

Probabilistic Graphical Models

Text Generation

Zhiteng Hu

Lecture 19, March 27, 2019

Reading: see class homepage



Text Generation Tasks

- Generates natural language from input data or machine representations





Text Generation Tasks

- Generates natural language from input data or machine representations
- Spans a broad set of natural language processing (NLP) tasks:

<u>Task</u>	<u>Input X</u>	<u>Output Y (Text)</u>
Chatbot / Dialog System	Utterance	Response
Machine Translation	English	Chinese
Summarization	Document	Short paragraph
Description Generation	Structured data	Description
Captioning	Image/video	Description
Speech Recognition	Speech	Transcript





Two Central Goals

- Generating human-like, grammatical, and readable text
 - I.e., generating **natural** language
- Generating text that contains desired information inferred from inputs
 - Machine translation
 - Source sentence --> target sentence w/ the same meaning
 - Data description
 - Table --> data report describing the table
 - Attribute control
 - Sentiment: positive --> ``I like this restaurant''
 - Conversation control
 - Control conversation strategy and topic





Two Central Goals

- Generating human-like, grammatical, and readable text
 - I.e., generating **natural** language
- Generating text that contains desired information inferred from inputs
 - Machine translation
 - Source sentence --> target sentence w/ the same meaning
 - Data description
 - Table --> data report describing the table
 - Attribute control
 - Sentiment: positive --> ``I like this restaurant''
 - Conversation control
 - Control conversation strategy and topic

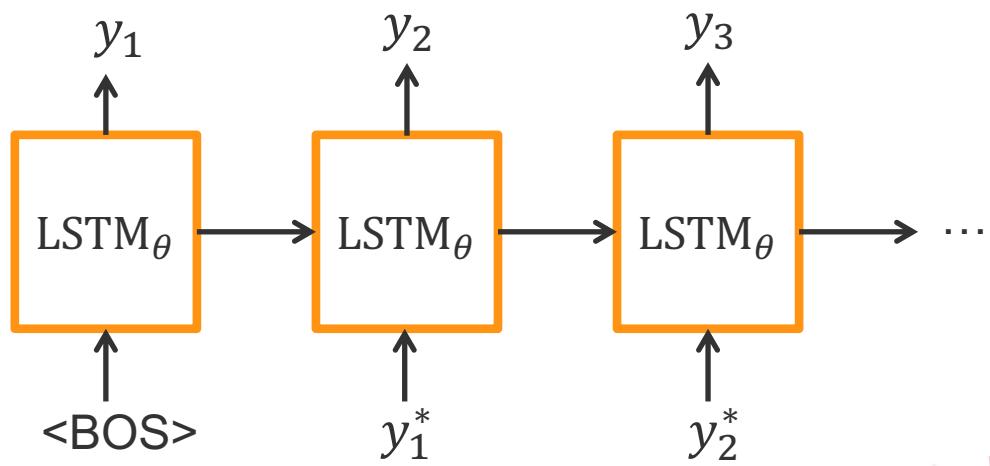




Common Model for Text Generation: Language Model

- Calculates the probability of a sentence:
 - Sentence: $\mathbf{y} = (y_1, y_2 \dots, y_T)$

$$p_{\theta}(\mathbf{y}) = \prod_t p_{\theta}(y_t \mid \mathbf{y}_{1:t-1})$$

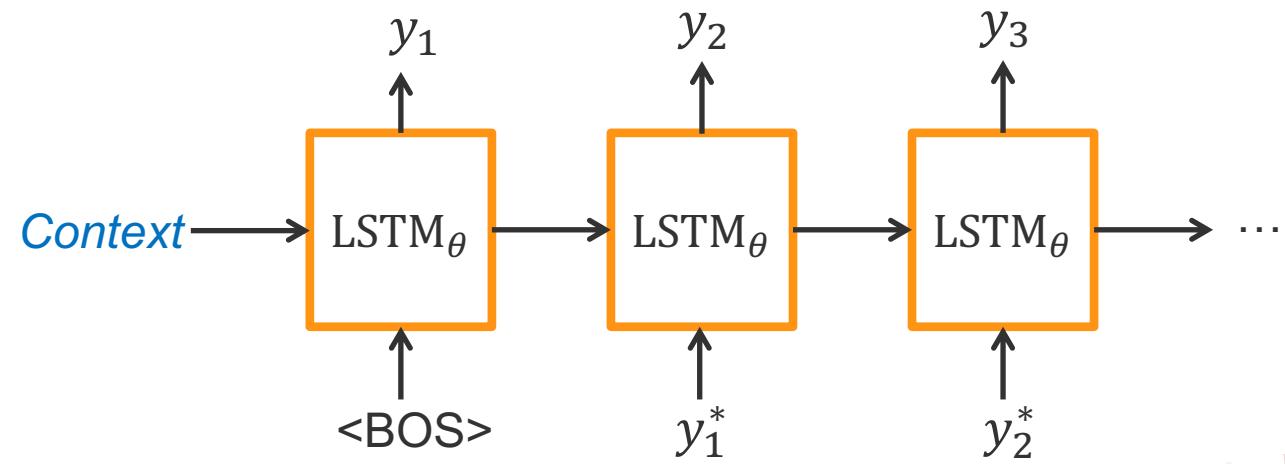




Common Model for Text Generation: Conditional Language Model

- Calculates the probability of a sentence:
 - Sentence: $\mathbf{y} = (y_1, y_2 \dots, y_T)$, Context: \mathbf{x}

$$p_{\theta}(\mathbf{y} | \mathbf{x}) = \prod_t p_{\theta}(y_t | \mathbf{y}_{1:t-1}, \mathbf{x})$$

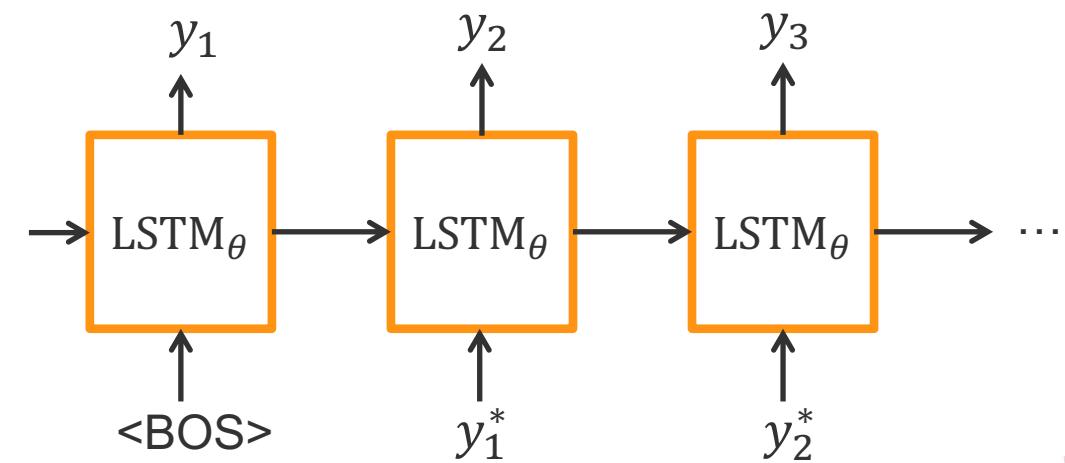




Common Learning Algorithm: Maximum Likelihood Estimation (MLE)

- Training
 - Maximize data log-likelihood
 - Given ground-truth data $\mathbf{y}^* = (y_1^*, y_2^*, \dots, y_{T^*}^*)$

$$\mathcal{L}_{\text{MLE}}(\boldsymbol{\theta}) = \log p_{\boldsymbol{\theta}}(\mathbf{y}^* \mid \mathbf{x}) = \log \prod_t p_{\boldsymbol{\theta}}(y_t^* \mid \mathbf{y}_{1:t-1}^*, \mathbf{x})$$



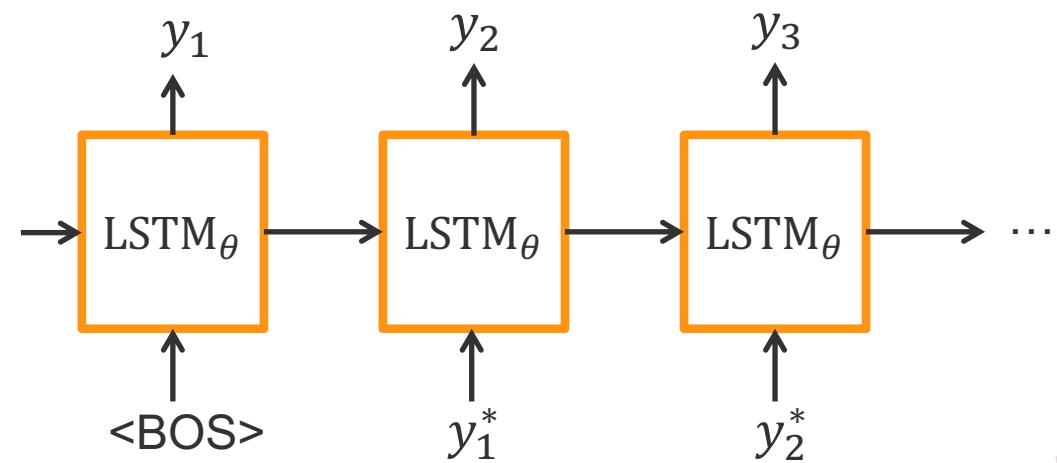


Common Learning Algorithm: Maximum Likelihood Estimation (MLE)

- Training
 - Maximize data log-likelihood
 - Given ground-truth data $\mathbf{y}^* = (y_1^*, y_2^*, \dots, y_{T^*}^*)$

$$\mathcal{L}_{\text{MLE}}(\boldsymbol{\theta}) = \log p_{\boldsymbol{\theta}}(\mathbf{y}^* \mid \mathbf{x}) = \log \prod_t p_{\boldsymbol{\theta}}(y_t^* \mid \mathbf{y}_{1:t-1}^*, \mathbf{x})$$

