

# Long and Diverse Text Generation with Planning-based Hierarchical Variational Model (PHVM)介绍与实验

涂皓钦

清华大学NGNLab(实习)

2020.9.17

# Structure

- **模型介绍**
- **数据集介绍**
- **实验及思考**

# 模型介绍

Input:

- |  |                                  |  |
|--|----------------------------------|--|
| 1. <类型, 裙><br><Category, Dress / Skirt>    | 5. <风格, 青春><br><Style, Youthful> | 9. <裙腰型, 高腰><br><Waist, High-rise>     |
| 2. <版型, 显瘦><br><Design, Figure Flattering> | 6. <风格, 清新><br><Style, Fresh>    | 10. <裙长, 半身裙><br><Length, Skirt>       |
| 3. <材质, 棉><br><Material, Cotton>           | 7. <图案, 格子><br><Pattern, Plaid>  | 11. <裙款式, 不规则><br><Element, irregular> |
| 4. <风格, 文艺><br><Style, Aesthetic>          | 8. <裙下摆, 荷叶边><br><Hem, Flounce>  |  |

Generation:



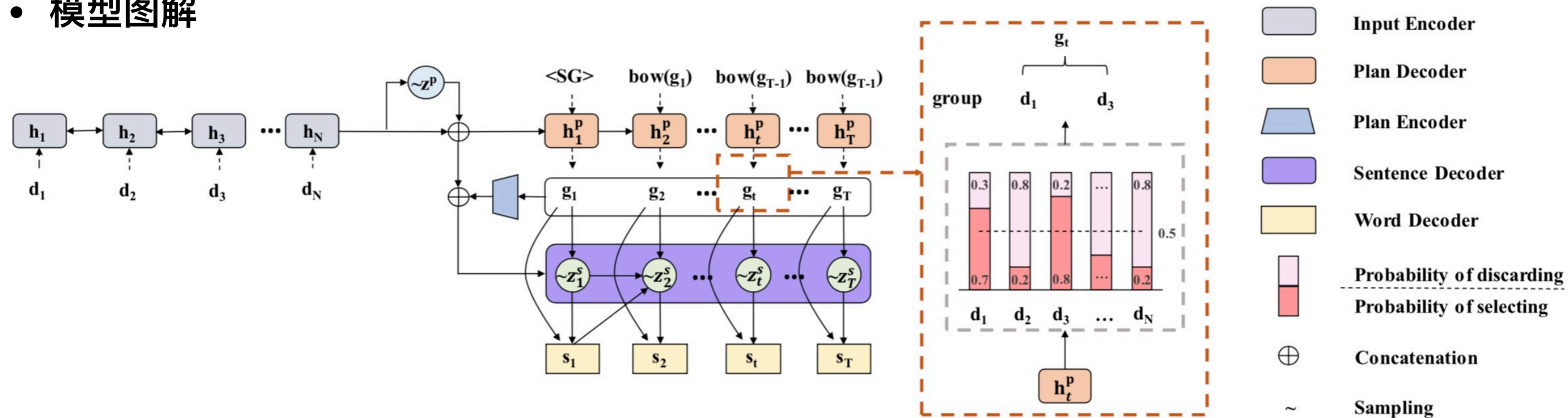
目标: 根据输入不同的描述性二元组, 生成与之相关的广告\菜单等连贯、逻辑通顺的长文本

输入: 给定商品的data, 若干个 $d_i = \langle \text{属性}, \text{值} \rangle$ 组成的计划 $x = \{d_1, d_2, \dots, d_N\}$

输出: 与计划相关的文本 $y = \{s_1, s_2, \dots, s_T\}$

# 模型介绍

## 模型图解

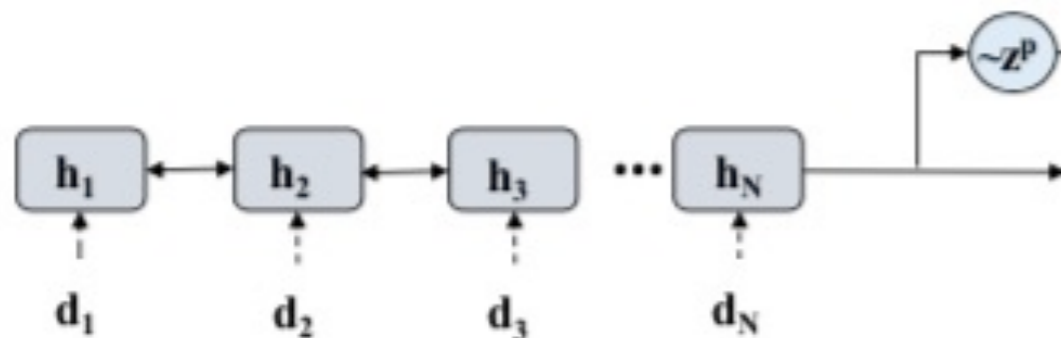


数据在模型中的流动：

1. 通过 *input-encoder* 把 *data* 表征
2. 利用 *plan-decoder* 指定生成文本的计划，再通过 *plan-encoder* 将计划表征
3. 最后分别交给 *sentence-decoder* 去生成某个句子，*word-decoder* 去生成该句子中对应的单词

