Paraphrase Generation with Latent Bag of Words

Yao Fu, Yansong Feng, John P. Cunningham
Columbia University | Peking University

Hareesh Bahuleyan
Borealis AI

Problem Setting

- To change the sentence structure and/or expression, while conveying the same meaning
- **Parallel corpus**
  - For each input there are K paraphrases available for training
  - *How do I improve my English? | What is the best way to learn English?*
- **Input/Output**
  - $(x_1, x_2, \ldots, x_m) \rightarrow (y_1, y_2, \ldots, y_n)$

Modelling Approaches

- **Traditionally** —> rule-based: find lexical substitutions from WordNet
  - Designing rules is not scalable
- **Recent neural models** —> seq2seq learning framework
  - Not interpretable as to why the model produces certain output

How to improve interpretability?

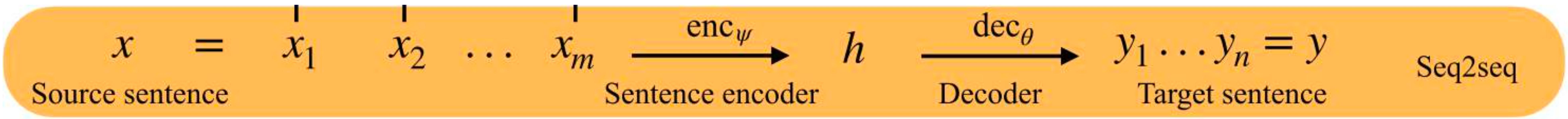
- Separate the generation process into two steps:
  - **Content Planning**: what to say?
  - **Surface realization**: how to say it?
- Example: Image Captioning
- For paraphrase generation in the traditional setting, it can be achieved as follows:
  - word neighbours are *retrieved* from WordNet (the planning stage)
  - then words are *substituted and re-organized* to form a paraphrase (the realization stage)
  - *"neighbours"* of a given word refer to words that are semantically close to the given word (e.g. improve → learn)

Combining the 2-step process

- Separation of planning and realization can result in ***non-differentiable*** process and thus not possible to do end-to-end training
- In this paper:
  - optimize a discrete latent variable (z) that represents bag-of-words information
  - z is grounded with explicit lexical semantics (from the target)
  - use z to guide the decoder's generation process
- Their model follows the planning and realization steps, yet fully differentiable

And how is it done?