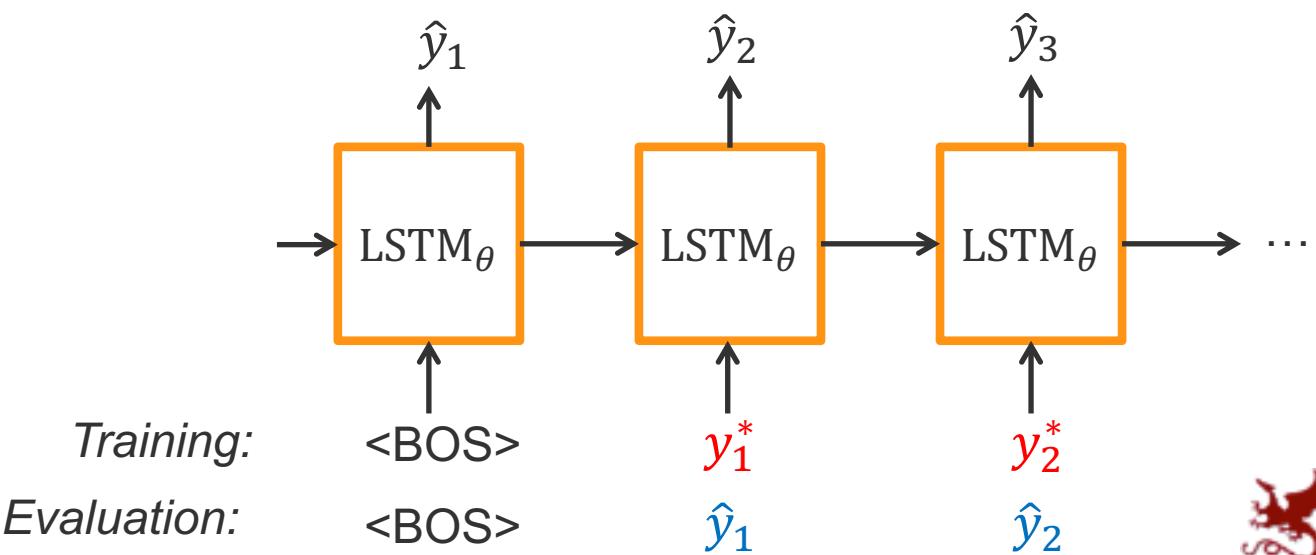
- Evaluation
 - Task-specific metrics
 - BLEU for machine translation
 - ROUGE for summarization
 -





Two Issues of MLE

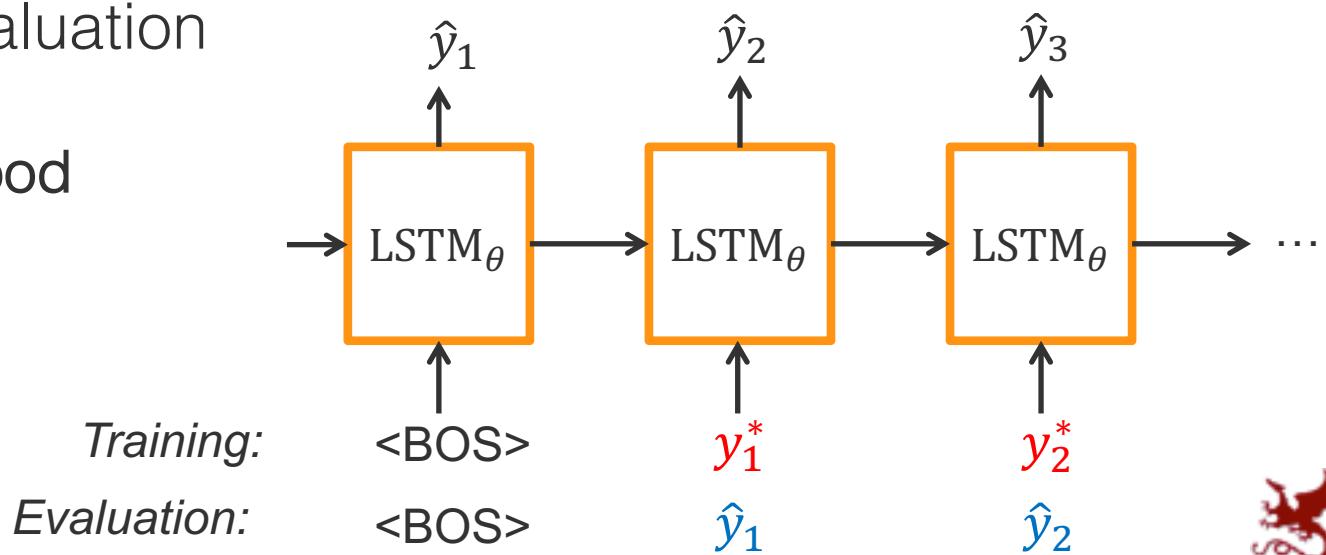
- Exposure bias [Ranzato et al., 2015]
 - Training: predict next token given the previous **ground-truth sequence**
 - Evaluation: predict next token given the previous sequence that are generated by the model itself





Two Issues of MLE

- Exposure bias [Ranzato et al., 2015]
 - Training: predict next token given the previous **ground-truth sequence**
 - Evaluation: predict next token given the previous sequence that are generated by the model itself
- Mismatch between training & evaluation criteria
 - Train to maximize data log-likelihood
 - Evaluate with, e.g., BLEU





Possible Solutions

- Reinforcement learning [e.g., Ranzato et al., 2015]
 - Maximize expected reward under the model distribution

$$\max_{\theta} \mathbb{E}_{p_{\theta}(y)}[R(y, y^*)]$$





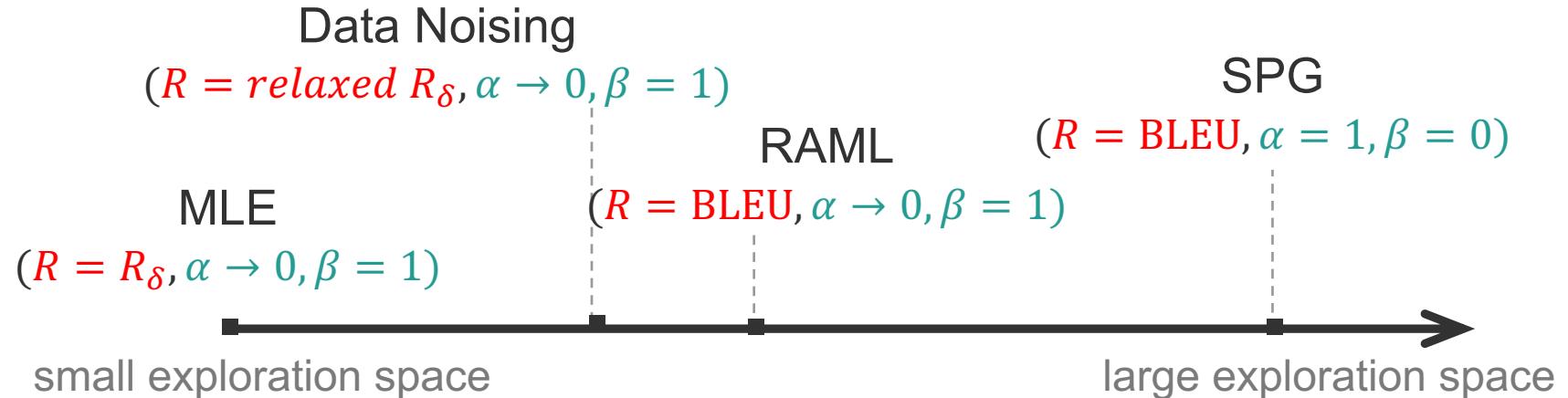
Possible Solutions

- Reinforcement learning [e.g., Ranzato et al., 2015]
 - Maximize expected reward under the model distribution
$$\max_{\theta} \mathbb{E}_{p_{\theta}(y)}[R(y, y^*)]$$
 - Problems
 - Extremely large sequence space ($\sim 50000^{50}$)
 - High variance and poor exploration efficiency during training
- Recent work for more practical training
 - Reward Augmented Maximum Likelihood (RAML) [Norouzi et al., 16]
 - Add reward-aware perturbation to the MLE data examples
 - Softmax Policy Gradient (SPG) [Ding & Soricut, 17]
 - Use reward distribution for effective sampling and estimating policy gradient
 - Data noising [Xie et al., 17]
 - Add random noise to data





Connecting the Dots



- Establish a unified perspective of the diverse learning algorithms
- All these algorithms are special instances of a generalized **entropy regularized policy optimization** (ERPO) framework
- The only difference is the choice of **reward** and the values of some **hyperparameters**
- The unified view inspires new, improved algorithms





Generalized Entropy Regularized Policy Optimization (ERPO)

- Consider a sequence generation model $p_{\theta}(\mathbf{y} | \mathbf{x})$
- Given a reward function $R(\mathbf{y}|\mathbf{y}^*) \in \mathbb{R}$, e.g., BLEU(\mathbf{y}, \mathbf{y}^*)
- Assume a variational distribution $q(\mathbf{y}|\mathbf{x})$
- The generalized ERPO objective:

$$\mathcal{L}(q, \boldsymbol{\theta}) = \mathbb{E}_q[R(\mathbf{y}|\mathbf{y}^*)] - \alpha \text{KL}(q(\mathbf{y}|\mathbf{x}) || p_{\theta}(\mathbf{y}|\mathbf{x})) + \beta H(q)$$