# 模型介绍

## • 输入编码器(BiGRU)



输入:  $x = \{d_1, d_2, \dots, d_N\}$

输出: 关于这些记录的特征向量  $enc(x)$  以及一个全局计划隐变量  $z^P$ , 即它可以决定怎么组织句子生成文本。

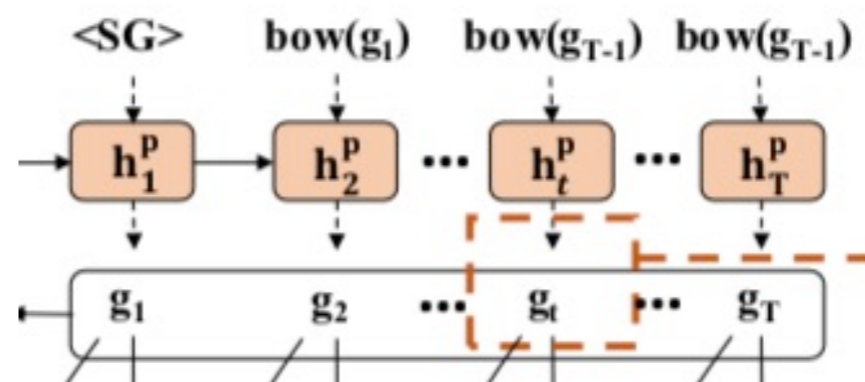
$z^P$  通过采样生成:

- 先验分布  $p_\theta(z^P | x) = \mathcal{N}(\mu^P, \sigma^{P^2} \mathbf{I})$
- 后验分布  $q_{\theta'}(z^P | x, y) = \mathcal{N}(\mu^{P'}, \sigma^{P'^2} \mathbf{I})$

•  $[\mu^P; \log \sigma^{P^2}] = MLP_\theta(x)$  变分推断时用先验

•  $[\mu^{P'}; \log \sigma^{P'^2}] = MLP_{\theta'}(x, y)$  训练时用后验

## • 计划解码器(GRU)



输入: 上面步骤得到的  $enc(x)$  与  $z^P$  与  $d_i$  的特征  $h_i$

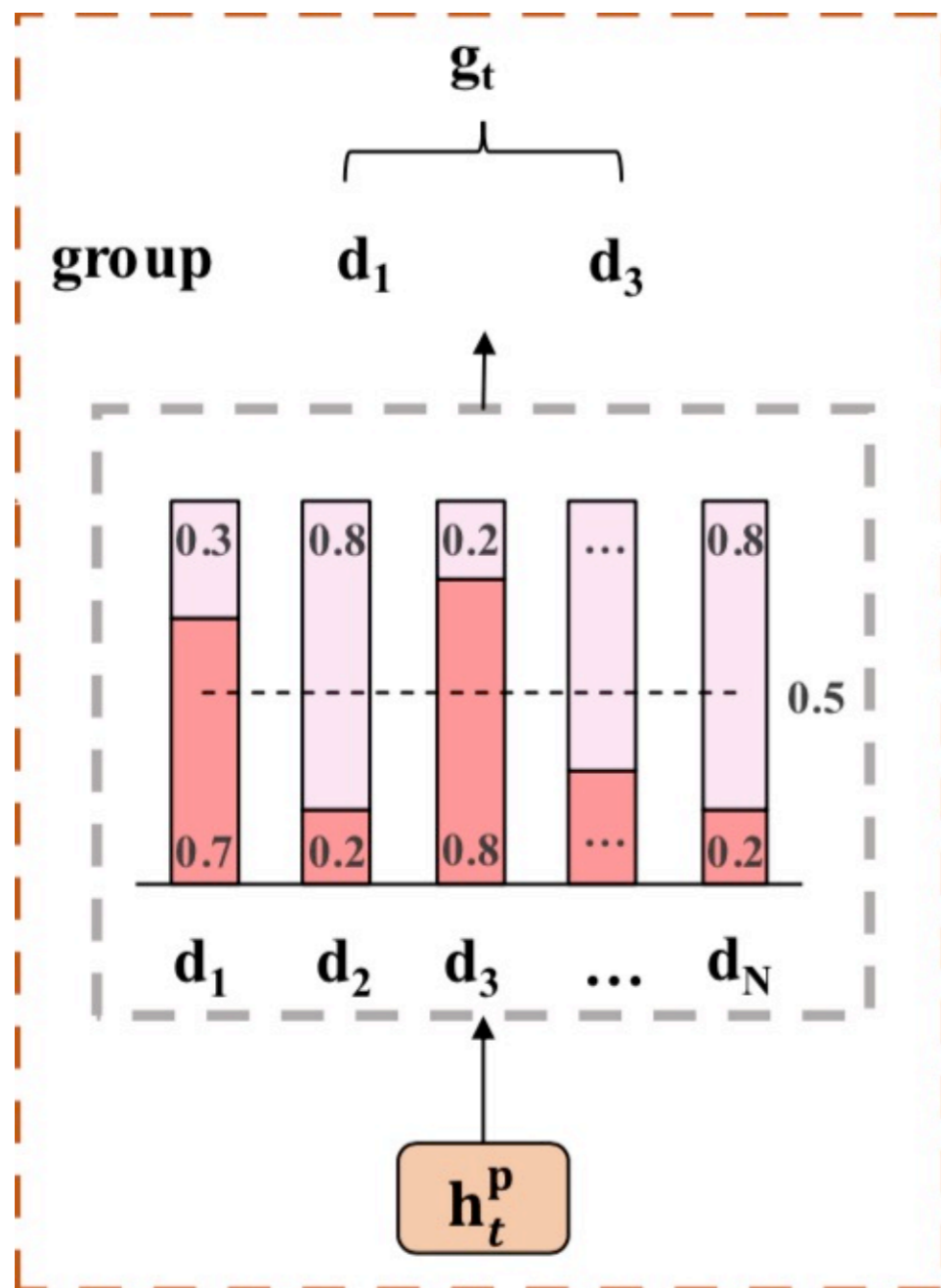
输出: 一组“输出计划”  $g = g_1 g_2 \dots g_T$ ,  $g_i$  表示第  $i$  个句子需要表达的“属性”, 参与第  $i$  个句子的生成。

