

SongNet：硬格式控制下的文本生成

SongNet 模型是腾讯 AI Lab NLP 中心自研的一套基于预训练语言模型的硬格式控制下的文本生成模型。它能接受任意指定的格式进行文本生成，例如生成歌词、唐诗宋词、莎士比亚十四行诗等。以下为论文的详细解读。

Rigid Formats Controlled Text Generation

1 背景

歌词、诗词等是比较独特的文体形式，有严格的格式、韵律等要求。平时我们见到的文本生成任务如对话生成、故事生成、摘要生成等多以自由生成方式为主，给定固定模板严格限制格式进行填词的文本生成任务研究较少。SongNet 模型旨在解决此类硬格式控制的文本生成问题。例如，给定一首歌的歌词，让模型能够将歌词全部改写，或者只改写部分歌词。在改写的同时保证文本的格式正确、韵律合理、句子完整等基本的质量需求。具体的，给定《十年》这首歌词的格式，我们可以通过 SongNet 重新进行配词，并保证格式不变，可以根据原来的曲谱进行演唱：

原歌词：十年之前/我不认识你/你不属于我/我们还是一样/陪在一个陌生人左右/走过渐渐熟悉的街头

新配词：夜深人静/思念你模样/多少次孤单/想伴在你身旁/是什么让我如此幻想/为何会对你那般痴狂

2 方法

2.1 问题定义

输入：一个模板 $C = \{c_1 c_2 c_3, c_4 c_5 c_6 c_7 \cdot\}$

输出：将 c_* 替换成真实的文字，生成符合模板格式的完整句子。

2.2 模型框架

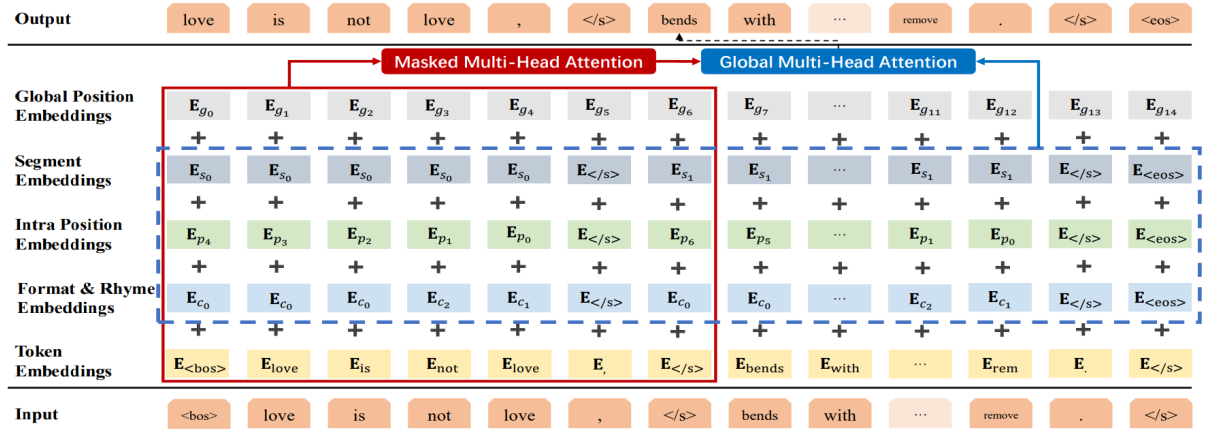


图 1 模型框架

基于 Transformer 的自回归语言模型，我们提出了自己的框架。为了解决模板、韵律、句子完整性问题，我们定义了几种符号来对这些特殊的信息进行建模。

模板和韵律符号：

$$C = \{c_0, c_0, c_0, c_2, c_1, \langle /s \rangle\}$$

$$c_0, c_0, c_0, c_0, c_0, c_2, c_1, \langle /s \rangle, \langle eos \rangle\}$$

其中 c_0 代表普通文本符号， c_2 代表该位置是韵脚， c_1 代表标点符号。如此设计的目的是让模型在读模板的时候，碰到 c_2 的时候能推测出该处是结尾词、韵脚词，遇到 c_1 的时候能够知道该输出一个标点符号（也可以保留模板中的标点符号不做改写）。

内部位置符号：

$$P = \{p_4, p_3, p_2, p_1, p_0, \langle /s \rangle\}$$

$$p_6, p_5, p_4, p_3, p_2, p_1, p_0, \langle /s \rangle, \langle eos \rangle\}$$

模板中的每个句子都会有一个内部位置序列，特别强调的一点是，我们这里将符号倒序排列，旨在让模型能够学习到句子的一个渐进结束的过程，当遇到 p_0 的时候句子结束，那么当遇到 p_1, p_2 的时候模型能够意识到句子生成动态到了尾声。除此之外，句子内部位置信息还能捕捉诸如对联一类文本的前后句对仗位置特征。