- Impose supervision R on q
- The KL divergence enforces model p_{θ} to stay close to q
- Additional entropy regularizer on q
- The objective is a generalization of, or closely related to, many popular RL algorithms
 - Relative entropy policy search [Peters et al., 10], Trust Region Policy Optimization [Schulman et al., 15], maximum entropy policy gradient [Ziebart., 10], and others [Haarnoja et al., 17, The et al., 17, etc]





Generalized Entropy Regularized Policy Optimization (ERPO)

- The generalized ERPO objective:

$$\mathcal{L}(q, \theta) = \mathbb{E}_q [R(y|y^*)] - \alpha \text{KL}(q(y|x) || p_\theta(y|x)) + \beta H(q)$$

- Solve with an **EM-style procedure**. At iteration n

- E-step
$$q^{n+1}(y|x) \propto \exp \left\{ \frac{\alpha \log p_{\theta^n}(y|x) + R(y|y^*)}{\alpha + \beta} \right\}$$
- M-step
$$\theta^{n+1} = \operatorname{argmax}_{\theta} \mathbb{E}_{q^{n+1}} [\log p_{\theta}(y|x)]$$

- Some intuitive interpretations:

- $\alpha \rightarrow \infty$, then $q^{n+1} = p_\theta^n$ (i.e., minimal KL divergence)
- $\beta \rightarrow \infty$, then q^{n+1} is a uniform distribution (i.e., maximal entropy)
- M-step is to maximize the log-likelihood of samples from q^{n+1}





MLE as a Special Case of ERPO

- E-step $q^{n+1}(y|x) \propto \exp \left\{ \frac{\alpha \log p_{\theta^n}(y|x) + R(y|y^*)}{\alpha + \beta} \right\}$
 - M-step $\theta^{n+1} = \operatorname{argmax}_{\theta} \mathbb{E}_{q^{n+1}} [\log p_{\theta}(y|x)]$
-





MLE as a Special Case of ERPO

- E-step $q^{n+1}(\mathbf{y}|\mathbf{x}) \propto \exp \left\{ \frac{\alpha \log p_{\theta^n}(\mathbf{y}|\mathbf{x}) + R(\mathbf{y}|\mathbf{y}^*)}{\alpha + \beta} \right\}$
- M-step $\boldsymbol{\theta}^{n+1} = \operatorname{argmax}_{\boldsymbol{\theta}} \mathbb{E}_{q^{n+1}} [\log p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})]$
- Let $R = R_{\delta}(\mathbf{y}|\mathbf{y}^*) := \begin{cases} 1 & \text{if } \mathbf{y} = \mathbf{y}^* \\ -\infty & \text{otherwise} \end{cases}$
 $\alpha \rightarrow 0, \beta = 1$





MLE as a Special Case of ERPO

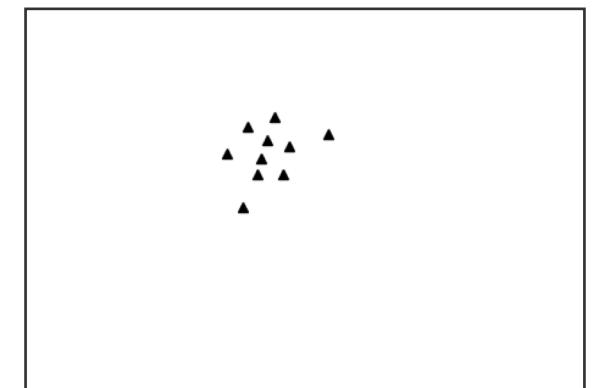
- E-step $q^{n+1}(\mathbf{y}|\mathbf{x}) \propto \exp \left\{ \frac{\alpha \log p_{\theta^n}(\mathbf{y}|\mathbf{x}) + R(\mathbf{y}|\mathbf{y}^*)}{\alpha + \beta} \right\}$
- M-step $\theta^{n+1} = \operatorname{argmax}_{\theta} \mathbb{E}_{q^{n+1}} [\log p_{\theta}(\mathbf{y}|\mathbf{x})]$
- Let $R = R_{\delta}(\mathbf{y}|\mathbf{y}^*) := \begin{cases} 1 & \text{if } \mathbf{y} = \mathbf{y}^* \\ -\infty & \text{otherwise} \end{cases}$
 $\alpha \rightarrow 0, \beta = 1$
- Then we have
 - E-step $q(\mathbf{y}|\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{y} = \mathbf{y}^* \\ 0 & \text{otherwise} \end{cases}$ \longrightarrow empirical data distribution
 - M-step $\theta^{n+1} = \operatorname{argmax}_{\theta} \log p_{\theta}(\mathbf{y}^*|\mathbf{x})$ \longrightarrow maximum likelihood estimation





MLE as a Special Case of ERPO

- E-step $q^{n+1}(\mathbf{y}|\mathbf{x}) \propto \exp \left\{ \frac{\alpha \log p_{\theta^n}(\mathbf{y}|\mathbf{x}) + R(\mathbf{y}|\mathbf{y}^*)}{\alpha + \beta} \right\}$
- M-step $\boldsymbol{\theta}^{n+1} = \operatorname{argmax}_{\boldsymbol{\theta}} \mathbb{E}_{q^{n+1}} [\log p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})]$
- Let $R = R_{\delta}(\mathbf{y}|\mathbf{y}^*) := \begin{cases} 1 & \text{if } \mathbf{y} = \mathbf{y}^* \\ -\infty & \text{otherwise} \end{cases}$
 $\alpha \rightarrow 0, \beta = 1$
- MLE is a policy optimization with a δ -function reward
 - Make void any exploration beyond the training data, and thus the exposure bias
 - The regular shape of exploration space makes the implementation of the algorithm very simple and efficient



exploration space of MLE



RAML as a Special Case of ERPO

- E-step $q^{n+1}(y|x) \propto \exp \left\{ \frac{\alpha \log p_{\theta^n}(y|x) + R(y|y^*)}{\alpha + \beta} \right\}$
 - M-step $\theta^{n+1} = \operatorname{argmax}_{\theta} \mathbb{E}_{q^{n+1}} [\log p_{\theta}(y|x)]$
-

- Let $R : \text{a common reward such as } \text{BLEU}(y, y^*)$

$$\alpha \rightarrow 0, \beta = 1$$





RAML as a Special Case of ERPO

- E-step $q^{n+1}(y|x) \propto \exp \left\{ \frac{\alpha \log p_{\theta^n}(y|x) + R(y|y^*)}{\alpha + \beta} \right\}$

- M-step $\theta^{n+1} = \operatorname{argmax}_{\theta} \mathbb{E}_{q^{n+1}} [\log p_{\theta}(y|x)]$

- Let R : a common reward such as $\text{BLEU}(y, y^*)$

$$\alpha \rightarrow 0, \beta = 1$$

- Then we have

- E-step $q(y|x) \propto \exp\{R(y, y^*)\}$ \longrightarrow *exponentiated reward distribution*

- M-step $\max_{\theta} \mathbb{E}_q [\log p_{\theta}(y|x)]$ \longrightarrow *Reward augmented maximum likelihood (RAML)*





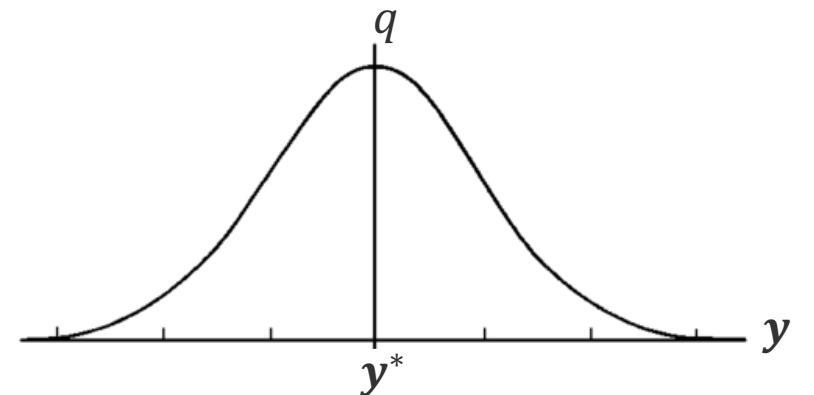
RAML as a Special Case of ERPO

- E-step $q^{n+1}(y|x) \propto \exp \left\{ \frac{\alpha \log p_{\theta^n}(y|x) + R(y|y^*)}{\alpha + \beta} \right\}$
- M-step $\theta^{n+1} = \operatorname{argmax}_{\theta} \mathbb{E}_{q^{n+1}} [\log p_{\theta}(y|x)]$

-
- Let R : a common reward such as BLEU(y, y^*)

$$\alpha \rightarrow 0, \beta = 1$$

- Then we have



- E-step $q(y|x) \propto \exp\{R(y, y^*)\} \longrightarrow \text{exponentiated reward distribution}$

- M-step $\max_{\theta} \mathbb{E}_q [\log p_{\theta}(y|x)] \longrightarrow \text{Reward augmented maximum likelihood (RAML)}$





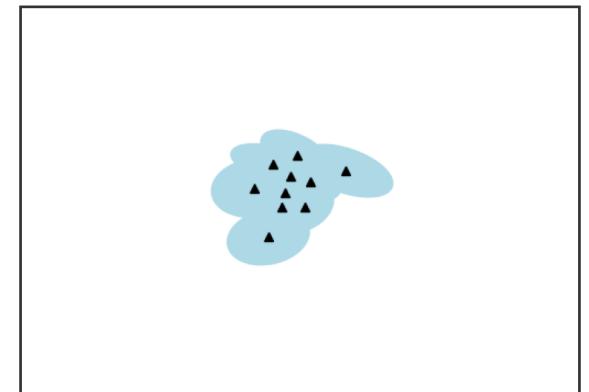
RAML as a Special Case of ERPO

- E-step $q^{n+1}(y|x) \propto \exp \left\{ \frac{\alpha \log p_{\theta^n}(y|x) + R(y|y^*)}{\alpha + \beta} \right\}$
 - M-step $\theta^{n+1} = \operatorname{argmax}_{\theta} \mathbb{E}_{q^{n+1}} [\log p_{\theta}(y|x)]$
-