细节:

- 解码器初始的隐藏状态为  $enc(x)$  和  $z^P$
- 每次  $h_t^P$  的更新的输入, 都是  $g_{t-1}$  特征的平均池化

$$\text{bow}(g_t) = \text{average pool} \left( \{h_i \mid d_i \in g_t\} \right)$$

# 模型介绍



$g_t$ 具体的生成：在时间步  $t$ ，我们是对每个  $d_i$  做二分类，来判断  $d_i$  是否在  $g_t$  中  $P(d_i \in g_t | g_{<t}, x, z^p)$ ，
$$P(d_i \in g_t) = \sigma \left( v_p^T \tanh \left( W_p [h_i; h_t^p] + b_p \right) \right)$$

决定  $d_i$  是否在  $g_t$  中：
$$g_t = \{ d_i \mid P(d_i \in g_t) > 0.5 \}$$

结束解码的标志： $P_t^{\text{stop}} = \sigma(W_c h_t^p + b_c)$ ，默认设置为超过0.5即停止解码，后面会给它引入一个单独的loss

# 模型介绍

## • Hierarchical Generation Process

知道了每句话需要描述的内容，生成长文本建模如下：

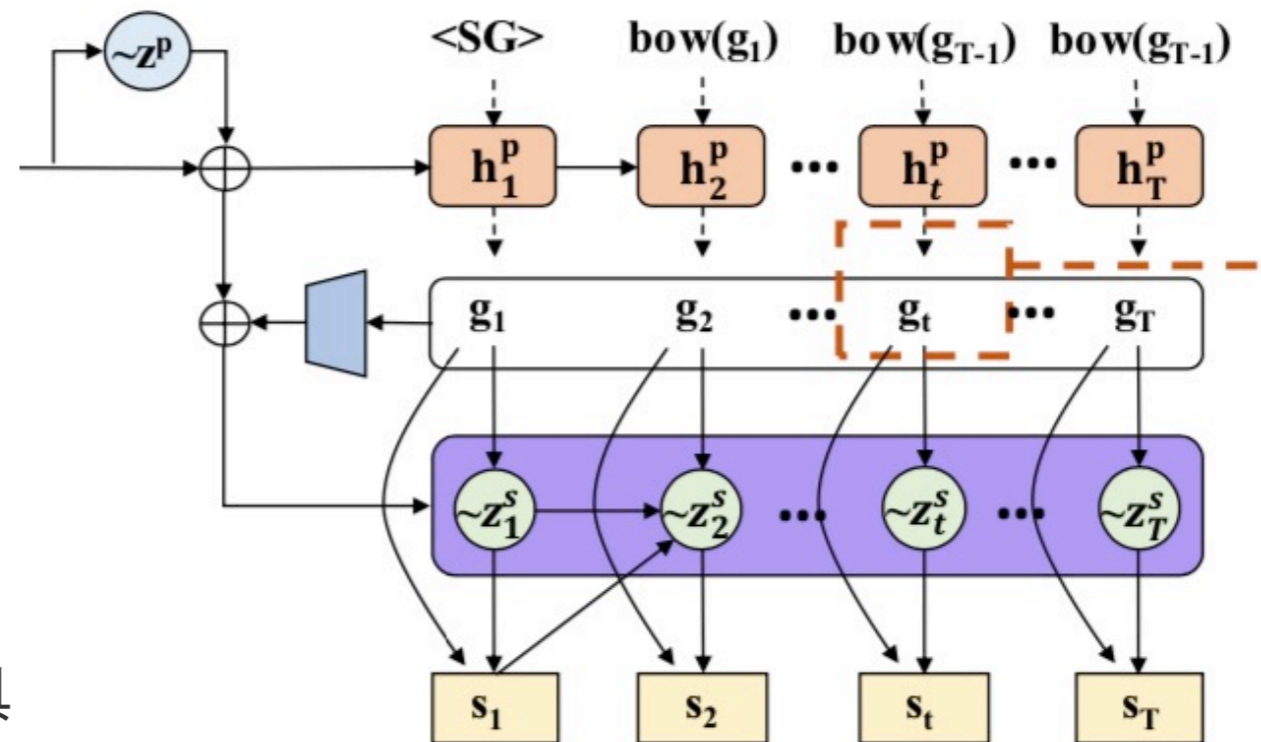
$$c = \{x, z^P\}$$

$$y = \operatorname{argmax}_y P(y | g, c)$$

$$y = s_1 s_2 \dots s_T$$

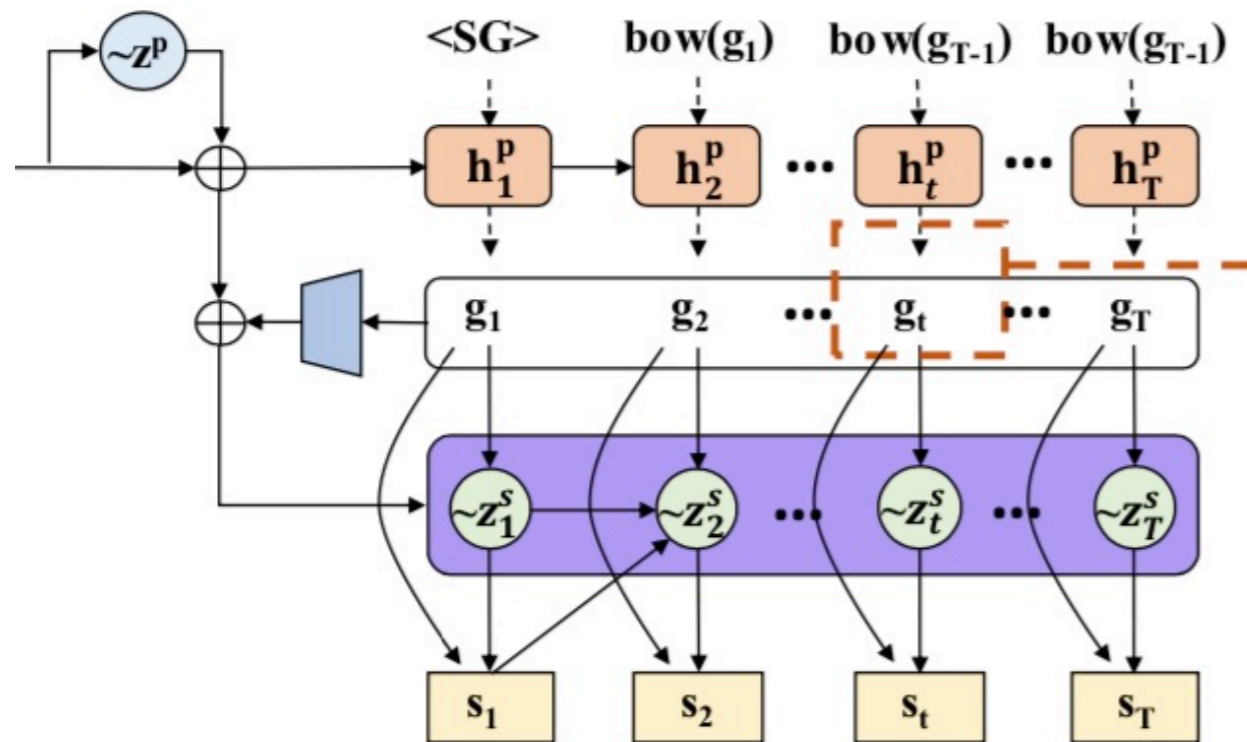
输入：  $x, z^P, g$

输出：一方面是每个位置句子解码都可以得到一个**隐变量**  $z_t^S$  (第二次变分) 指导生成具体的特征，以及最终输出的文本  $y = s_1 s_2 \dots s_T$



紫色部分为句子解码器，通过GRU实现。这一块模型要做的事情是得到每个时间  $t$  下需要采样的**局部隐变量**  $z_t^S$ ，跟着之前提到的特征  $z^P, g_i$  一起具体参与到第  $t$  个句子的生成过程。

