论文笔记《Paraphrase generation with latent bag of words》

论文来源：2019 NIPS
论文代码：https://github.com/FranxYao/dgm_latent_bow

论文主要内容

作者提出了一个a latent bag of words(BOW)模型来进行paraphrase生成。作者首先使用source word取预测他们自己的邻近词，然后用他们的softmax分布去预测target word，在从softmax分布中取词时，作者采用了Gumble top-k的重参化技巧。

首先，传统的使用词替换的方式来生成paraphrase，主要分为两步：plan stage和relization stage

- plan stage：从wordNet中找到source word的邻近词
- realization stage：把词替换掉，并且从组织新句子

作者使用来自source sentence中的词去预测他们的邻近词，把target sentence中的词作为target BOW，这一步可以看做是plan stage。从plan stage的所有词中,sample出一个词子集，然后去重新组织这些词，形成新的句子，这就是realization stage。在sample这一步中，作者使用了Gumble top-k 重参化的技巧。

模型

模型主要分为两部分： the sequence to sequence base model, bag of words for content planning

the sequence to sequence base model

传统的seq2seq模型的框架，encoder和decoder，其中encoder和decoder采用的都是LSTM

$$
\begin{aligned}
h &= \text{enc}_\psi(x) \\
p(y|x) &= \text{dec}_\theta(h) \\
\mathcal{L}_{\text{S2S}} &= \mathbb{E}_{(x^\star, y^\star)\sim\mathbb{P}^\star}[-\log p_\theta(y^\star|x^\star)],
\end{aligned} \tag{1}
$$

bag of words for content planning

作者有一个假设，就是说从target sentence构建的BOW，应该和source sentence的邻近词构建的BOW基本相似。
首先第一步，对于每一个source word获取他们的邻近词表示，对于$wordx_i$，它的邻近词z_{ij}是一个V维的(V代表词表大小)one-hot向量，并且每一个词的邻近词数目固定为l个，source word 共有m个。

$$p(z_{ij}|x_i) = Categorical(\phi_{ij}(x_i))$$

其实在实现时，作者是在encoder时，对于LSTM的每一个隐层输出，作者采用了一个softmax层接在每个隐层输出的后面，然后由于固定了每个词的邻近词数目l个，所以可以从词表V中选出每个词的l个邻近词。
然后把这ml个邻近词混合在一起进行表示【每个source word的邻近词都是一个one-hot表示的V维向量】,得到一个向量 $\tilde{z}$,其实就是ml个向量相加在求平均值

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

这时得到的$\tilde{z}$是一个V维的向量，其中每一维i上的值(i=1,2...V)，可以看做是在词表V中取第i个词的概率，作者这里采用Gumbel top-k的方法来获取k个概率最大的词，具体的方法就是：

- $\tilde{z}$的第i维的值，设为π_i,常规的方法就是对π_i进行排序，取分值最高的k个
- 作者在这里引入了Gumbel top-k的技巧，来引入随机性，具体就是用如下两个公式实现的，引入了随机变量g_i
    - $$a_i = log\pi_i + g_i$$
    - $$g_i \sim Gumbel(0,1)$$
- 然后取分值最高的k个a_i作为邻近词，然后去词表中检索这k个词对应的词向量$w_1, w_2 ... w_k$，并和他们的权重π_i相乘再相加，最后求平均，得到一个最终的所有邻近词的一个融合的权重词向量表示。
- 最后把这个融合的权重词向量表示和encoder得到的句子隐向量h作为decoder部分LSTM的初始状态

所以，上面的过程可以认为是选择k个邻近词的过程，可以用如下的一个公式代替：

$$z \sim p_\phi(\tilde{z}|x)(samplektimeswithoutreplacement)$$

$$y \sim p_\theta(y|x,z) = dec_\theta(x,z)$$

最终的优化部分，可以看做由两部分组成，优化$p(y|x,z)$和$p_{\tilde{z}}(\tilde{z}|x)$这两个负对数似然函数。

$$\begin{aligned}\mathcal{L}_{\text{S2S}'} &= \mathbb{E}_{(x^\star,y^\star)\sim\mathbb{P}^\star,z\sim p_\phi(\tilde{z}|x)}[-\log p_\theta(y^\star|x^\star,z)] \\ \mathcal{L}_{\text{BOW}} &= \mathbb{E}_{z^*\sim\mathbb{P}^*}[-\log p_\phi(z^*|x)] \\ \mathcal{L}_{\text{tot}} &= \mathcal{L}_{\text{S2S}'} + \mathcal{L}_{\text{BOW}}\end{aligned} \tag{5}$$

其中，P^*是从target sentence中获得的BOW的分布，z^*是target BOW的一个k-hot的向量表示；

整个模型的结构如下：

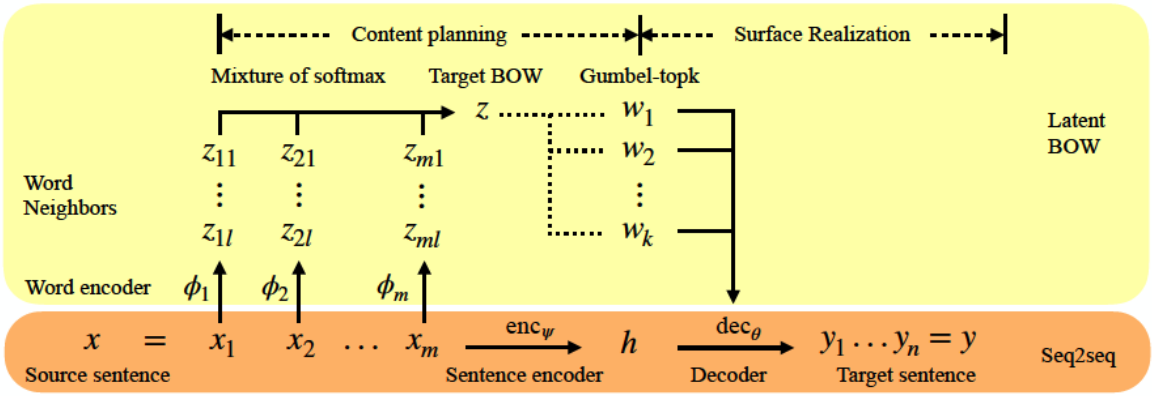

Figure 1: Our model equip the seq2seq model(lower part) with latent bag of words(upper part).

实验

作者采用Quora和MSCOCO这两个数据集进行实验，作者使用一个seq2seq的LSTM，并有残差连接和attention机制的模型作为baseline，作者也使用了一个β-VAE模型作为baseline，通过调控β参数来平衡重建和识别网络。其中LBOW-Topk网络是没有采用Gumbel重参化的方法挑选top k的词，LBOW-Gumbel是使用了Gubmel技巧的网络，Cheating BOW模型是在生成的时候能够看到target sentence中的BOW，可以看做是LBOW模型的上限，实验结果如下：

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

生成一个句子的具体过程如下，主要分为了三个阶段：

- 生成source word的邻近词
- 从邻近词的组合中进行sample
- 利用sample得到的结果进行句子生成。

  具体case过程如下：

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.