- Let R : a common reward such as BLEU(y, y^*)

$$\alpha \rightarrow 0, \beta = 1$$

- Compared to MLE, RAML uses a task dependent reward
 - More smooth than R_δ
 - Permit a larger exploration space surrounding the training data
 - $\alpha \rightarrow 0$ ignores the model distribution for exploration



exploration space of RAML



SPG as a Special Case of ERPO

- E-step $q^{n+1}(y|x) \propto \exp \left\{ \frac{\alpha \log p_{\theta^n}(y|x) + R(y|y^*)}{\alpha + \beta} \right\}$
 - M-step $\theta^{n+1} = \operatorname{argmax}_{\theta} \mathbb{E}_{q^{n+1}} [\log p_{\theta}(y|x)]$
-

- Let R : a common reward such as $\text{BLEU}(y, y^*)$

$$\alpha = 1, \beta = 0$$





SPG as a Special Case of ERPO

- E-step $q^{n+1}(y|x) \propto \exp \left\{ \frac{\alpha \log p_{\theta^n}(y|x) + R(y|y^*)}{\alpha + \beta} \right\}$

- M-step $\theta^{n+1} = \operatorname{argmax}_{\theta} \mathbb{E}_{q^{n+1}} [\log p_{\theta}(y|x)]$

- Let R : a common reward such as $\text{BLEU}(y, y^*)$

$$\alpha = 1, \beta = 0$$

- Then we have

- E-step $q(y|x) \propto p_{\theta}(y|x) \exp\{R(y, y^*)\}$

----> *Softmax Policy Gradient (SPG)*

- M-step $\max_{\theta} \mathbb{E}_q [\log p_{\theta}(y|x)]$





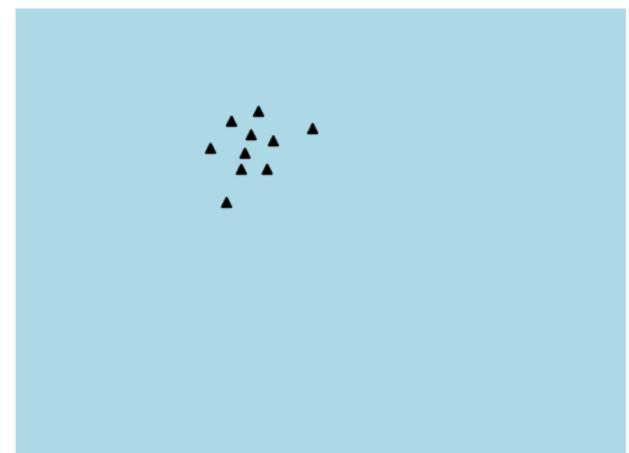
SPG as a Special Case of ERPO

- E-step $q^{n+1}(y|x) \propto \exp \left\{ \frac{\alpha \log p_{\theta^n}(y|x) + R(y|y^*)}{\alpha + \beta} \right\}$
 - M-step $\theta^{n+1} = \operatorname{argmax}_{\theta} \mathbb{E}_{q^{n+1}} [\log p_{\theta}(y|x)]$
-

- Let R : a common reward such as BLEU(y, y^*)

$$\alpha = 1, \beta = 0$$

- SPG uses both the model distribution and the reward for exploration
 - Largest exploration space
 - Increased learning difficulty, need more tricks during training



exploration space of SPG





Data Noising as a Special Case of ERPO

- E-step $q^{n+1}(\mathbf{y}|\mathbf{x}) \propto \exp \left\{ \frac{\alpha \log p_{\theta^n}(\mathbf{y}|\mathbf{x}) + R(\mathbf{y}|\mathbf{y}^*)}{\alpha + \beta} \right\}$
- M-step $\theta^{n+1} = \operatorname{argmax}_{\theta} \mathbb{E}_{q^{n+1}} [\log p_{\theta}(\mathbf{y}|\mathbf{x})]$

-
- Let R : a locally relaxed variant of $R_{\delta}(\mathbf{y}|\mathbf{y}^*)$
e.g., $R'_{\delta}(\mathbf{y}|\mathbf{y}^*) := \begin{cases} 1 & \text{if } \operatorname{diff}(\mathbf{y}, \mathbf{y}^*) = 1 \\ -\infty & \text{otherwise} \end{cases}$
- Randomly replace a single token with another uniformly picked token



$$\alpha \rightarrow 0, \beta = 1$$





Data Noising as a Special Case of ERPO

- E-step $q^{n+1}(\mathbf{y}|\mathbf{x}) \propto \exp \left\{ \frac{\alpha \log p_{\theta^n}(\mathbf{y}|\mathbf{x}) + R(\mathbf{y}|\mathbf{y}^*)}{\alpha + \beta} \right\}$
- M-step $\theta^{n+1} = \operatorname{argmax}_{\theta} \mathbb{E}_{q^{n+1}} [\log p_{\theta}(\mathbf{y}|\mathbf{x})]$

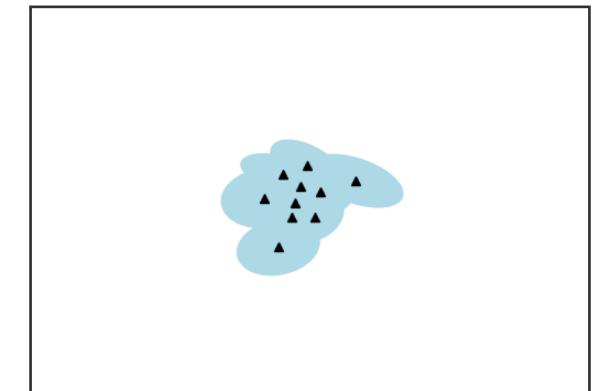
-
- Let R : a locally relaxed variant of $R_{\delta}(\mathbf{y}|\mathbf{y}^*)$

e.g., $R'_{\delta}(\mathbf{y}|\mathbf{y}^*) := \begin{cases} 1 & \text{if } \operatorname{diff}(\mathbf{y}, \mathbf{y}^*) = 1 \\ -\infty & \text{otherwise} \end{cases}$

$$\alpha \rightarrow 0, \beta = 1$$

- Data noising is similar to RAML
 - Data noising adds *random* noise, which is easy to implement
 - RAML adds *reward-aware* noise, which can be hard to implement

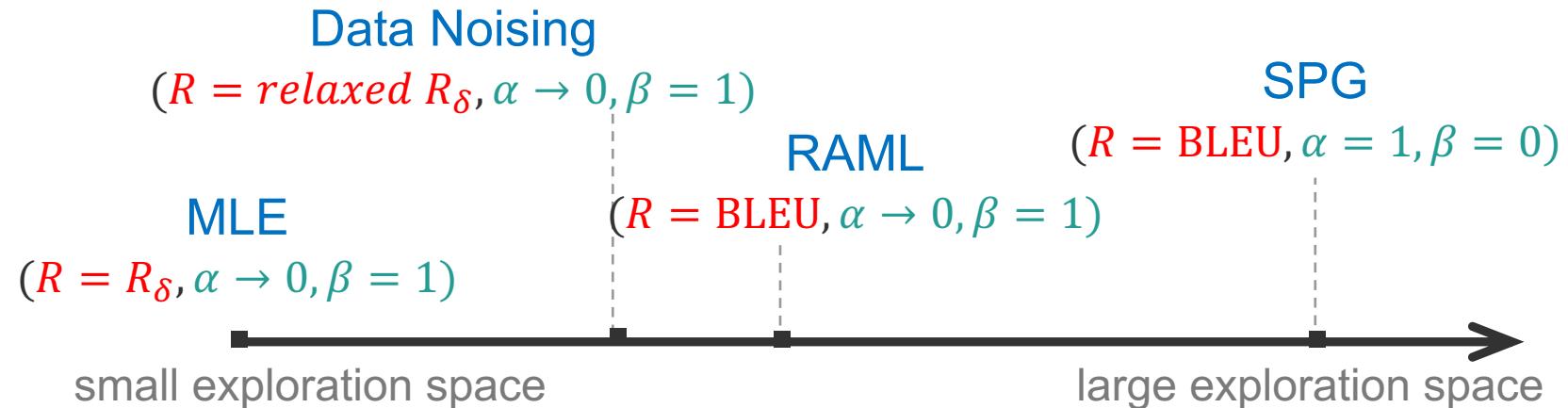
Randomly replace a single token with another uniformly picked token