# 模型介绍



$z_t^s$ 的采样与训练过程:

$$p_\phi(z_t^s | s_{<t}, g_t, c) = \mathcal{N}(\mu_t^s, \sigma_t^{s^2} \mathbf{I}) \text{ 先验分布}$$
$$q_{\phi'}(z_t^s | s_{\leq t}, g_t, c) = \mathcal{N}(\mu_t^{s'}, \sigma_t^{s'^2} \mathbf{I}) \text{ 后验分布}$$

采取VAE中拟合分布方差和期望的步骤:

$$h_t^s = \mathbf{GRU}_s \left( [z_{t-1}^s; h_{t-1}^w], h_{t-1}^s \right)$$

$$[\mu_t^s; \log \sigma_t^{s^2}] = \mathbf{MLP}_\phi \left( h_t^s, \text{bow}(g_t) \right)$$

$$[\mu_t^{s'}; \log \sigma_t^{s'^2}] = \mathbf{MLP}_{\phi'} \left( h_t^s, \text{bow}(g_t), s_t \right)$$

其中 $h_{t-1}^w$ 为 $t-1$ 步word-decoder的隐藏状态。这样的话，采样得到的隐变量，就可以很好地表达指定属性数据(有 $z_{t-1}^s$ 的参与)，也能保持一定的语义连贯性(有 $h_{t-1}^s, h_{t-1}^w$ 的参与)

## Word-Level Generation

GRU进行生成，在 $\{h_t^s, z_t^s, g_t\}$ 条件下句子 $s_t = \operatorname{argmax}_{s_t} P(s_t | s_{<t}, z_t^s, g, c)$ ,

$$c = \{x, z^p\}; w_k^t \sim P(w_k^t | w_{<k}^t, s_{<t}, z_t^s, g, c)$$



# 模型介绍

- **Loss Function**

loss function分为3个:

