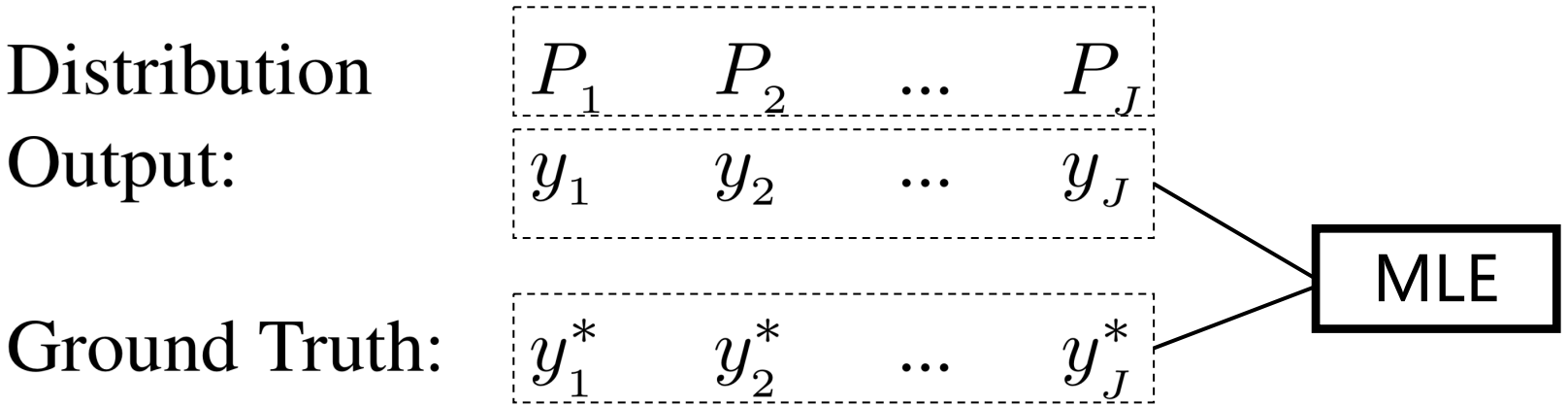
RNNSearch



Transformer

Training

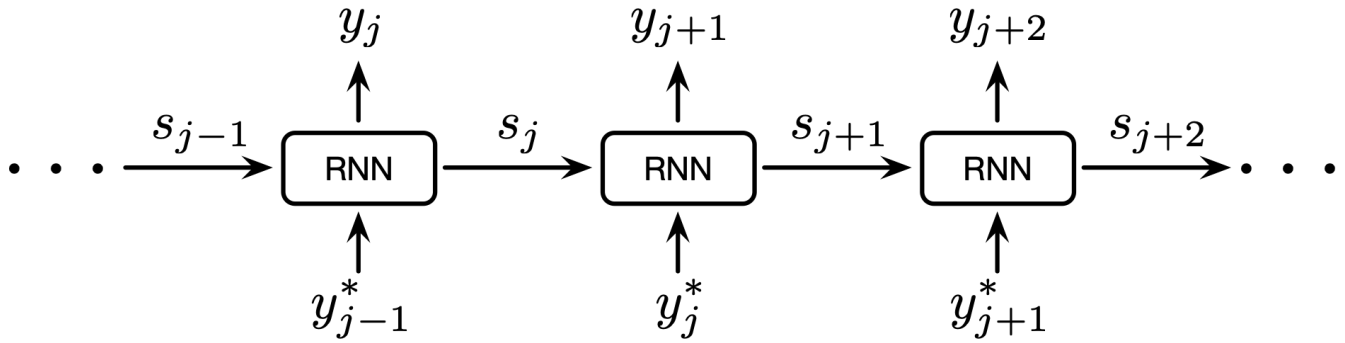
- Teacher Forcing



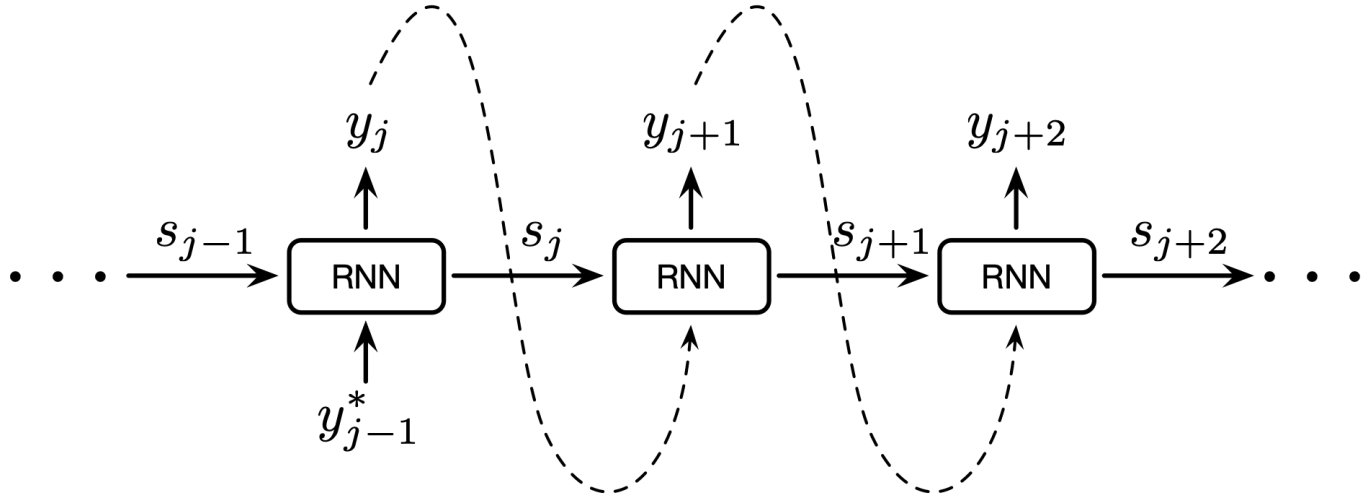
Loss:

$$\mathcal{L}_{\text{MLE}} = -\sum_{j=1}^{J} \log P_j(y_j^*)$$

Problems

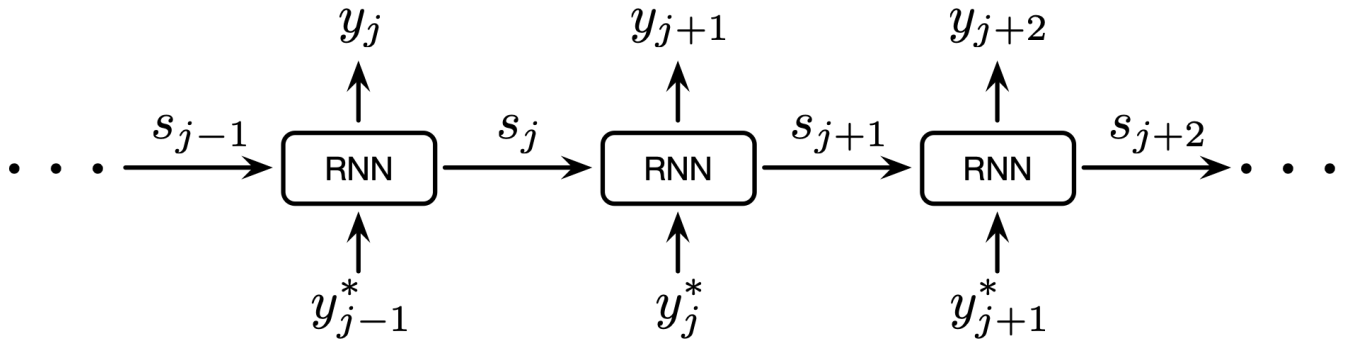


Training

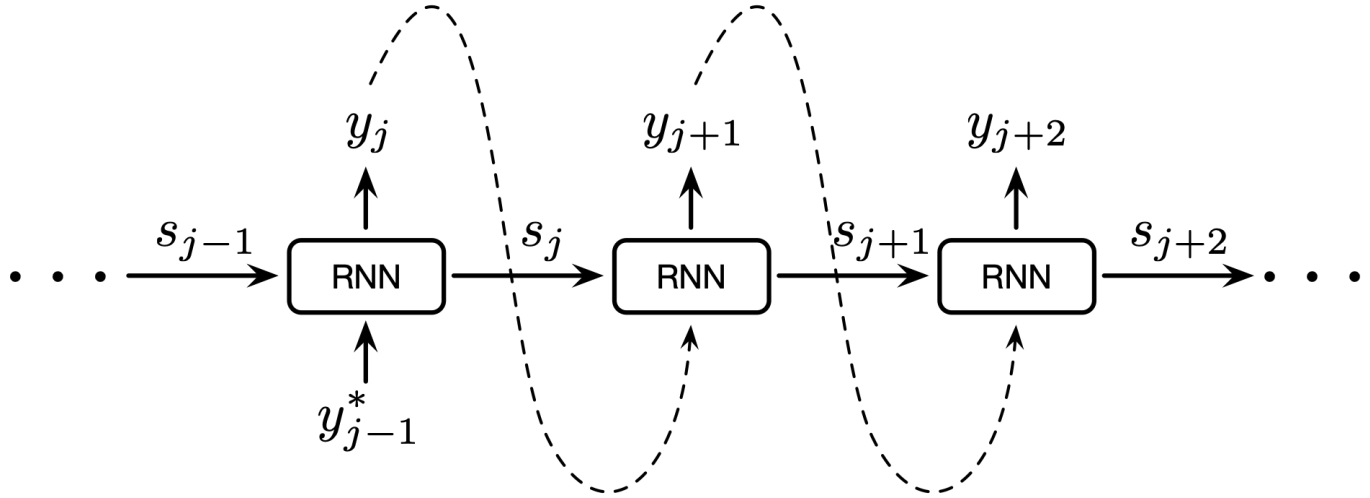


Inference

Problems



Training



Inference

Exposure Bias

Problems

Ground Truth:	How should I deal with this issue ?
Output:	How should I process this issue ?

Problems

Ground Truth:	How should I deal with this issue ?
Output:	How should I process this issue ?