exploration space of data noising



Interpolation algorithm

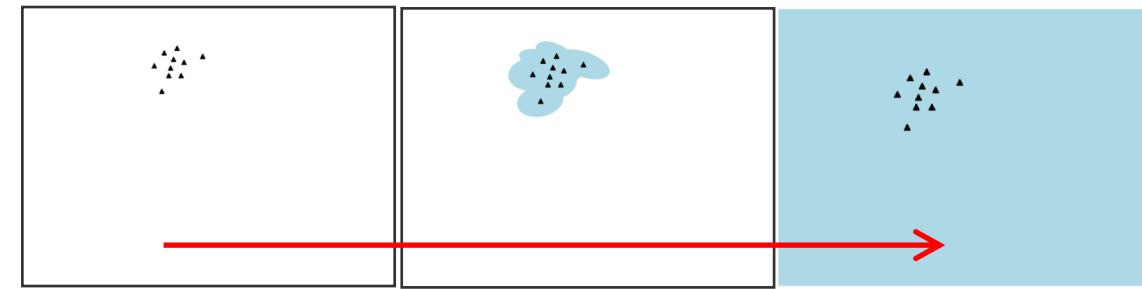
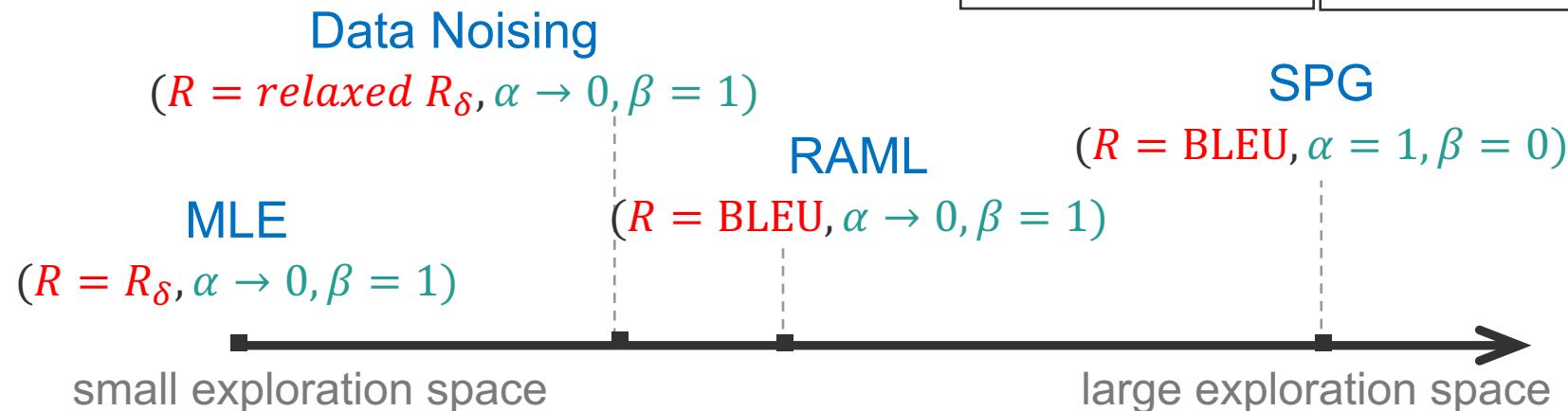


- Every algorithm corresponds to a **point** in the hyperparameter space





Interpolation algorithm

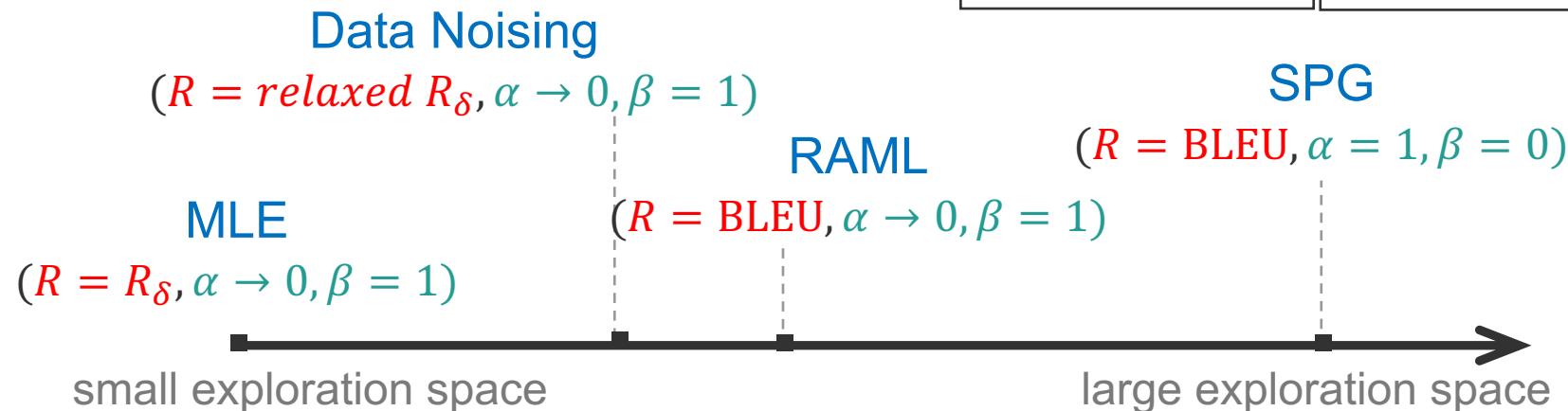
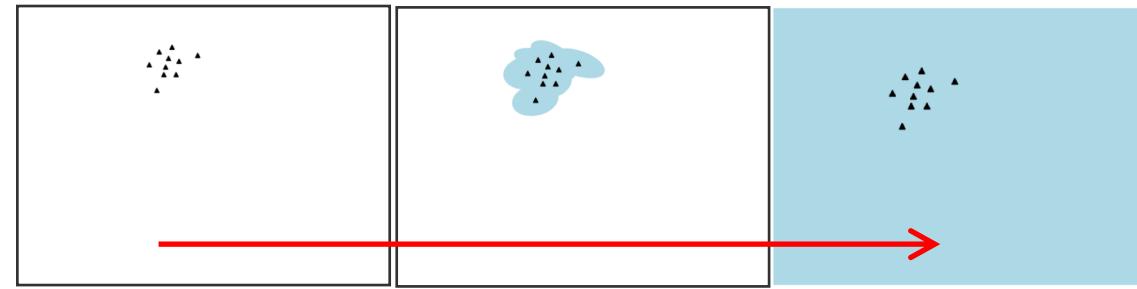


- Every algorithm corresponds to a point in the hyperparameter space
- From left to right:
 - Increasingly larger exploration space
 - Better test performance in theory
 - More difficult for training





Interpolation algorithm



- Every algorithm corresponds to a **point** in the hyperparameter space
- From left to right:
 - Increasingly larger exploration space
 - Better test performance in theory
 - More difficult for training
- Idea: **interpolating** among the algorithms
 - Start from MLE hyperparameter values, gradually anneal to SPG hyperparameter values





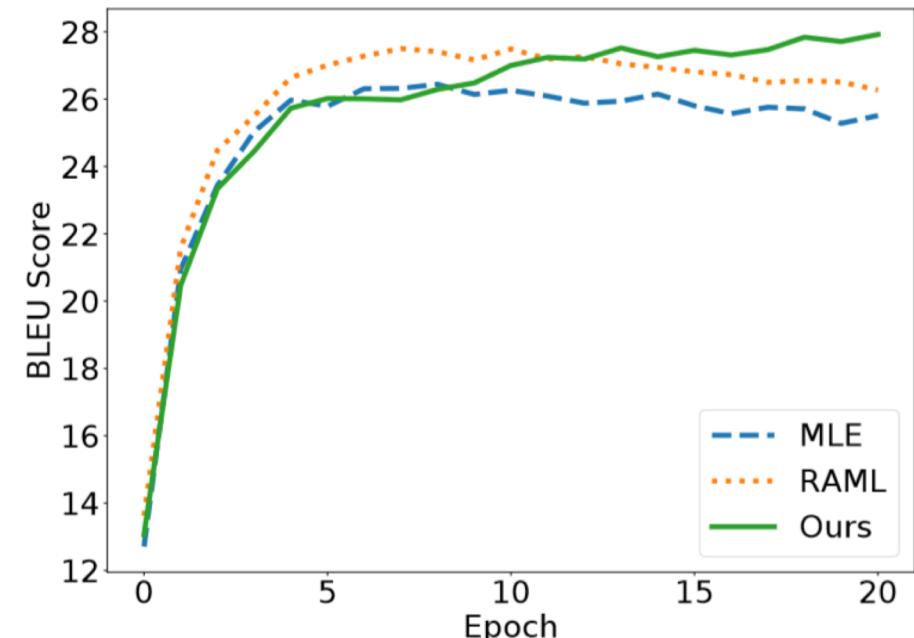
Preliminary Experimental Results

- Machine translation

Model	BLEU
MLE	26.44 ± 0.18
RAML (Norouzi et al., 2016)	27.22 ± 0.14
SPG (Ding & Soricut, 2017)	26.62 ± 0.05
MIXER (Ranzato et al., 2015)	26.53 ± 0.11
Scheduled Sampling (Bengio et al., 2015)	26.76 ± 0.17
Ours	27.86 ± 0.10

- Text Summarization

Method	ROUGE-1	ROUGE-2	ROUGE-L
MLE	36.11 ± 0.21	16.39 ± 0.16	32.32 ± 0.19
RAML (Norouzi et al., 2016)	36.30 ± 0.04	16.69 ± 0.20	32.49 ± 0.17
SPG (Ding & Soricut, 2017)	36.48 ± 0.24	16.84 ± 0.26	32.79 ± 0.26
MIXER (Ranzato et al., 2015)	36.34 ± 0.23	16.61 ± 0.25	32.57 ± 0.15
Scheduled Sampling (Bengio et al., 2015)	36.59 ± 0.12	16.79 ± 0.22	32.77 ± 0.17
Ours	36.72 ± 0.29	16.99 ± 0.17	32.95 ± 0.33





Two Central Goals

- Generating human-like, grammatical, and readable text
 - Exposure bias, criteria mismatch
 - A unified framework of sequence generation learning algorithms
 - MLE, RAML, SPG, Data Noising, Policy Gradient, ...
- Generating text that contains desired information inferred from inputs
 - Machine translation
 - Source sentence --> target sentence w/ the same meaning
 - Data description
 - Table --> data report describing the table
 - Attribute control
 - Sentiment: positive --> ``I like this restaurant''
 - Modify sentiment from positive to negative
 - Conversation control
 - Control conversation strategy and topic