1. 整个输出为句子的变分loss(ELBO), 即对应采样全局隐变量 $z^P$ 的ELBO:

$$\log P(y | x) \geq E_{q_{\theta'}(z^P | x, y)} \left[ \log P(y | x, z^P) \right] - D_{KL} \left( q_{\theta'}(z^P | x, y) \| p_{\theta}(z^P | x) \right)$$

$$\log P(y | x, z^P) = \log P(g, y | x, z^P) = \sum_{t=1}^T \log P(g_t | g_{<t}, x, z^P) + \log P(s_t | s_{<t}, g, x, z^P)$$

$$\log P(s_t | s_{<t}, g, x, z^P) \geq E_{q_{\phi'}(z_t^s | s_{\leq t}, g, x, z^P)} [\log P(s_t | s_{<t}, z_t^s, g, x, z^P)] - D_{KL} \left( q_{\phi'}(z_t^s | s_{\leq t}, g, x, z^P) \| p_{\phi}(z_t^s | s_{<t}, g, x, z^P) \right)$$

$$2. \mathcal{L}_2 = \sum_{t=1}^{T-1} \log P_t^{stop} + \log \left( 1 - P_T^{stop} \right)$$

3. 一个**bag-of-word loss**\*用来对付散度崩塌(KL collapse)情况

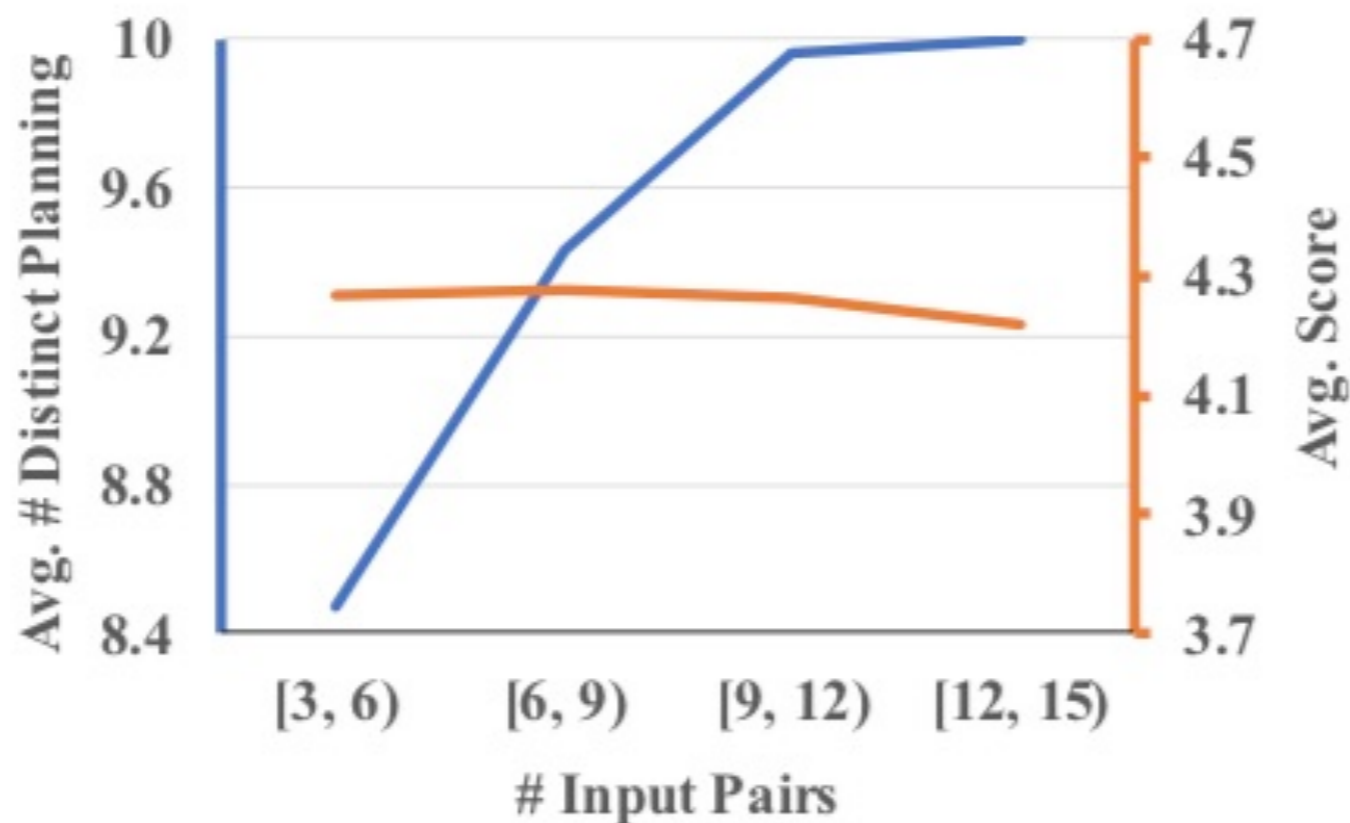
\*(Zhao et al.,2017) Learning discourse-level diversity for neural dialog models using conditional variational auto encoders.