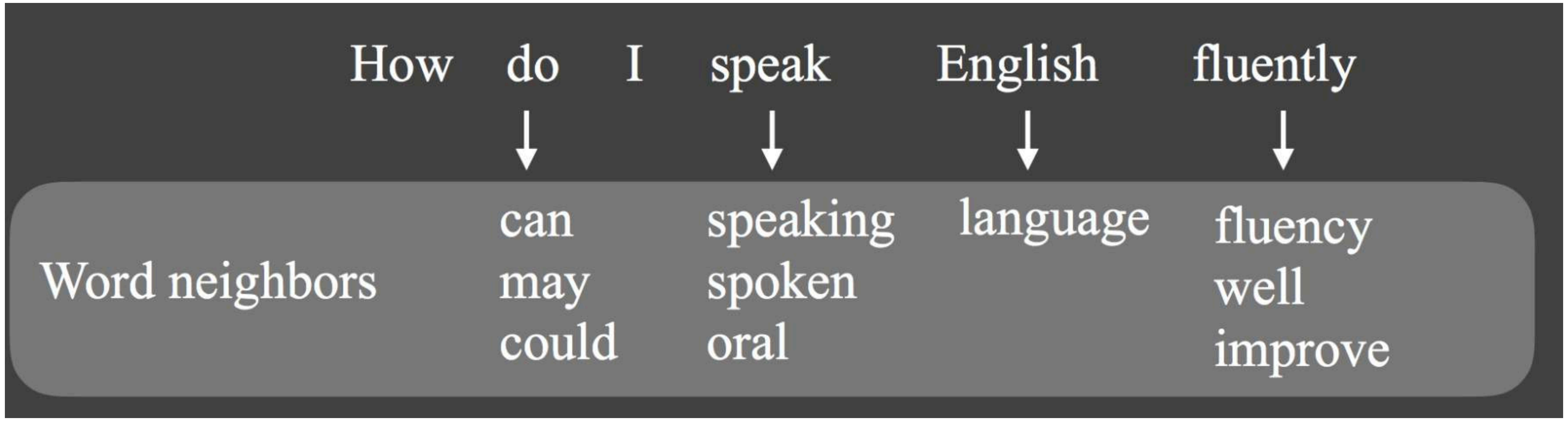
Start with Seq2Seq



- LSTM Encoder-Decoder Architecture
- Cross Entropy Loss

$$
\begin{aligned}
h &= \text{enc}_\psi(x) \\
p(y|x) &= \text{dec}_\theta(h) \\
\mathcal{L}_{\text{S2S}} &= \mathbb{E}_{(x^\star, y^\star) \sim \mathbb{P}^\star}[-\log p_\theta(y^\star | x^\star)]
\end{aligned}
$$

Predict neighbour words for source tokens



- For each source token, predict L different neighbour (present in the model vocabulary)
  - $p(z_{ij} | x_i) =$ Categorical$(\phi_{ij}(x))$; z is a vector of probabilities
  - ϕ_{ij} is parameterized by a neural network: hidden states -> softmax over vocabulary

Mix the probabilities from all source neighbours

$$\tilde{z} \sim p_\phi(\tilde{z}|x) = \frac{1}{ml} \sum_{i,j} p(z_{ij}|x_i)$$

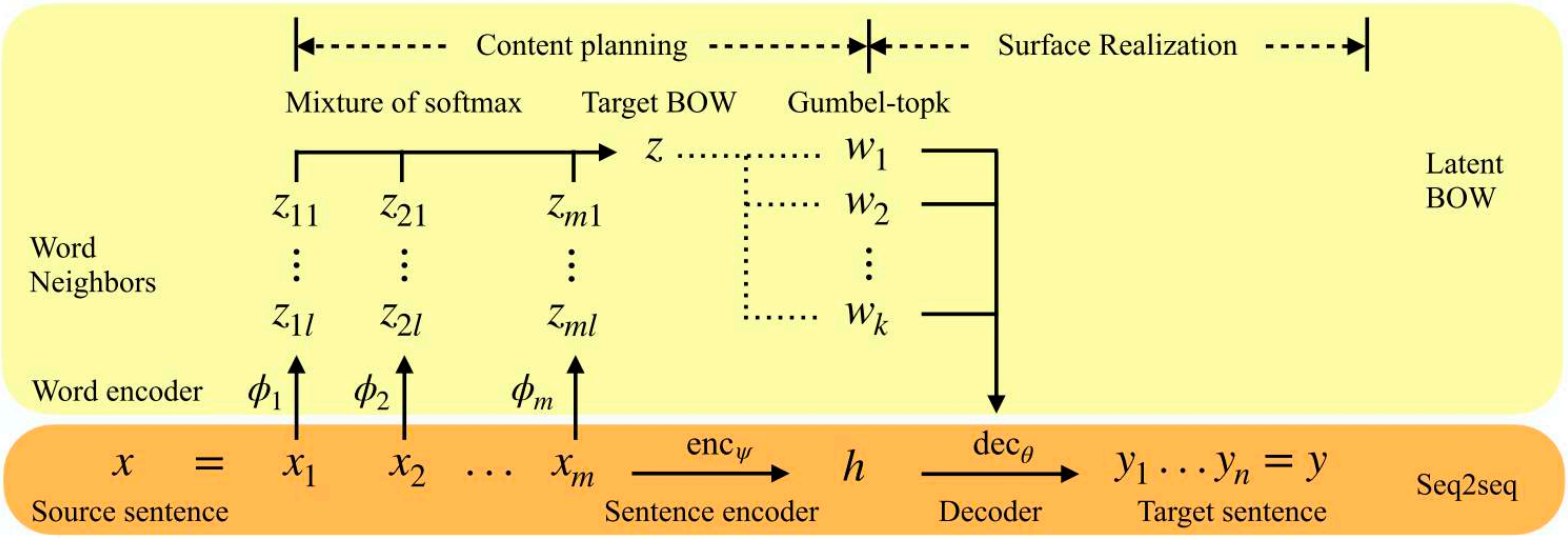
- where ml is the maximum number of predicted words.
- $\tilde{z}$ is a categorical variable
  - which represents a mixture of probabilities
  - of all neighbors of all source words
- one source sentence may correspond to multiple target sentences.
  - optimize $\tilde{z}$ to be close to the target BOW —> words from all target sentences