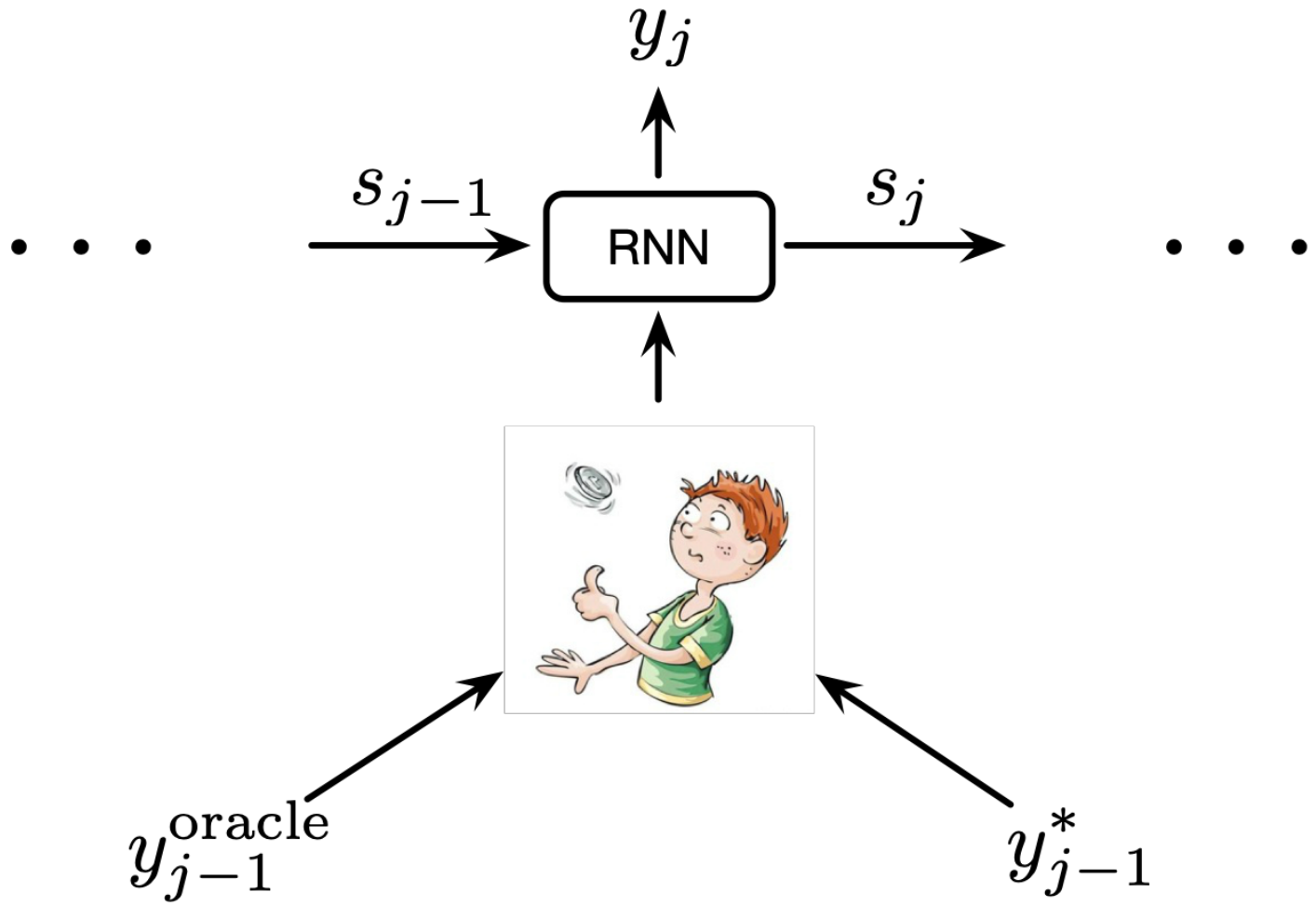
Word-level Matching

CONTENTS

CONTENTS

CONTENTS

The Model



Zhang et al., ACL 2019

Key parts

- How to generate oracle translation
- How to sample context
- How to train the model

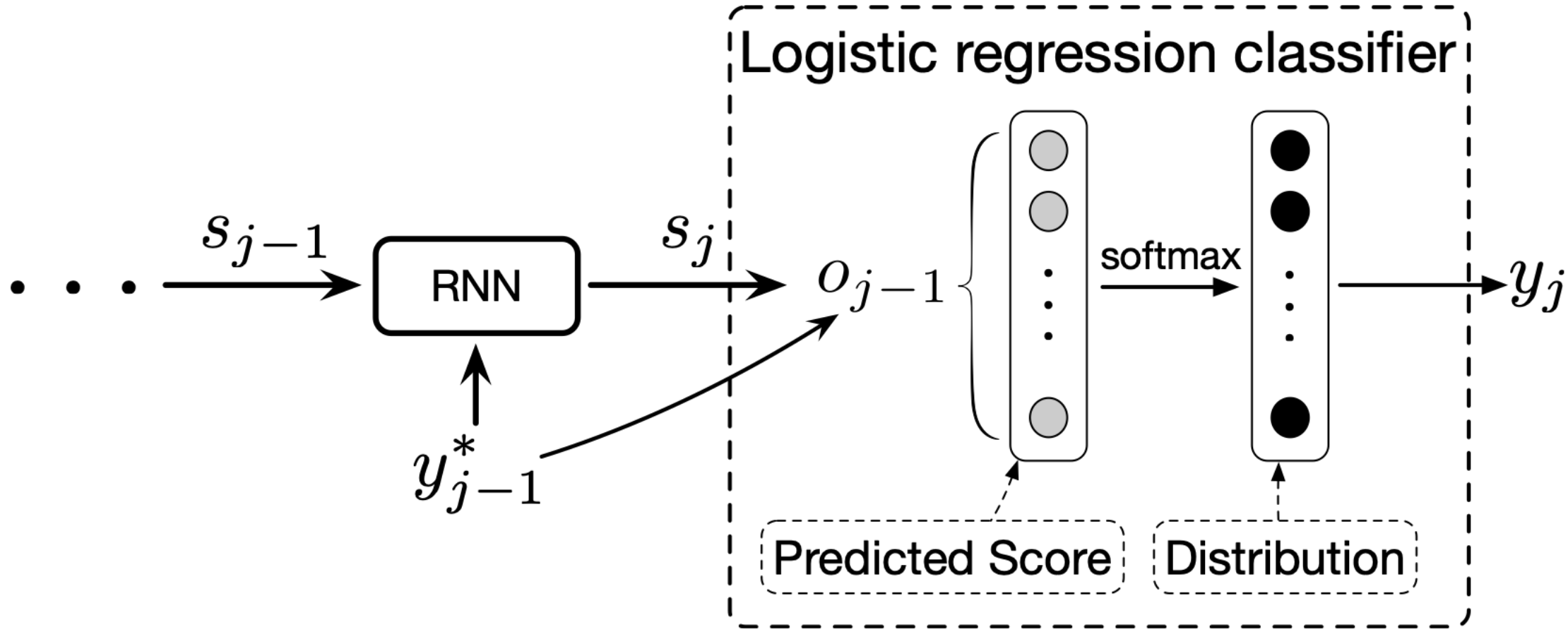
Zhang et al., ACL 2019

Oracle Translation Generation

- Word-level Oracle (WO)
- Sentence-level Oracle (SO)