# 模型介绍

- Diversity of planning

同样的数据集通过不停地采样隐变量，可以获得多种多样的表达方式。相比于之前的文本生成，这里的工作保证了他确实能够丰富表达方式。

可以看到，如果输入的数据有10对以上，几乎10次采样的结果，都会有10种不同的表达。而且人工评分基本都保持在4.0/5.0以上

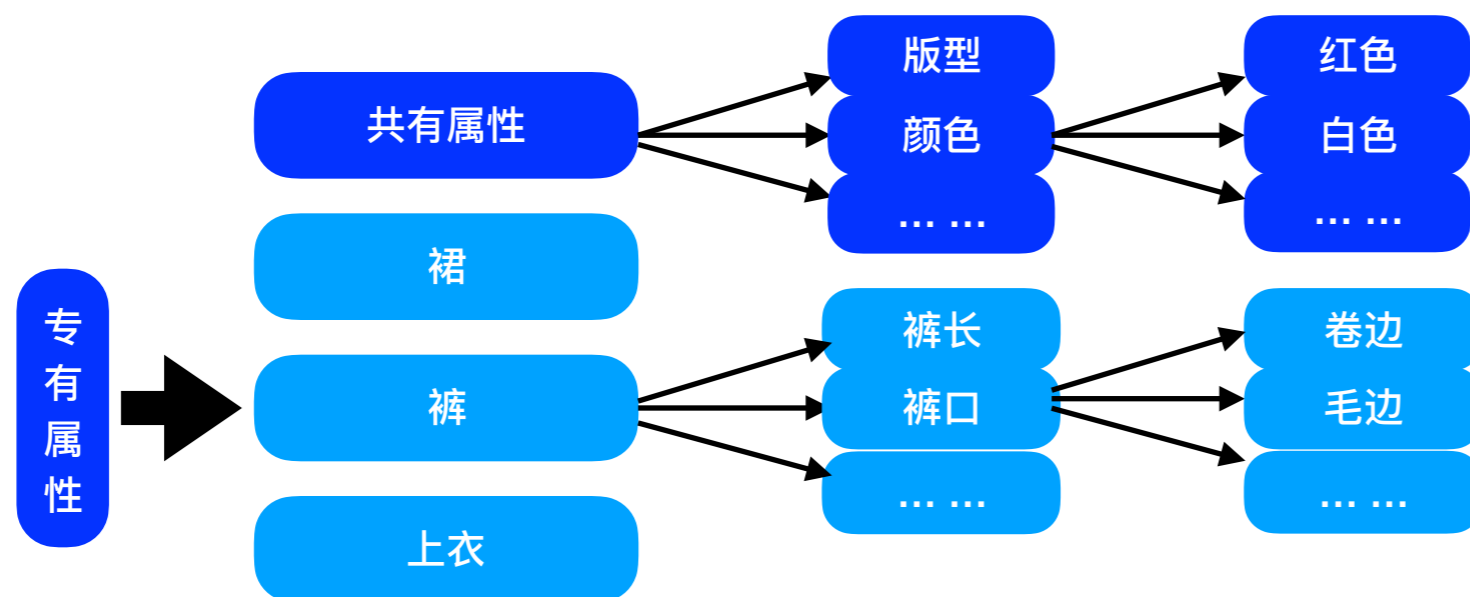


# 数据集介绍

从淘宝平台构造的数据集，共119K条数据，构造如下：

```
{  
  "feature": [ ["类型", "裤"], ["风格", "简约"], ["风格", "潮"], ["图案", "格子"], ["图案", "几何"], ["图案", "线条"], ["裤长", "七分裤"], ["裤型", "阔腿裤"] ],  
  "desc": "这款阔腿裤，整体设计简约利落，时尚的阔腿款式带来鲜明的几何设计美感，褪去传统装束的厚重与臃肿，更具轻盈美感。搭配七分裤长修饰出挺拔的腿部线条，气质的格纹图案不显单调，尽显女性优雅气质。斜门襟设计潮流出众，让你时刻保持动人的女性风采。",  
  "专有属性": [ ["裤长", "七分裤"], ["裤型", "阔腿裤"], ["类型", "裤"] ],  
  "共有属性": [ ["风格", "简约"], ["风格", "潮"], ["图案", "格子"], ["图案", "几何"], ["图案", "线条"] ],  
  "segment": {  
    "seg_0": { "segId": 0, "key_type": ["裤型", "图案", "风格"], "order": [ ["风格", "简约"], ["裤型", "阔腿裤"], ["图案", "几何"], "seg": "这款阔腿裤，整体设计简约利落，时尚的阔腿款式带来鲜明的几何设计美感，褪去传统装束的厚重与臃肿，更具轻盈美感。"},  
    "seg_1": { "segId": 1, "key_type": ["图案", "裤长"], "order": [ ["裤长", "七分裤"], ["图案", "线条"], ["图案", "格子"], "seg": "搭配七分裤长修饰出挺拔的腿部线条，气质的格纹图案不显单调，尽显女性优雅气质。"},  
    "seg_2": { "segId": 2, "key_type": ["风格"], "order": [ ["风格", "潮"], "seg": "斜门襟设计潮流出众，让你时刻保持动人的女性风采。"} } } ]
```