Sampling from the categorical distribution

- Categorical distribution over the words in the vocabulary: $p_\phi(\tilde{z}|x)$
- Construct the bag of words:
  - Sample k times without replacement (Content Planning)
- Use the (weighted) average of the embeddings of the k sampled words as input to the decoder
- Decoding process (Surface Realization):

$$y \sim p_\theta(y|x, z) = \mathrm{dec}_\theta(x, z)$$

Model



Content planning
Surface Realization
Mixture of softmax
Target BOW
Gumbel-topk
Latent BOW
Word Neighbors
Word encoder
Source sentence
Sentence encoder
Decoder
Target sentence
Seq2seq

Loss Function

$$\mathcal{L}_{\mathrm{S2S'}} = \mathbb{E}_{(x^\star,y^\star)\sim\mathbb{P}^\star, z\sim p_\phi(\tilde{z}|x)}[-\log p_\theta(y^\star|x^\star, z)]$$
$$\mathcal{L}_{\mathrm{BOW}} = \mathbb{E}_{z^*\sim\mathbb{P}^*}[-\log p_\phi(z^*|x)]$$
$$\mathcal{L}_{\mathrm{tot}} = \mathcal{L}_{\mathrm{S2S'}} + \mathcal{L}_{\mathrm{BOW}}$$

- Additional BOW regularization term
  - encourages the model to assign high probability to the BOWs present in ALL the target sentences corresponding to input x
- In regular seq2seq setting, the NLL loss forces the generation to be close to the the current target
- With the BOW-loss, the model is encouraged to use information from all the targets (i.e., learning happens at the **corpus** level, rather than **sentence** level)
- The total loss to optimize over the:
  - Encoder parameters ψ
  - Decoder parameters θ
  - Hidden state to neighbouring word FF layers φ

Gumbel-Softmax Reparameterization Trick

- Gumbel-Max trick:
  - efficient way to draw samples z from the Categorical distribution with class probabilities π_i

$$z = \texttt{one\_hot}\left(\arg\max_i \left[g_i + \log \pi_i\right]\right)$$

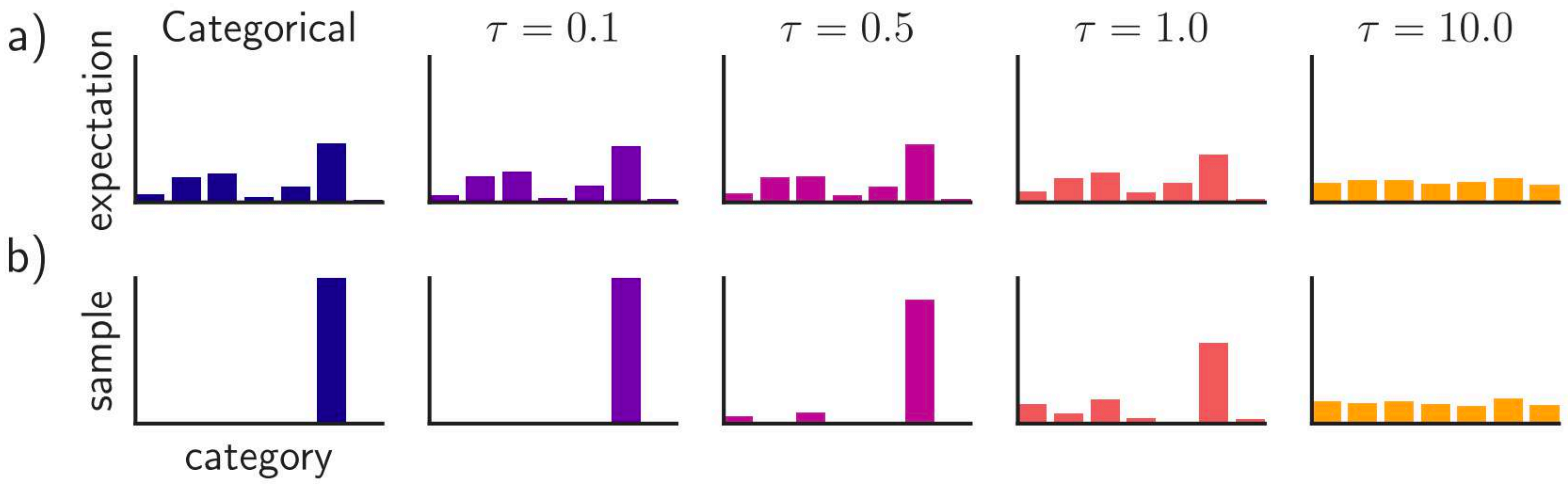
  - where, $g = -\log(-\log(u))$ and $u \sim$ Uniform(0,1)
  - argmax is not differentiable
  - **Softmax-Approximation:**