Zhang et al., ACL 2019

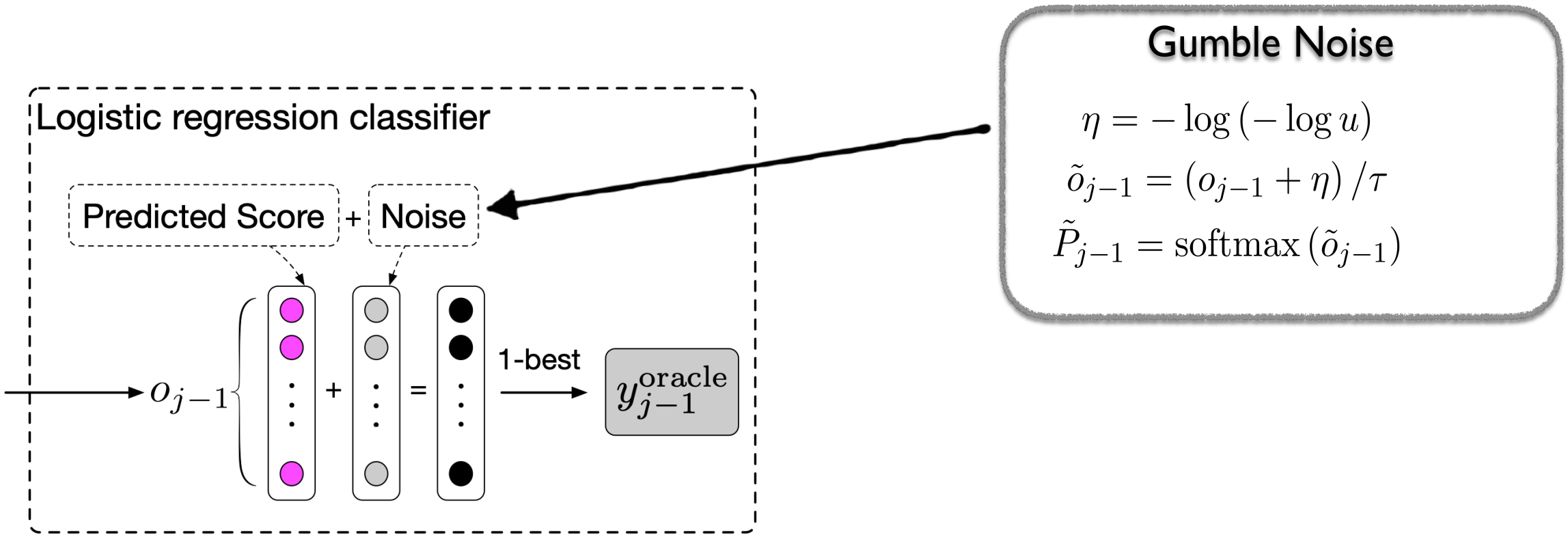
The RNNSearch Model



Zhang et al., ACL 2019

Oracle Translation Generation

- Word Oracle



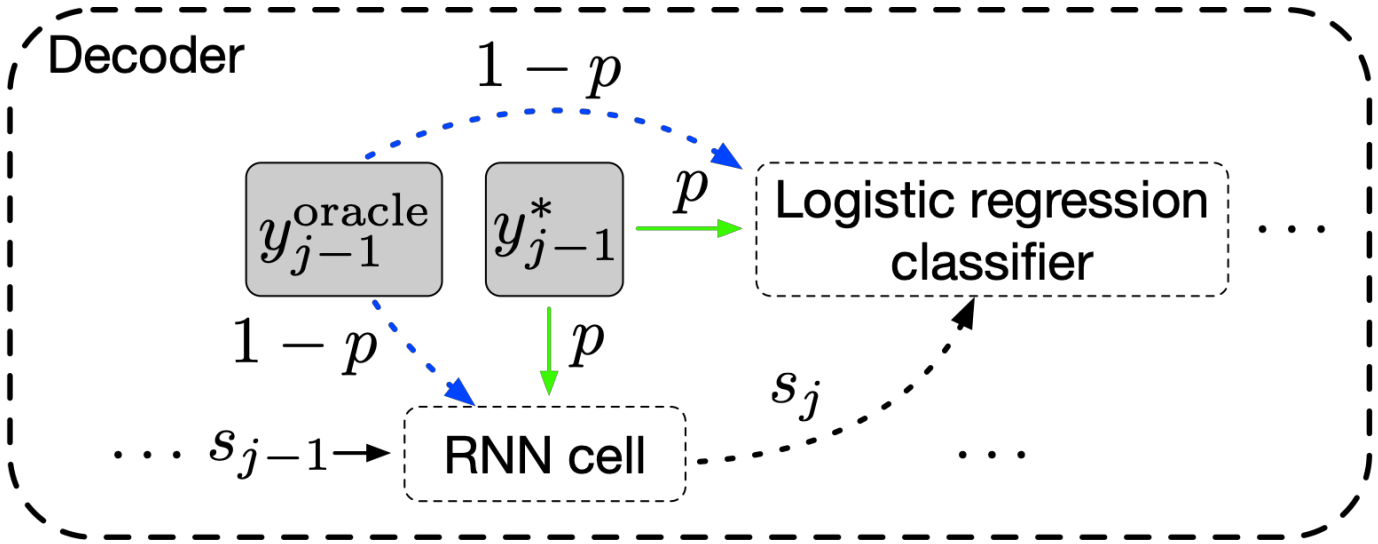
Zhang et al., ACL 2019

Oracle Translation Generation

- Sentence Oracle
  - Generate top-k translations by beam search
  - Rerank the top-k translation with BLEU
  - Select the top