对于每个feature二元组的构成，可以抽象为以下格式：**二元组是由<二级属性，三级属性>组成**



# 数据集介绍

版型	材质/面料	颜色	风格	裙型/裙身	裙下摆/裙摆/下摆	裙腰型	裙长
宽松	棉/纯棉/棉料/全棉/棉	白色	原创设计	蛋糕/蛋糕裙	开叉/开衩	自然腰	中长裙
显瘦/修身	亚麻	黑色	街头	百褶/百褶裙	荷叶边/荷叶下摆	低腰	短裙
立体剪裁/立体裁	蚕丝/桑蚕丝/真丝	红色	通勤	西装裙	层叠/叠层/层层叠叠	高腰/超高腰	中裙
H/H廓形	天丝	紫色	韩版	拼接裙	弧形/圆弧	中腰	长裙
	牛仔布/牛仔/牛仔褲/牛	纯色	淑女	A字/A字裙/a字/a字裙	毛边	松紧腰	超短裙
	绸缎	蓝色	欧美	背带裙/背带	花边	超低腰	不过膝
	丝绒	粉色	瑞丽	牛仔裙	压褶	宽松腰	半身裙/半裙
	棉麻	灰色	日系	花苞裙/花苞	垂坠/垂坠感/垂顺		连衣裙
	羊毛	黑白	英伦	蓬蓬裙/蓬蓬			
	羊绒	金色	复古	小黑裙/小黑			
	莱卡	银色	文艺	伞裙			
	条绒/灯芯绒	黄色	宫廷	铅笔裙/铅笔			
	针织	绿色	简约	仙女裙			
	网纱/纱网	浅蓝色/浅蓝	乡村	抹胸裙/抹胸			
	蕾丝	淡蓝色/淡蓝	民族风	公主裙			
	水洗	深蓝色/深蓝	OL	灯笼裙			
	雪纺/厚雪纺	迷彩色/迷彩	知性	包臀裙/包臀			
	混纺	酒红色/酒红	居家	一步裙/一步			
	西装面料	玫瑰金色/玫瑰金	运动	大裙摆/大摆/大摆裙/大的裙摆			
	涤纶	卡其色/卡其	休闲	鱼尾裙/鱼尾			
				直筒裙/直筒			
				衬衫裙			
				网纱裙			

其中共有属性共5个二级属性：  
版型、材质/面料、颜色、风格、  
图案/印花。

# 实验及思考

- 固定指导每句话生成的**二元组个数**——选取较高概率

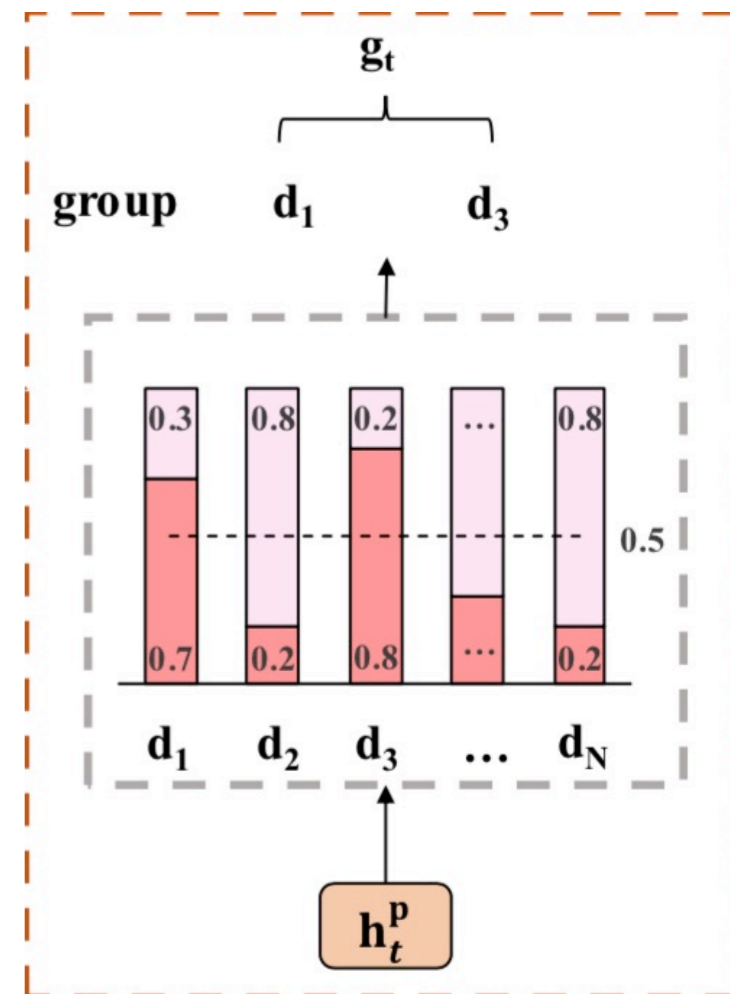
每句话选取概率最高的1个二元组指导生成

- [[['图案', '格子'], ['图案', '线条'], ['风格', '潮'], ['裤型', '阔腿裤']], [{"desc": "这款 **格子** 西裤，采用 维希 **格纹** 设计，时尚与经典并存。一体式设计，线条流畅。时尚 **潮流**，彰显与众不同的时尚品味。**阔腿** 的版型，时尚有型。"}]

每句话选取概率最高的2个二元组指导生成

- [[['版型', '宽松'], ['衣样式', '针织衫'], ['图案', '复古'], ['风格', '复古'], ['衣样式', '针织衫'], ['衣样式', '针织衫'], ['图案', '复古']], [{"desc": "**针织衫** 采用 **宽松** 的衣身版型，将女性的身形衬托得更加立体饱满，结合着浪漫唯美的花朵元素，带来强烈的视觉冲击，衬的人身材更加高挑有型，将女性的柔美气质衬托的淋漓尽致。**复古** 的款式，整体造型自然和谐，彰显十足的活力。"}]

- [[['图案', '复古'], ['风格', '复古'], ['风格', '知性'], ['图案', '花色'], ['图案', '线条'], ['裤型', '阔腿裤'], ['图案', '印花'], ['颜色', '绿色']], [{"desc": "设计师采用了 **复古** 的苧麻面料，其结合的主色调和 **复古** 的配色，十分的大气，带来舒适的穿着体验。花色的花色运用在裤子的设计里，采用了 **知性** 干练的 **花色**，尽显女性的 **知性** 与温柔。"}]