Two Central Goals

- Generating human-like, grammatical, and readable text
 - Exposure bias, criteria mismatch
 - A unified framework of sequence generation learning algorithms
 - MLE, RAML, SPG, Data Noising, Policy Gradient, ...
- Generating text that contains desired information inferred from inputs
 - Machine translation
 - Source sentence --> target sentence w/ the same meaning
 - Data description
 - Table --> data report describing the table
 - Attribute control
 - Sentiment: positive --> ``I like this restaurant''
 - Modify sentiment from positive to negative
 - Conversation control
 - Control conversation strategy and topic





Two Central Goals

- Generating human-like, grammatical, and readable text
 - Exposure bias, criteria mismatch
 - A unified framework of sequence generation learning algorithms
 - MLE, RAML, SPG, Data Noising, Policy Gradient, ...
 - Generating text that contains desired information inferred from inputs
 - Machine translation
 - Source sentence --> target sentence w/ the same meaning
 - Data description
 - Table --> data report describing the table
 - Attribute control
 - Sentiment: positive --> ``I like this restaurant''
 - Modify sentiment from positive to negative
 - Conversation control
 - Control conversation strategy and topic
- #supervision data**
- | | | |
|----------------------|--------|------------------------|
| Machine translation | -----> | 10s of millions |
| Data description | -----> | 10s of 1000s |
| Attribute control | -----> | 10s of 1000s |
| Conversation control | -----> | 0 |





Two Central Goals

Controlled generation in unsupervised settings

- Generating human-like, grammatical, and readable text
 - Exposure bias, criteria mismatch
 - A unified framework of sequence generation learning algorithms
 - MLE, RAML, SPG, Data Noising, Policy Gradient, ...
 - Generating text that contains desired information inferred from inputs
 - Machine translation
 - Source sentence --> target sentence w/ the same meaning
 - Data description
 - Table --> data report describing the table
 - Attribute control
 - Sentiment: positive --> ``I like this restaurant''
 - Modify sentiment from positive to negative
 - Conversation control
 - Control conversation strategy and topic
- #supervision data**
- | | | |
|----------------------|--------|------------------------|
| Machine translation | -----> | 10s of millions |
| Data description | -----> | 10s of 1000s |
| Attribute control | -----> | 10s of 1000s |
| Conversation control | -----> | 0 |





Unsupervised Controlled Generation of Text

- Sentence-level control
 - Text attribute transfer (style transfer) [Hu et al., 2017; Yang et al., 2018]
 - Text content manipulation [Wang, Hu et al., 2019]
- Conversation-level control
 - Target-guided Open-domain Conversation





Unsupervised Controlled Generation of Text

- Sentence-level control
 - Text attribute transfer (style transfer) [Hu et al., 2017; Yang et al., 2018]
 - Text content manipulation [Wang, Hu et al., 2019]
- Conversation-level control
 - Target-guided Open-domain Conversation





Text Attribute Transfer

- Modify a given sentence to
 - Have desired attribute values
 - While keeping all other aspects unchanged
- Attribute: sentiment, tense, voice, gender, ...
- E.g., transfer sentiment from **negative** to **positive**:
 - ``It was super **dry** and had a **weird** taste to the entire slice .”
 - ``It was super **fresh** and had a **delicious** taste to the entire slice .”
- Applications:
 - Personalized article writing, conversation systems, authorship obfuscation





Text Attribute Transfer

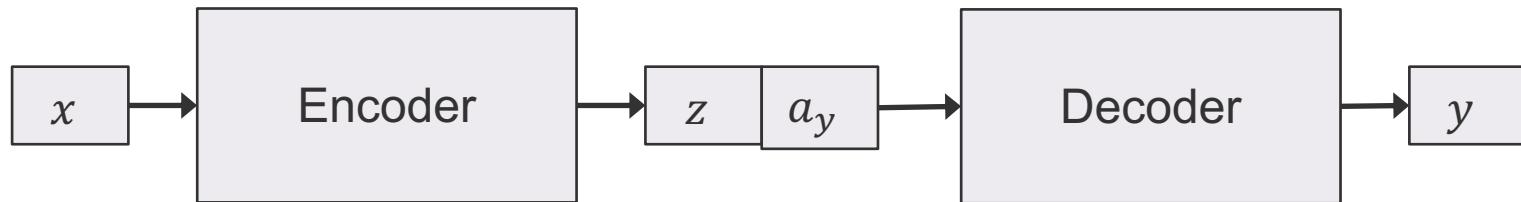
- Original sentence x , original attribute a_x
- Target sentence y , target attribute a_y
- Task: $(x, a_y) \rightarrow y$
 - y has the desired attribute a_y
 - y keeps all attribute-independent properties of x
- Usually, only have pairs of (x, a_x) , but no $((x, a_x), (y, a_y))$ for training
 - E.g., two sets of sentences: one with positive sentiment, the other with negative





Text Attribute Transfer: Solution

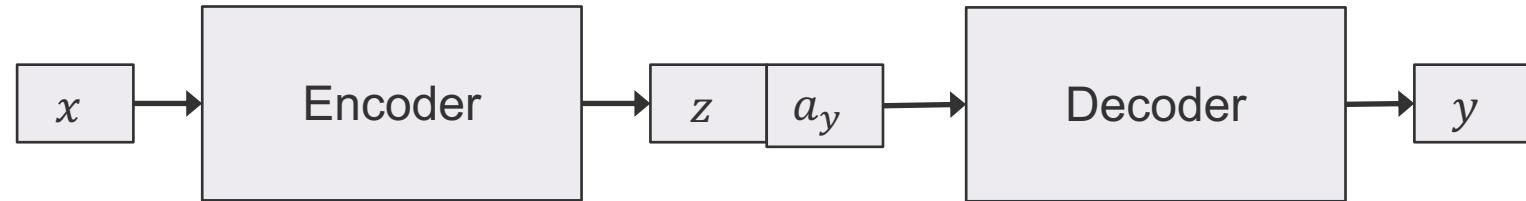
- Task: $(x, a_y) \rightarrow y$
 - y has the desired attribute a_y
 - y keeps all attribute-independent properties of x
- Model $p_\theta(y|x, a_y)$





Text Attribute Transfer: Solution

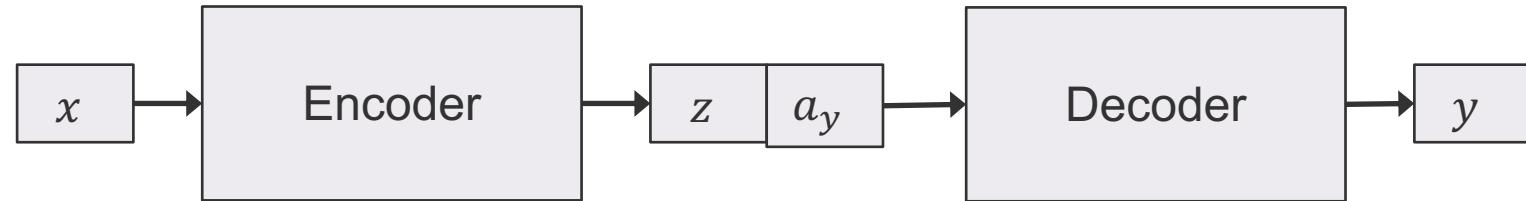
- Task: $(x, a_y) \rightarrow y$
 - y has the desired attribute a_y
 - y keeps all attribute-independent properties of x
- Model $p_\theta(y|x, a_y)$
- Key intuition for learning:
 - Decompose the task into competitive sub-objectives
 - Use direct supervision for each of the sub-objectives





Text Attribute Transfer: Solution

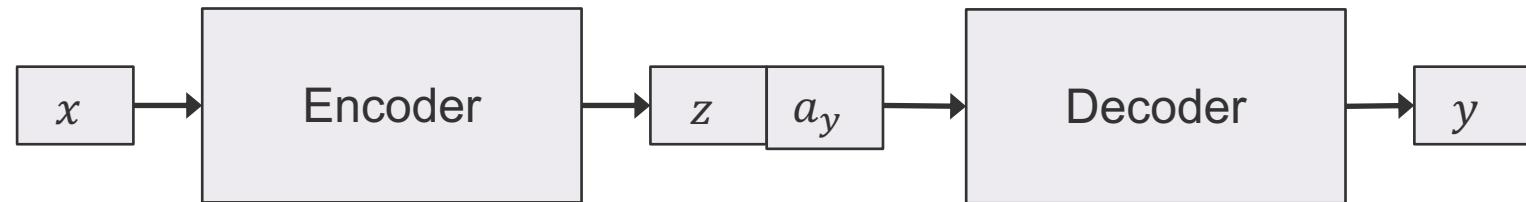
- Task: $(x, a_y) \rightarrow y$
 - y has the desired attribute a_y
 - y keeps all attribute-independent properties of x
- Model $p_\theta(y|x, a_y)$
- Key intuition for learning:
 - Decompose the task into competitive sub-objectives
 - Use direct supervision for each of the sub-objectives
- Auto-encoding loss: $(x, a_x) \rightarrow x$





Text Attribute Transfer: Solution

- Task: $(x, a_y) \rightarrow y$
 - y has the desired attribute a_y
 - y keeps all attribute-independent properties of x