$$y_i = \frac{\exp((\log(\pi_i) + g_i)/\tau)}{\sum_{j=1}^{k} \exp((\log(\pi_j) + g_j)/\tau)} \qquad \text{for } i = 1, ..., k.$$

Categorical Reparameterization with Gumbel-Softmax. Jang et. al (2017)

Temperature Parameter

- τ controls the *peakiness* of the distribution
- Start with large τ (uniform distribution) and move towards small τ (peaky distribution) as training progresses



Categorical Reparameterization with Gumbel-Softmax. Jang et. al (2017)

Experiments

- **Datasets**:
  - Quora Questions
  - MSCOCO - 5 different captions about the same image
- **Baselines**:
  - Seq2Seq
  - β-VAE

Model Variants

- **LBOW-TopK:** directly choose the most k probable words from the BOW distribution
- **LBOW-Gumbel:** sample from the BOW distribution with Gumbel reparameterization, thus injecting randomness into the model
- **BOW-Hard (lower bound):** Optimize the encoder (with BOW loss) and decoder (with NLL loss) separately
- **Cheating BOW (upper bound):** No sampling, but use the BOW of the actual target sentences during generation

Results

Table 1: Results on the Quora and MSCOCO dataset. B for BLEU and R for ROUGE.

Quora

Model	B-1	B-2	B-3	B-4	R-1	R-2	R-L
Seq2seq[40]	54.62	40.41	31.25	24.97	57.27	33.04	54.62
Residual Seq2seq-Attn [40]	54.59	40.49	31.25	24.89	57.10	32.86	54.61
β-VAE, $\beta = 10^{-3}$[17]	43.02	28.60	20.98	16.29	41.81	21.17	40.09
β-VAE, $\beta = 10^{-4}$[17]	47.86	33.21	24.96	19.73	47.62	25.49	45.46
BOW-Hard (lower bound)	33.40	21.18	14.43	10.36	36.08	16.23	33.77
LBOW-Topk (ours)	**55.79**	**42.03**	**32.71**	**26.17**	**58.79**	**34.57**	**56.43**
LBOW-Gumbel (ours)	55.75	41.96	32.66	26.14	58.60	34.47	56.23
RbM-SL[26]	-	43.54	-	-	64.39	38.11	-
RbM-IRL[26]	-	43.09	-	-	64.02	37.72	-
Cheating BOW (upper bound)	72.96	61.78	54.40	49.47	72.15	52.61	68.53

MSCOCO

Model	B-1	B-2	B-3	B-4	R-1	R-2	R-L
Seq2seq[40]	69.61	47.14	31.64	21.65	40.11	14.31	36.28
Residual Seq2seq-Attn [40]	71.24	49.65	34.04	23.66	41.07	15.26	37.35
β-VAE, $\beta = 10^{-3}$[17]	68.81	45.82	30.56	20.99	39.63	13.86	35.81
β-VAE, $\beta = 10^{-4}$[17]	70.04	47.59	32.29	22.54	40.72	14.75	36.75
BOW-Hard (lower bound)	48.14	28.35	16.25	9.28	31.66	8.30	27.37
LBOW-Topk (ours)	**72.60**	**51.14**	**35.66**	**25.27**	42.08	**16.13**	**38.16**
LBOW-Gumbel (ours)	72.37	50.81	35.32	24.98	**42.12**	16.05	38.13
Cheating BOW (upper bound)	80.87	75.09	62.24	52.64	49.95	23.94	43.77