# 实验及思考

- **手动输入二元组**，生成与之相关的文字
- 发现数据中一定会出现一个固定的属性二元组：<类型，… …>，例如<类型，上衣>、<类型，裙>。并且在指导生成句子的时候，类型对应二元组**并不会明显地出现在某个句子中**，且不会直接输入模型指导任何句子的生成。因此在指定二元组时遵循这一规则，即第一个二元组必定是<类型，… …>格式（认为无实际含义）
- 二元组按照属性划分，分为“共有属性”以及“专有属性”因此在指定二元组的时候，抽取了一个、两个、三个二元组分别进行句子生成，设定**1个二元组指导每个句子生成**

二元组个数	分布情况
1	1个共有属性 / 1个专有属性
2	1个共有属性 + 1个专有属性
3	1个共有+2个专有 / 2个共有+1个专有
4	2个共有属性 + 2个专有属性
6	3个共有属性 + 3个专有属性
8	4个共有属性 + 4个专有属性

# 实验及思考

生成句子：

- 1个二元组（共有属性）：

[[ "类型", "裤"], [ "图案", "线条" ] ] — — — — [ [ '图案', '线条' ], [], [], [] ]

裤子版型设计，对身材没有任何的要求，方便穿脱。后部UNK插兜，**线条**流畅，独具特色。

- 1个二元组（专有属性）：

[[ "类型", "裤"], [ "裤型", "阔腿裤" ] ] — — — — [ [ '裤型', '阔腿裤' ], [], [], [] ]

时尚的**阔腿裤**，对腿型的修饰效果非常好，展现出与众不同的美。红祥带的腰带设计，可以张扬出个人魅力。前短后长的侧缝插袋，既能很好的打破单调性，也使其可能不可能。

- 2个二元组（1专有+1共有）：

[[ "类型", "裙"], [ "裙长", "连衣裙"], [ "颜色", "深色" ] ] — — — — [ [ '颜色', '深色' ], [ [ '裙长', '连衣裙' ], [], [], [] ] ]

假两件的**连衣裙**，是一件利用率极高的单品，无论是在正式场合还是在搭配上，都需要一件精致的连衣裙，给人一种假两件的感觉，充满了浪漫的气息，将少女感表现的淋漓尽致，简单大气的纯色调，搭配上**深色**系，很有时尚感。

- 3个二元组（1专有+2共有）：

[[ "类型", "裤"], [ "裤型", "阔腿裤"], [ "图案", "几何"], [ "图案", "线条" ] ] — — [ [ '裤型', '阔腿裤' ], [ [ '图案', '几何' ], [ [ '图案', '线条' ] ] ]

**几何**纹理足以修饰身形，显脸小的视觉效果，尽显女性魅力。百褶设计，增加裙身的垂坠感，令双腿**线条**更显修长。**阔腿裤**修饰腿型，穿上显得双腿纤细修长。

# 实验及思考

- 给定生成句子，对指导其生成的二元组进行抽取

思考：

- 考虑决策树分类与textCNN分类。难点：**类别多**（考虑<属性，值>可分为400+类，仅考虑“值”可分为322类）、**句子较短**，一个句子中可能不只有一个二元组，抽取语意可能不准确、句子中二元组能够**明显地找出**，因此用正则匹配进行查找。
- 二元组<属性，值>中的“值”并不是某一个属性特有的，而句子中匹配出来的字眼又往往是“值”因此在正则匹配遇到这样的情况（<a, b>与<c, b>）难以准确地抽取整个二元组。目前只考虑抽取“值”进行计算。
- 从上面的例子看出来，生成的句子并不能把所给的所有二元组包含进去，但是也不会一个都不包含进去

考虑到正则匹配可以将句子中所有的二元组准确地抽取出来，以此衡量模型生成的句子



# 实验及思考

Recall: 理解为模型生成句子能够全部包含输入二元组中“值”的概率

输入个数 指导个数	1	2
1(专有 共有)	93.54% 97.60%	X
2	79.85%	90.18%
3	40.95%	80.46%

Precision: 理解为模型生成句子中二元组与输入二元组中“值”完全相同的概率

输入个数 指导个数	1	2
1(专有 共有)	47.28% 69.05%	68.51% 73.10%
2	58.14%	65.10%
3	28.02%	56.75%

# 实验及思考

- 思考与后续工作

1. 当指定小于6个二元组输入时，某个生成的句子大都没有与具体指导其生成的二元组完全对应。但是生成的句子还是会包含这些二元组的值。这时考虑全局隐变量 $z^P$ 对其指导作用更大。
2. 对于模型生成句子中包含二元组准确率不高的情况，可以考虑将没有包含进去的二元组再次放入模型中进行多次生成，直到生成的句子将绝大多数手动输入的二元组包含进去
3. 观察发现，生成句子比较容易包含特定的二元组，而包含另外的一些二元组则相对较困难。因此后续工作考虑优化数据集，不仅剔除难以包含到句子中的二元组。并且固定<属性， 值>二元组，使每一个“值”对应唯一的属性，这样可以通过“值”抽取出完整的二元组。
4. 对于指定输入二元组，需要做实验检验生成文本质量，利用分类器将其与标签文本进行分类查看其生成句子效果如何。

# Thank you!

涂皓钦

清华大学NGNLab

thq415\_ic@yeah.net

2020.9.17