Zhang et al., ACL 2019

Scheduled Sampling with Decay



where

$$p = \frac{\mu}{\mu + \exp\left(e/\mu\right)}$$

epoch number

Zhang et al., ACL 2019

Training

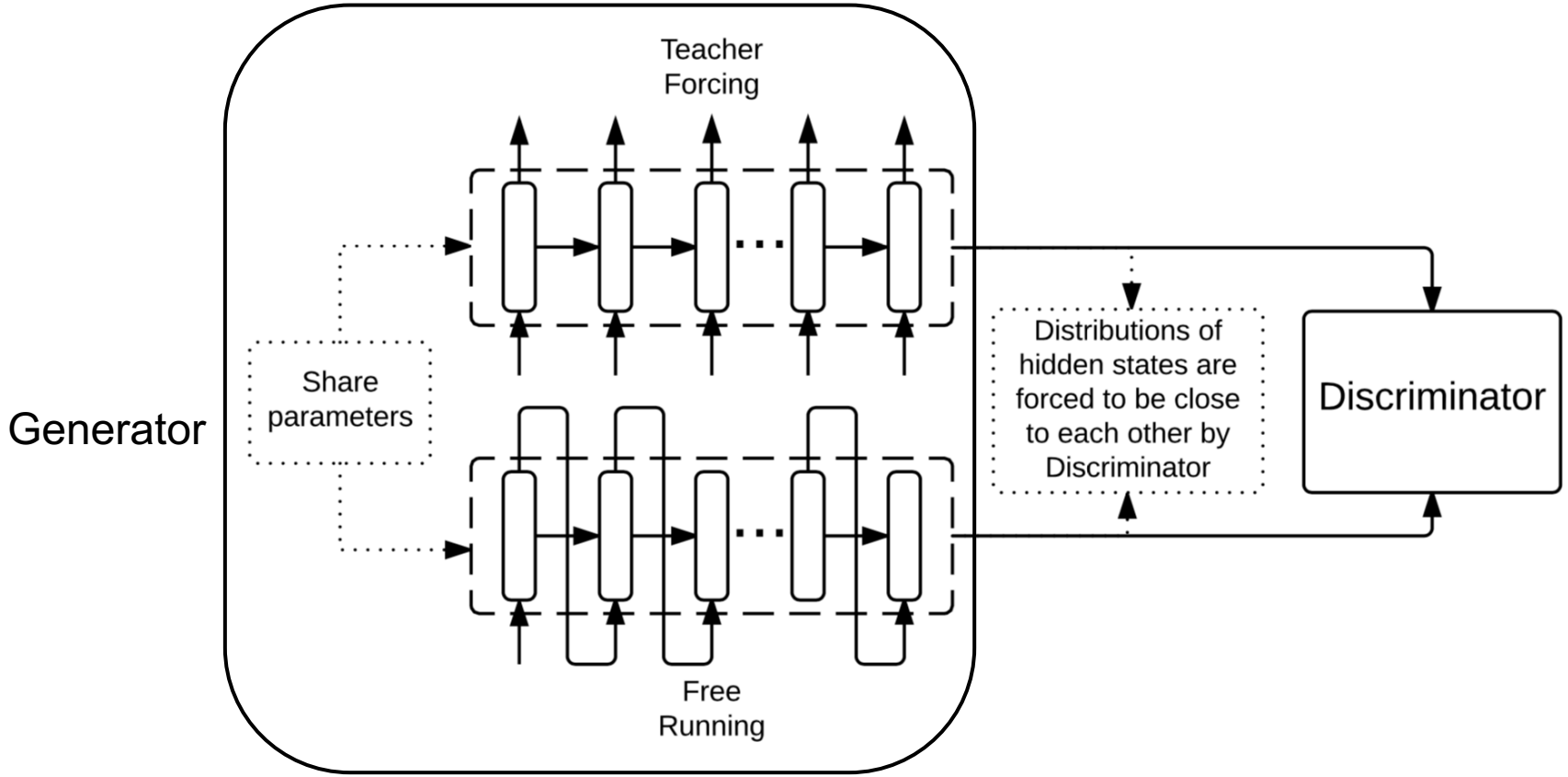
- Teacher Forcing + MLE

$$\mathcal{L}(\theta) = -\sum_{n=1}^{N} \sum_{j=1}^{|\mathbf{y}^n|} \log P_j^n \left[y_j^{*n} \right]$$