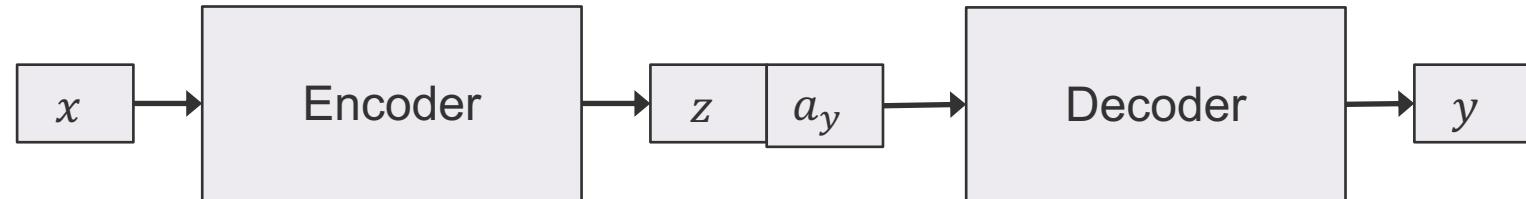
- Model $p_\theta(y|x, a_y)$
- Key intuition for learning:
 - Decompose the task into competitive sub-objectives
 - Use direct supervision for each of the sub-objectives
- Auto-encoding loss: $(x, a_x) \rightarrow x$
- Classification loss: $\hat{y} \sim p_\theta(y|x, a_y), f(\hat{y}) \rightarrow a_y$
 - where f is a pre-trained attribute classifier





Text Attribute Transfer: Solution

- Task: $(x, a_y) \rightarrow y$

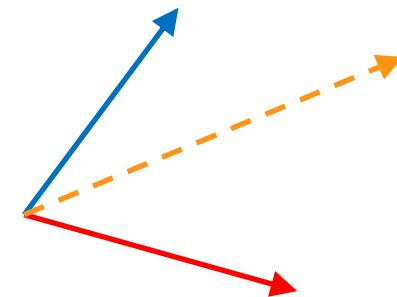


- y has the desired attribute a_y
 - y keeps all attribute-independent properties of x

- Model $p_\theta(y|x, a_y)$
- Key intuition for learning:

- Decompose the task into competitive sub-objectives
 - Use direct supervision for each of the sub-objectives

- Auto-encoding loss: $(x, a_x) \rightarrow x$
- Classification loss: $\hat{y} \sim p_\theta(y|x, a_y), f(\hat{y}) \rightarrow a_y$
 - where f is a pre-trained attribute classifier
- The above two losses are competitive; minimize jointly to avoid collapse





Text Attribute Transfer: Results & Improvement

- Performance on sentiment:
 - Accuracy: 92%
 - BLEU against input sentence: 54





Text Attribute Transfer: Results & Improvement

- Performance on sentiment:
 - Accuracy: 92%
 - BLEU against input sentence: 54
- Problem:
 - Language quality is often not good
 - LM perplexity: 239.8

Original: if i could give them a zero star review i would !
Output: if i **lite** give them a **sweetheart** star review i would !

Original: uncle george is very friendly to each guest
Output: uncle george is very **lackluster** to each guest

Original: the food is fresh and the environment is good
Output: the food is **atrocious** and the environment is **atrocious**





Text Attribute Transfer: Results & Improvement

- Performance on sentiment:
 - Accuracy: 92%
 - BLEU against input sentence: 54
- Problem:
 - Language quality is often not good
 - LM perplexity: 239.8
- Improvement:
 - Use an LM as a direct supervision!
 - $\hat{y} \sim p_\theta(y|x, a_y)$, $\max_\theta \text{LM}(\hat{y})$
 - Accuracy: 91%
 - BLEU against input sentence: 57
 - LM perplexity: 60.9

Original: if i could give them a zero star review i would !
Output: if i **lite** give them a **sweetheart** star review i would !

Original: uncle george is very friendly to each guest
Output: uncle george is very **lackluster** to each guest

Original: the food is fresh and the environment is good
Output: the food is **atrocious** and the environment is **atrocious**





Text Attribute Transfer: Results & Improvement

- Performance on sentiment:
 - Accuracy: 92%
 - BLEU against input sentence: 54
- Problem:
 - Language quality is often not good
 - LM perplexity: 239.8
- Improvement:
 - Use an LM as a direct supervision!
 - $\hat{y} \sim p_\theta(y|x, a_y)$, $\max_\theta \text{LM}(\hat{y})$
 - Accuracy: 91%
 - BLEU against input sentence: 57
 - LM perplexity: 60.9

Original: if i could give them a zero star review i would !
Output: if i **lite** give them a **sweetheart** star review i would !
+ LM: if i can give them a great star review i would !

Original: uncle george is very friendly to each guest
Output: uncle george is very **lackluster** to each guest
+ LM: uncle george is very rude to each guest

Original: the food is fresh and the environment is good
Output: the food is **atrocious** and the environment is **atrocious**
+ LM: the food is bland and the environment is bad .





Unsupervised Controlled Generation of Text

- Sentence-level control
 - Text attribute transfer (style transfer) [Hu et al., 2017; Yang et al., 2018]
 - Text content manipulation [Wang, Hu et al., 2019]
- Conversation-level control
 - Target-guided Open-domain Conversation

Key idea:

- Decompose the task into **competitive** sub-objectives
- Use **direct supervision** for each of the sub-objectives





Unsupervised Controlled Generation of Text

- Sentence-level control
 - Text attribute transfer (style transfer) [Hu et al., 2017; Yang et al., 2018]
 - Text content manipulation [Wang, Hu et al., 2019]
- Conversation-level control
 - Target-guided Open-domain Conversation

Key idea:

- Decompose the task into **competitive** sub-objectives
- Use **direct supervision** for each of the sub-objectives





Text Content Manipulation

- Generate a sentence to describe content in a given data record
- But language is rich with variation -- there are diverse possible ways of saying the same content (writing style):
 - word choice, expressions, transitions, tones, ...

Content Record	PLAYER LeBron_James	PT 32	RB 4	AS 7	PLAYER Kyrie_Irving	PT 20
Reference Sentence	Jrue_Holiday led the way with 26 points and 6 assists , while Goran_Dragic scored 23 points and pulled down 8 rebounds .					
Output	LeBron_James led the way with 32 points , 7 assists and 4 rebounds , while Kyrie_Irving scored 20 points .					





Text Content Manipulation

- Generate a sentence to describe content in a given data record
- But language is rich with variation -- there are diverse possible ways of saying the same content (writing style):
 - word choice, expressions, transitions, tones, ...
- We want to control the writing style: use the writing style of a reference sentence

Content Record	PLAYER LeBron_James	PT 32	RB 4	AS 7	PLAYER Kyrie_Irving	PT 20
Reference Sentence	Jrue_Holiday led the way with 26 points and 6 assists , while Goran_Dragic scored 23 points and pulled down 8 rebounds .					
Output	LeBron_James led the way with 32 points , 7 assists and 4 rebounds , while Kyrie_Irving scored 20 points .					





Text Content Manipulation - Results

Content x	PLAYER Gerald_Henderson	PTS 17	FGM 6	FGA 13	FG3M 1	FG3A 2	FTM 4	FTA 4	AST 5
Reference y'	Kawhi_Leonard also had a solid offensive game , scoring 16 points (7 - 13 FG , 0 - 1 3Pt , 2 - 5 FT) and adding 5 assists and 5 rebounds .								
Rule-based	Gerald_Henderson also had a solid offensive game , scoring 17 points (6 - 13 FG , 1 - 2 3Pt , 4 - 4 FT) and adding 5 assists and 5 rebounds .								
AdvST	Gerald_Henderson also had a solid offensive game , scoring 13 points (13 - 13 FG , 2 - 2 3Pt , 4 - 4 FT) and adding 5 assists and 5 rebounds .								
Ours w/o Cover.	Gerald_Henderson also had a solid offensive game , scoring 17 points (6 - 13 FG , 1 - 2 3Pt , 4 - 4 FT) and adding 5 assists and 5 rebounds .								
Ours	Gerald_Henderson also had a solid offensive game , scoring 17 points (6 - 13 FG , 1 - 2 3Pt , 4 - 4 FT) and adding 5 assists .								





Text Content Manipulation - Results

	Model	Precision %	Content Recall %	Style BLEU
1	AttnCopy-S2S	88.71±2.45	60.64±1.31	39.15±5.48
	Rule-based	62.63	63.64	100
2	MAST	33.15±0.78	31.09±0.63	95.29±2.53
	AdvST	66.51±1.08	56.03±0.56	72.22±1.47
3	Ours w/o Cover.	75.61±1.03	62.93±0.53	75.09±2.17
	Ours	78.31±0.94	65.64±0.47	80.83±1.89





Unsupervised Controlled Generation of Text

- Sentence-level control
 - Text attribute transfer (style transfer) [Hu et al., 2017; Yang et al., 2018]
 - Text content manipulation [Wang, Hu et al., 2019]
- Conversation-level control
 - Target-guided Open-domain Conversation

Key idea:

- Decompose the task into **competitive** sub-objectives
- Use **direct supervision** for each of the sub-objectives





Unsupervised Controlled Generation of Text

- Sentence-level control
 - Text attribute transfer (style transfer) [Hu et al., 2017; Yang et al., 2018]
 - Text content manipulation [Wang, Hu et al., 2019]
- Conversation-level control
 - Target-guided Open-domain Conversation

Key idea:

- Decompose the task into **competitive** sub-objectives
- Use **direct supervision** for each of the sub-objectives





Target-guided Open-domain Conversation

- Task-oriented dialog:
 - Address a specific task, e.g., booking a flight
 - Close domain
- Open-domain chit-chat:
 - Improve user engagement
 - Random conversation, hard to control
- Target-guided conversation:
 - Open-domain conversation
 - Controlled conversation strategy to reach a *desired topic* in the end of conversation
 - Applications:
 - Bridges task-oriented dialog and open-domain chit-chat
 - Conversational recommender system, education, psychotherapy





Target-guided Open-domain Conversation

- Two goals:
 - Starting from any topic, reach a desired topic in the end of conversation
 - Natural conversation: smooth transition

Target: **e-books**

Agent : hi how are you doing

Human : not so good . i am really tired

Agent : oh i'm sorry to hear

Human : i have a lot of work to do

Agent : what kind was it ?