* [26] external data used as negative samples

Model Interpretability

Quora

Input	why	do	people	ask	questions	on	quora	instead	of	googling	it
Neighbor				*post*	*quora*		*quora*			*google*	
				answer	*questions*		*questions*			*search*	
BOW sample	*ask, quora, people, questions, google, googling, easily, googled, search, answer*										
Output	why do people ask questions on quora that can be easily found on a google search ?										

Input	how	do	i	talk	english	fluently	?
Neighbor				*speak*	*english*	*fluently*	
				better	*improve*	*confidence*	
BOW sample	*english, speak, improve, fluently, talk, spoken, better, best, confidence*						
Output	how can i improve my english speaking ?						

MSCOCO

Input	A	tennis	player	is	walking	while	holding	his	racket
Neighbor		*court*	*holding*		*walks*		*carrying*		*court*
		racket	*man*		*across*		*holds*		*racquet*
BOW sample	*holding, man, tennis, walking, racket, court, player, racquet, male, woman, walks*								
Output	A man holding a tennis racquet on a tennis court								

Input	A	big	airplane	flying	in	the	blue	sky
Neighbor		*large*	*airplane*	*sky*			*blue*	*clear*
		large	*jet*	*airplane*			*clear*	*flying*
BOW sample	*blue, airplane, flying, large, plane, sky, clear, air, flies, jet*							
Output	A large jetliner flying through a blue sky							

word morphology | synonym | entailment | metonymy

Figure 2: Sentence generation samples. Our model exhibits clear interpretability with three generation steps: (1) generate the neighbors of the source words (2) sample from the neighbor BOW (3) generate from the BOW sample. Different types of learned lexical semantics are highlighted.

- Unsupervised learning of word neighbours
- Separating out content planning and surface realization

BOW prediction performance and utilization

Dataset	Performance		BOW utilization		
	Precision	Recall	# words from BOW	# words from LM	% BOW words
MSCOCO	59.41	39.54	6.75	11.66	57.89
Quora	46.99	80.32	6.88	13.84	49.71

Performance and utilization of the BOW

Input	why do	people	love	pokemon	go so	much
Neighbor		*people*	*like*	*manaply*	*going*	*spending*
			love	*pokémon*	*pokemon*	
Reference	what makes pokémon go so popular					

An example of corpus level word neighbors. The learned neighbors are from other training instances, not from this particular instance

- The model heavily uses the predicted BOW
- More than 50% of the decoder's word choices come from the BOW
- Indication the BOW prediction accuracy is essential to a good generation quality (help in reducing the search space)

Controlled Generation

Input	A man on a motorcycle with a bird on the handle
BOW sample 1	man motorcycle *sitting*
Output 1	A man is *sitting* on a motorcycle
BOW sample 2	man motorcycle *riding* *road*
Output 2	A man *riding* a motorcycle on a dirt *road*

Input	A man wearing a red tie holding it to show people
BOW sample 1	man suit tie
Output 1	A man wearing a suit and tie
BOW sample 2	man suit tie *holding* *picture*
Output 2	A man wearing *a* suit and tie is *holding* a picture

- In VAEs —> semantics cannot be directly controlled in the latent space.
- Needs to be done from a **geometric perspective** (latent vector arithmetic).
  - positive to negative sentence: Subtract the "positive" vector and add the "negative" vector
- Here it can interpreted from a **lexical semantics perspective** - by modifying the BOWs vector to contain the desired words in the output.

Summary

- LBOW model to bridge content planning and surface realization
- End-to-end training possible with Gumbel-Softmax reparameterization trick
- Improved performance on paraphrase generation
- Better interpretability and controlled generation with the BOWs latent variable

Thank You