Zhang et al., ACL 2019

CONTENTS

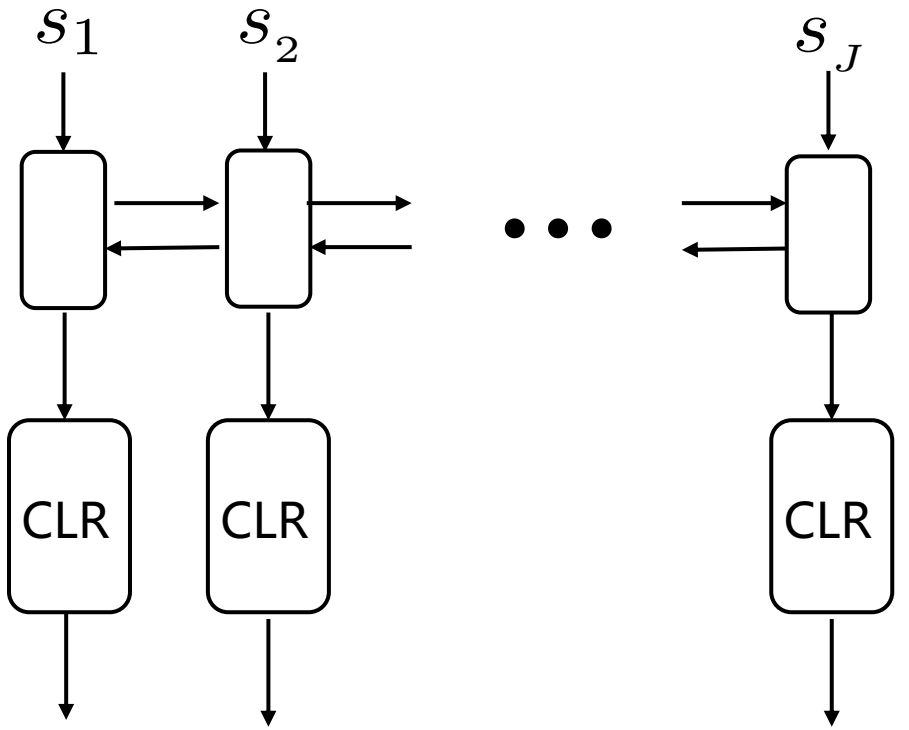
The Model



Goyal et al., ICLR 2016

Discriminator

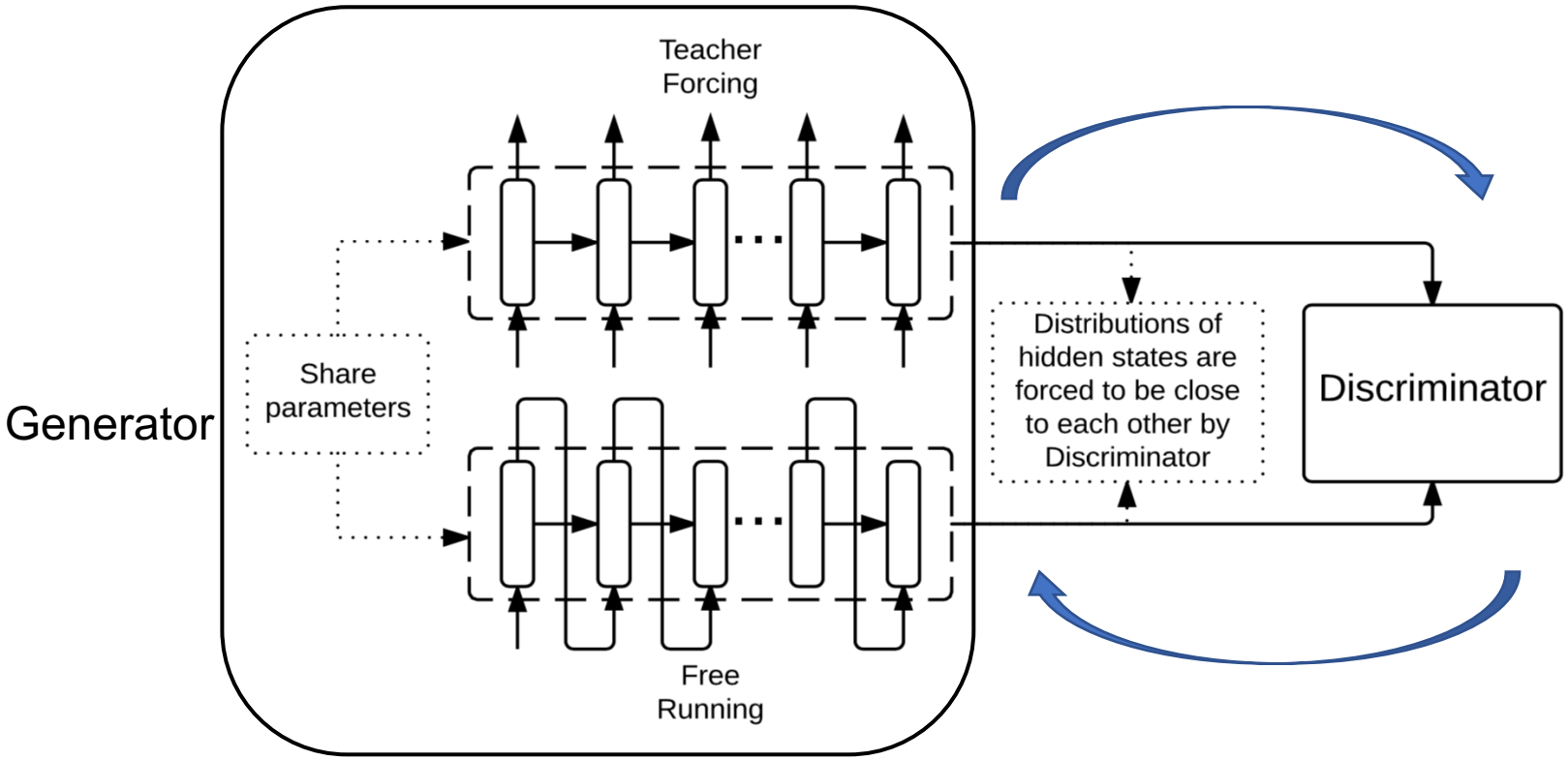
Free Running
or
Teacher Forcing



Goyal et al., ICLR 2016

Training

- Adversarial training in the GAN style



Goyal et al., ICLR 2016

Loss for Discriminator

$$C_d(\boldsymbol{\theta}_d|\boldsymbol{\theta}_g) = E_{(\boldsymbol{x},\boldsymbol{y})\sim\text{data}}[-\log D(B(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\theta}_g),\boldsymbol{\theta}_d)+E_{\boldsymbol{y}\sim P_{\boldsymbol{\theta}_g}(\boldsymbol{y}|\boldsymbol{x})}[-\log(1-D(B(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\theta}_g),\boldsymbol{\theta}_d)]]$$

Goyal et al., ICLR 2016

Loss for Generator

Conventional Loss:

$$NLL(\boldsymbol{\theta}_g) = E_{(\boldsymbol{x},\boldsymbol{y})\sim\text{data}}[-\log P_{\boldsymbol{\theta}_g}(\boldsymbol{y}|\boldsymbol{x})]$$

Losses to fool the discriminator:

$$C_t(\boldsymbol{\theta}_g|\boldsymbol{\theta}_d) = E_{(\boldsymbol{x},\boldsymbol{y})\sim\text{data}}[-\log(1 - D(B(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\theta}_g),\boldsymbol{\theta}_d))]$$

$$C_f(\boldsymbol{\theta}_g|\boldsymbol{\theta}_d) = E_{\boldsymbol{x}\sim\text{data},\boldsymbol{y}\sim P_{\boldsymbol{\theta}_g}(\boldsymbol{y}|\boldsymbol{x})}[-\log D(B(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\theta}_g),\boldsymbol{\theta}_d)]$$

Goyal et al., ICLR 2016

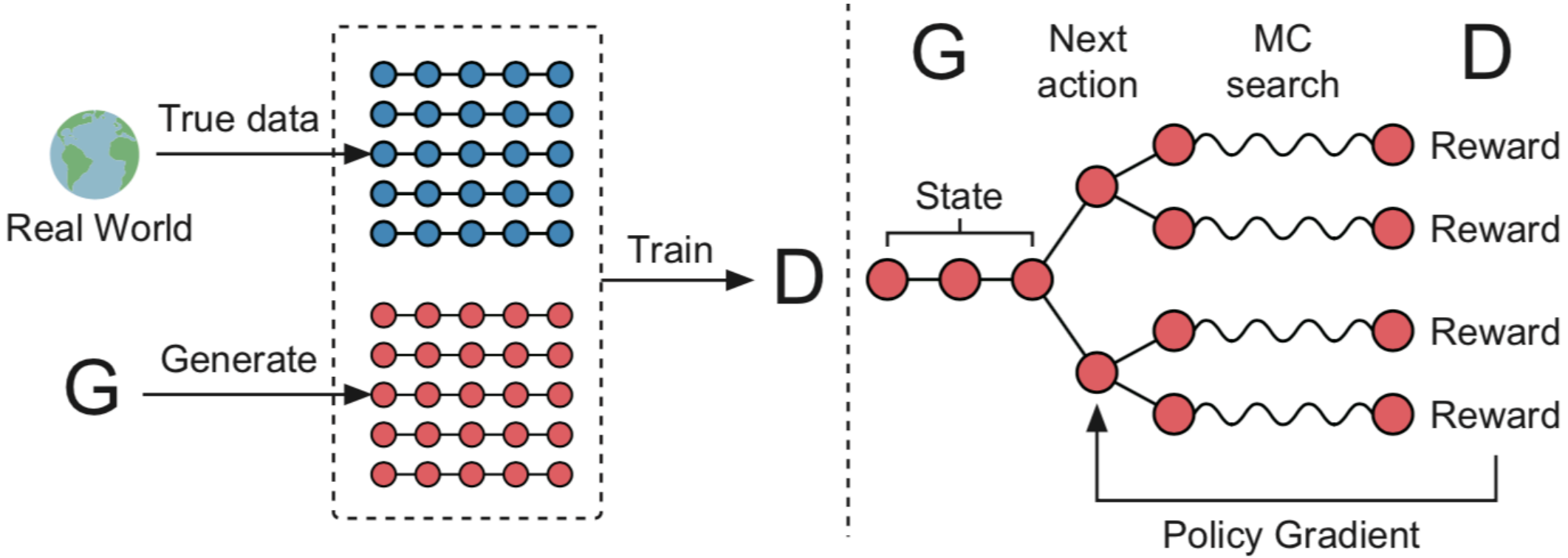
CONTENTS

Framework

- Generator
  - A sequence model
- Sequence-level Reward
  - BLEU, GLEU, etc
  - A score from a Discriminator
- Training strategy
  - Policy Gradient
  - Actor-Critic