Human : i am writing a chatbot program

Agent : interesting . i read about coding from a book

Human: really

Agent : i prefer **e-books** over paperback book





Target-guided Open-domain Conversation

- Two goals:
 - Starting from any topic, reach a desired topic in the end of conversation
 - Natural conversation: smooth transition

Target: **e-books**

Agent : hi how are you doing

tired Human : not so good . i am really tired

sorry Agent : oh i'm sorry to hear

work Human : i have a lot of work to do

work Agent : what kind was it ?

program Human : i am writing a chatbot program

coding, book Agent : interesting . i read about coding from a book

Human: really

e-books Agent : i prefer e-books over paperback book





Target-guided Open-domain Conversation

- Two goals:
 - Starting from any topic, reach a desired topic in the end of conversation
 - Natural conversation: smooth transition

Target: **e-books**

Agent : hi how are you doing

tired Human : not so good . i am really tired

sorry Agent : oh i'm sorry to hear

work Human : i have a lot of work to do

work Agent : what kind was it ?

program Human : i am writing a chatbot program

coding, book Agent : interesting . i read about coding from a book

e-books Human: really

↓ Agent : i prefer e-books over paperback book

Challenge: No supervised data for the task

Solution: Use competitive sub-objectives and partial supervision

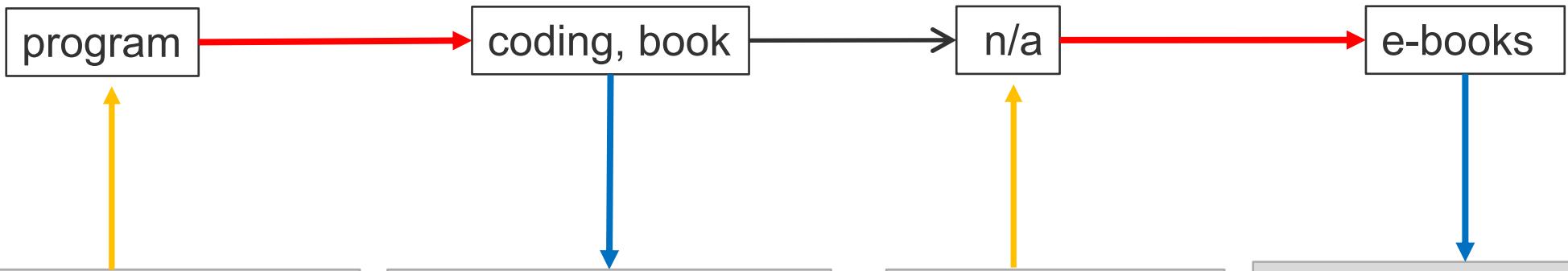
- **Natural conversation:** rich chit-chat data to learn smooth **single-turn** transition
- **Reaching desired target:** rule-based **multi-turn** planning





Target-guided Open-domain Conversation

Keywords:



Utterance:

Human: i am writing a
chatbot program

Agent: interesting . i
read about coding
from a book

Human: really

Agent: i prefer e-
books over
paperback book

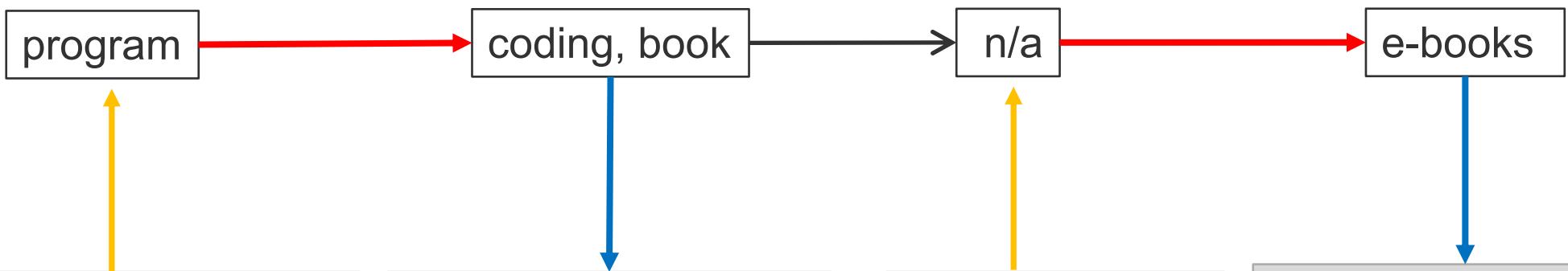




Target-guided Open-domain Conversation

- → keyword extraction

Keywords:



Utterance:

Human: i am writing a
chatbot program

Agent: interesting . i
read about coding
from a book

Human: really

Agent: i prefer e-
books over
paperback book

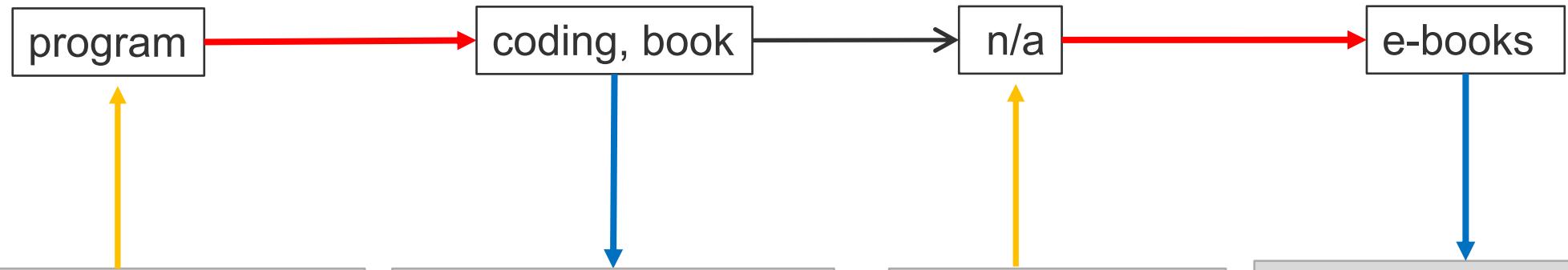




Target-guided Open-domain Conversation

- → keyword extraction
- → keyword conditional response retrieval

Keywords:



Utterance:

Human: i am writing a
chatbot program

Agent: interesting . i
read about coding
from a book

Human: really

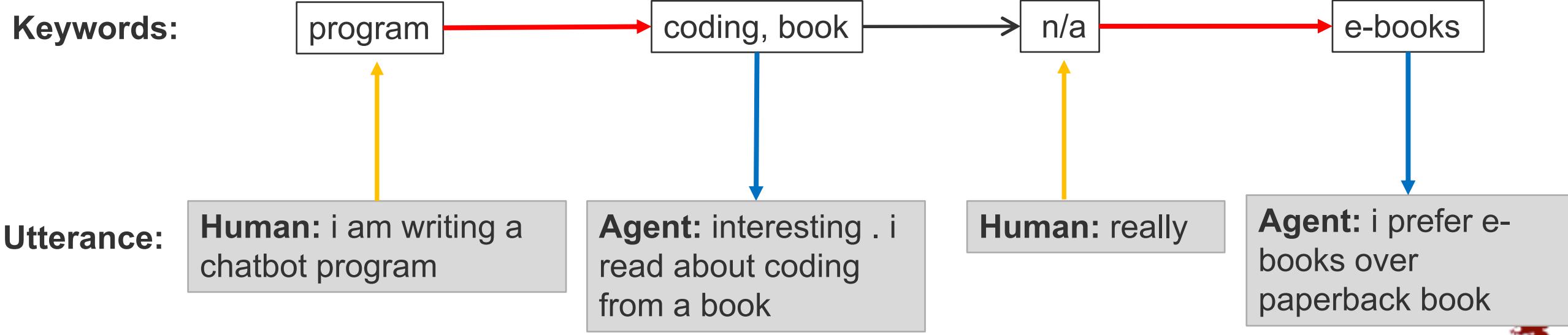
Agent: i prefer e-
books over
paperback book





Target-guided Open-domain Conversation

- → keyword extraction
- → keyword conditional response retrieval
- → learned kernel-based topic transition + target-guided rule:
 - (1) transit to keywords that are close in the word embedding space – smooth
 - (2) next keywords must get closer to the target keyword – target-guided
 - Tune relative weight of (1) and (2) to control aggressiveness





Example Results

Target: dance

Agent : hi how is it going with you

Human : fine . I'm playing sports

Agent : i play basketball . u play ?

Human : yeah i also like basketball .

Agent : you like rap music ? i listen to a lot of rap music

Human : no i like classic music

Agent : its a taylor swift song

Human : taylor swift is a great singer

Agent : sing to me , what are you going to sing

Human : i know nothing about singing

Agent : oh i think i can dance lol





Example Results – Failure Case

Target: listen

Agent : hi ! how are you ? i am very exciting about my new job !

Human : fine . what is your new job ?

Agent : that sounds exciting .

Human : i like read books

Agent : what kind of books do you read ?

Human : i read comics

Agent : i do not have tv

Human : watch anime is fun

Agent : what position d you play ?

Human : i play a lot of sports

Agent : oh really ? what kind of music . i listen to classical





Unsupervised Controlled Generation of Text

- Sentence-level control
 - Text attribute transfer (style transfer) [Hu et al., 2017; Yang et al., 2018]
 - Text content manipulation [Wang, Hu et al., 2019]
- Conversation-level control
 - Target-guided Open-domain Conversation

Key idea:

- Decompose the task into **competitive** sub-objectives
- Use **direct supervision** for each of the sub-objectives





Two Central Goals

- Generating human-like, grammatical, and readable text
 - I.e., generating **natural** language
- Generating text that contains desired information inferred from inputs
 - Machine translation
 - Source sentence --> target sentence w/ the same meaning
 - Data description
 - Table --> data report describing the table
 - Attribute control
 - Sentiment: positive --> ``I like this restaurant''
 - Conversation control
 - Control conversation strategy and topic