句子符号：

$$S = \{s_0, s_0, s_0, s_0, s_0, \langle /s \rangle\}$$

$$s_1, s_1, s_1, s_1, s_1, s_1, s_1, s_1, \langle /s \rangle, \langle eos \rangle\}$$

对句子位置进行建模，每个符号代表句子在序列中的 ID，旨在让模型推测出目前在生成的句子位置，捕捉学习到押韵的句对。例如莎士比亚的十四行诗押韵格式为“ABAB

CDCD EFEF GG”，有了句子位置特征模型就会比较容易捕捉到 1 句和 3 句押韵，2 和 4 押韵等。

训练的时候，所有的符号以及文本转化成向量后输入到基于 Transformer 的语言模型中：

$$\mathbf{H}_t^0 = \mathbf{E}_{w_t} + \mathbf{E}_{c_t} + \mathbf{E}_{p_t} + \mathbf{E}_{s_t} + \mathbf{E}_{g_t}$$

为了让模型在 auto-regressive 的这个动态过程中能够看到模板的未来信息，我们还引入了另一个变量，只融合了之前定义的符号信息，没有输入的文本信息避免了右侧信息泄露：

$$\mathbf{F}_t^0 = \mathbf{E}_{c_t} + \mathbf{E}_{p_t} + \mathbf{E}_{s_t}$$

然后通过两个 Attention 层对输入的信息进行建模，第一层是 auto-regressive 建模中的带 masking 的 Attention 建模机制：

$$\begin{aligned}\mathbf{C}_t^1 &= \text{LN}(\text{FFN}(\mathbf{C}_t^1) + \mathbf{C}_t^1) \\ \mathbf{C}_t^1 &= \text{LN}(\text{SLF-ATT}(\mathbf{Q}_t^0, \mathbf{K}_{\leq t}^0, \mathbf{V}_{\leq t}^0) + \mathbf{H}_t^0) \\ \mathbf{Q}^0 &= \mathbf{H}^0 \mathbf{W}^Q \\ \mathbf{K}^0, \mathbf{V}^0 &= \mathbf{H}^0 \mathbf{W}^K, \mathbf{H}^0 \mathbf{W}^V\end{aligned}$$

第二层是能够看到模板全局信息的 Attention 层：

$$\begin{aligned}\mathbf{H}_t^1 &= \text{LN}(\text{FFN}(\mathbf{H}_t^1) + \mathbf{H}_t^1) \\ \mathbf{H}_t^1 &= \text{LN}(\text{GLOBAL-ATT}(\mathbf{Q}_t^1, \mathbf{K}^1, \mathbf{V}^1) + \mathbf{C}_t^1) \\ \mathbf{Q}^1 &= \mathbf{C}^1 \mathbf{W}^Q \\ \mathbf{K}^1, \mathbf{V}^1 &= \mathbf{F}^0 \mathbf{W}^K, \mathbf{F}^0 \mathbf{W}^V\end{aligned}$$

采用最大似然估计作为优化方法：

$$\mathcal{L}^{\text{nll}} = - \sum_{t=1}^n \log P(\mathbf{y}_t | \mathbf{y}_{<t})$$

2.2 训练中的 masking 策略

在生成任务中，会遇到给定部分文本，补充其他文本的局部补全需求。例如，在模板中，给定部分句子，补全其他句子；在句子中，知道了部分词，补充其他词。为了使模型有解决该类需求的能力，我们在预训练阶段，设计了 masking 策略：模板中保留部分词，模型来学习预测其他的词。

$$C' = \{c_0, c_0, c_0, love, c_1, \langle /s \rangle \\ bends, c_0, c_0, c_0, c_0, remove, c_1, \langle /s \rangle, \langle eos \rangle\}$$

这样模型就可以根据两侧的 context 信息进行预测生成，近似的有了双向语言模型的能力。

3 Inference 方法

我们采用了 top-k sampling 的解码方法来生成多样性好的结果，通过调节超参 k 可以平衡多样性和质量。

4 试验

4.1 数据集

中文我们以宋词生成作为预测目标；英文生成莎士比亚十四行诗(sonnet)。

模型分预训练和微调两个阶段。在预训练阶段，中文语料主要是新闻和维基百科，英文语料是维基百科和 BooksCorpus。

4.2 评测指标

除了常用的文本生成评测指标如 ppl、distinct 外，我们还专门设计了度量模板准确率（Format）、韵律准确率（Rhyme）和句子完整度（integrity）的指标。