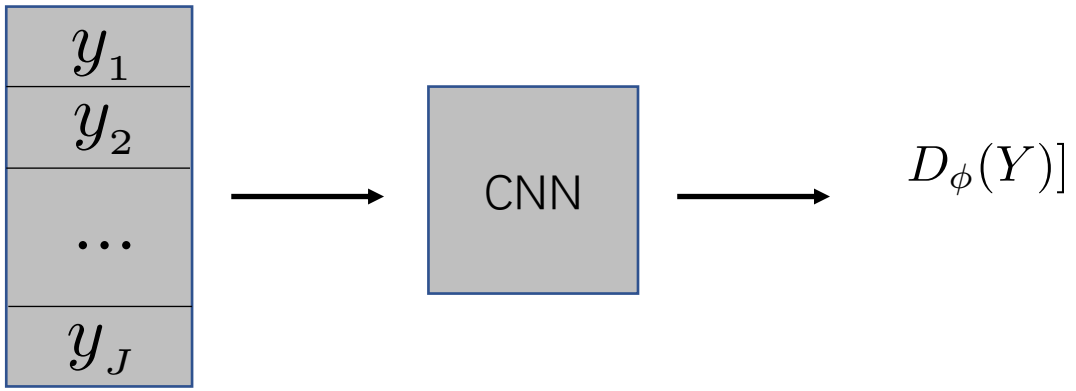
Yu et al., AAAI 2017

SeqGAN



Yu et al., AAAI 2017

Discriminator



Yu et al., AAAI 2017

Training

- Loss for discriminator:

$$\mathcal{L}_d(\phi) = -\mathbb{E}_{Y\sim p_{\text{data}}}[\log D_\phi(Y)] - \mathbb{E}_{Y\sim G_\theta}[\log(1-D_\phi(Y))]$$

- Loss for generator: **Policy Gradient**

$$\mathcal{L}_g(\theta) = -\sum_{\boldsymbol{y}\in G} p(\boldsymbol{y})\cdot r(\boldsymbol{y})$$

Yu et al., AAAI 2017

CONTENTS

Differentiable Sequence-level Loss

- Loss:
  - N-gram-based, e.g., BLEU, GLEU, etc

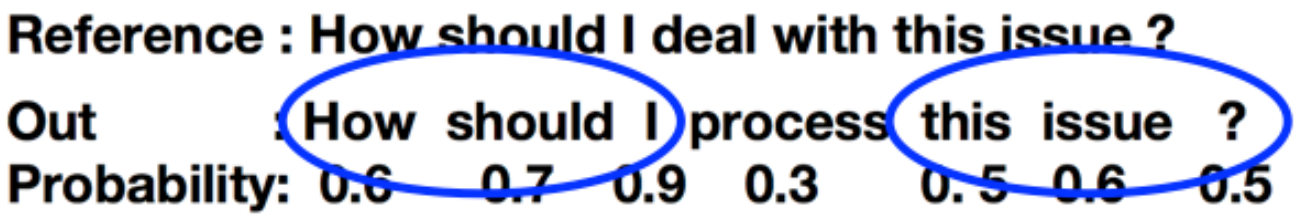
$$\text{BLEU} = \text{BP} \cdot \exp(\sum_{n=1}^{N} w_n \log p_n)$$

  - Probabilistic n-gram precision

$$\text{P-BLEU} = \text{BP} \cdot \exp(\sum_{n=1}^{N} w_n \log \tilde{p}_n)$$

Shao et al., EMNLP 2018

Probabilistic N-gram Accuracy: A 3-gram example

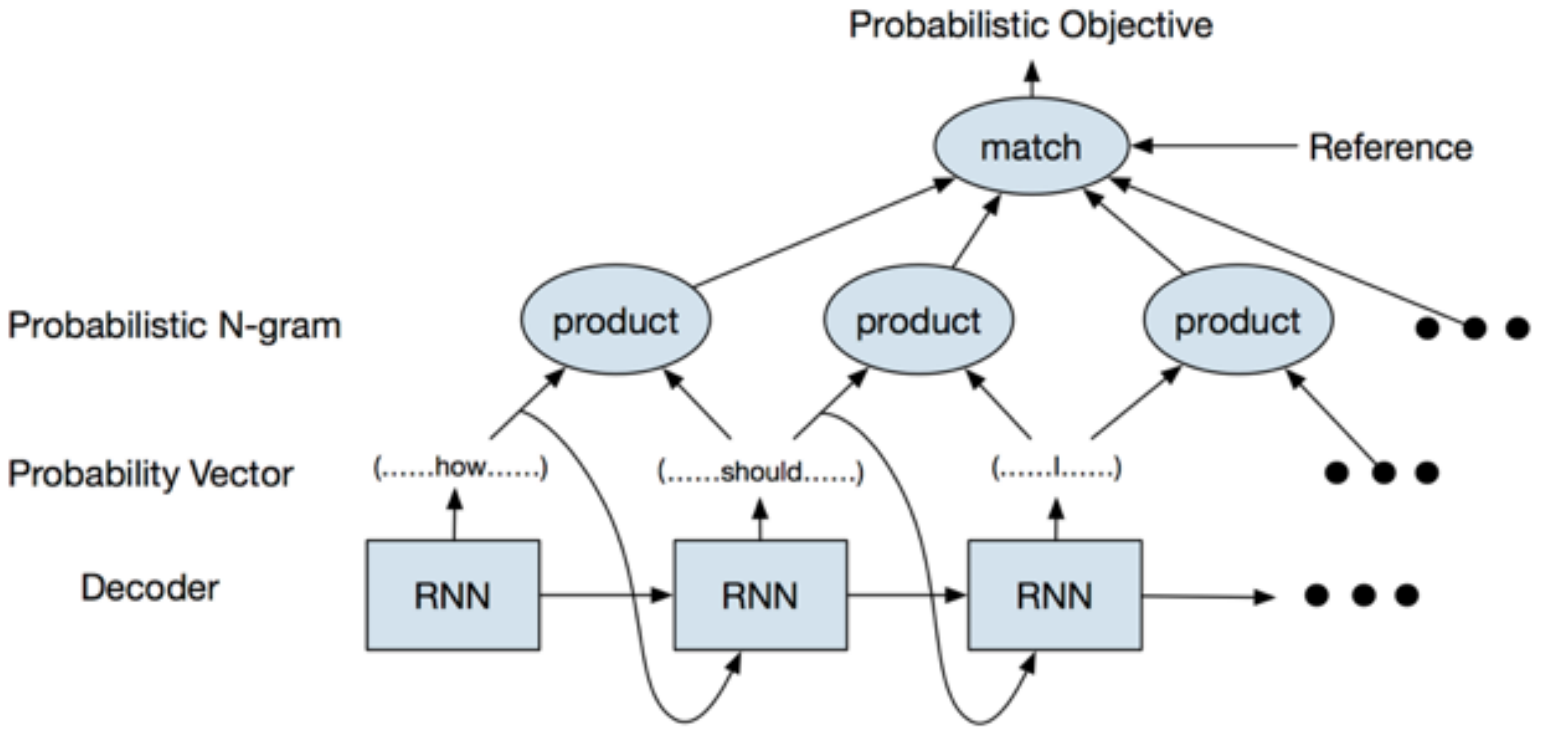


- The probabilistic count of 3-gram "How should I" : 0.6 * 0.7 * 0.9 = 0.378
- The probabilistic count of 3-gram "this issue ?" : 0.7 * 0.9 * 0.5 = 0.315
- The total probabilistic count of 3-grams : 1.107
- Precision of 3-gram: $\frac{0.378+0.315}{1.107}$

Shao et al., EMNLP 2018

Training

- A 2-gram example



Shao et al., EMNLP 2018

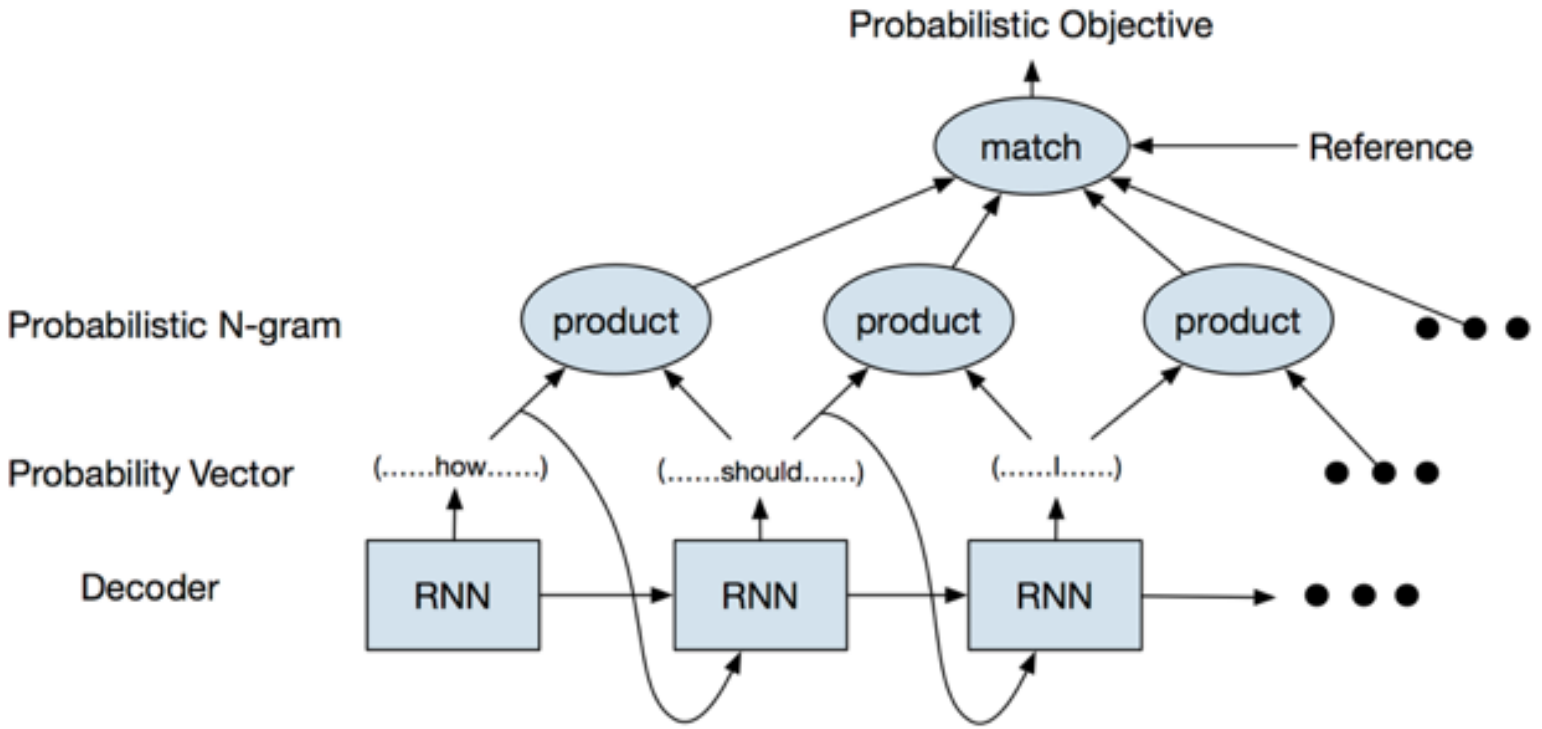
Training

- A 2-gram example

Greedy Search

P-BLEU

SGD



Shao et al., EMNLP 2018

References

- Wen Zhang, Yang Feng, Fandong Meng, Di You, Qun Liu. Bridging the Gap between Training and Inference for Neural Machine Translation. ACL 2019.
- Anirudh Goyal, Alex Lamb, Ying Zhang, Saizheng Zhang, Aaron Courville, Yoshua Bengio. Professor Forcing: A New Algorithm for Training Recurrent Networks. NIPS 2016.
- Lantao Yu, Weinan Zhang, Jun Wang, Yong Yu. SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient. AAAI 2017.
- Chenze Shao, Yang Feng, Xilin Chen. Greedy Search with Probabilistic N-gram Matching for Neural Machine Translation. EMNLP 2018.

Thanks for your attention!