4.3 结果

Model	PPL↓		Diversity (Distinct) ↑			
	VAL	TEST	MA-D-1	MI-D-1	MA-D-2	MI-D-2
S2S	19.61	20.43	75.35	2.48	98.35	36.23
GPT2	148.11	104.99	-	-	-	-
GPT2 w/ Fine-tuning	18.25	17.00	73.87	2.57	96.07	33.92
SongNet (only Pre-training)	24.41	16.23	74.84	4.59	95.09	54.98
SongNet (only Fine-tuning)	12.75	14.73	75.96	2.69	97.59	37.26
SongNet	11.56	12.64	75.04	2.66	97.29	36.78

Model	Format↑		Rhyme↑		Integrity↓
	MA-F1	MI-F1	MA-F1	MI-F1	
S2S	44.32	38.16	53.80	52.27	8.30±2.06
GPT2 w/ Fine-tuning	35.70	35.20	53.48	52.50	45.92±20.12
SongNet (only Pre-training)	29.12	29.46	53.77	53.13	30.98±14.06
SongNet (only Fine-tuning)	99.81	99.83	79.23	78.63	2.14±0.10
SongNet	99.88	99.89	73.21	72.59	1.77±0.16

表 1：宋词生成效果评测

Model	PPL↓		Diversity (Distinct) ↑			
	VAL	TEST	MA-D-1	MI-D-1	MA-D-2	MI-D-2
GPT2 w/ Fine-tuning	31.47	31.03	73.87	2.57	96.07	33.92
SongNet (only Pre-training)	28.56	28.07	49.92	25.14	85.35	65.70
SongNet (only Fine-tuning)	34.62	34.53	42.31	4.96	90.76	47.26
SongNet	27.46	27.63	43.01	10.43	80.06	56.14

Model	Format↑		Rhyme↑		Integrity↓
	MA-F1	MI-F1	MA-F1	MI-F1	
GPT2 w/ Fine-tuning	2.03	1.91	5.20	6.24	15.77±3.63
SongNet (only Pre-training)	99.99	99.99	3.93	4.01	15.28±2.04
SongNet (only Fine-tuning)	99.25	99.99	7.50	7.41	18.86±2.59
SongNet	98.73	98.73	11.46	11.41	11.86±3.01

表 2：莎士比亚十四行诗效果评测

表 1 和表 2 分别是在宋词和十四行诗上的格式生成评测结果，可以看出我们的模型 SongNet 在 ppl、多样性、模板准确率、押韵准确率、句子完整性上都取得了不错的指标。相比宋词，十四行诗的效果差一些，主要原因是十四行诗的训练样本太少（只有 100 首）。

5 例子

（1）指定模板生成

CiPai: Zhe Gu Tian, **Format:** 7. 7. 7. 7. 3. 3. 7. 7. 7.
南枝欲破春寒_浅(qian)。东风一霎春犹_暖(nuan)。海棠零乱暗香_残(can)，露颗未消红泪_满(man)。春不_管(guan)，人何_远(yuan)。江头旧日清溪_伴(ban)。欲凭楼上数行书，却为梦魂无觅_伴(ban)。

CiPai: Bu Suan Zi, **Format:** 5. 5. 7. 5. 5. 5. 7. 5.
新月挂帘旌，暗草蛩凄_楚(chu)。夜半银潢十二阑，画幕灯花_吐(tu)。何处笛声残，晓窗风_雨(yu)。独许寒梅伴小鬟，香在秋千_柱(zhu)。

CiPai: Self-Defined, **Format:** 3. 3. 5. 3. 3. 5. 7. 7.
翠岩中，流水外，别有小壶_天(tian)。自霜清，独斗芳，谁敢并飞_仙(xian)。夜久凉生一曲愁，月如花影似花_圆(yuan)。

CiPai: Self-Defined, **Format:** 9. 9. 9. 9.
雨洒黄泉粼粼细作_寒(han)。越水西来时节自清_闲(xian)。歌笑一杯长唱醉醒_间(jian)。春去不堪寻梦绕舍_山(shan)。

例子中模型不仅可以根据已有的格式模板（例如鹧鸪天、卜算子）进行生成，还能根据我们自己定义的格式进行宋词生成，而且押韵效果也不错。

（2）预先填入词的生成

CiPai: Bu Suan Zi, **Format:** 5, 5. 7, 5. 5, 5. 7, 5.

Format C: _____, _____ 到。_____, _____ 俏。_____, _____ 报。_____, _____ 笑。
(1)风暖莺声苦，又是年时到。长爱柳色三分里，多少为春俏。共少年时节，惯得花时报。争似酴醾入彀来，步步随春笑。
(2)春来春又去，花影几波到。门外绿杨柳萧洒，溪桥人人俏。门外东风马，似旧时香报。但怪桃李晚时羞，冷落无人笑。

Format C: 雨_____, _____ 到。_____, _____ 冰, _____ 俏。_____, _____ 春, _____ 春 报。_____, _____ 山花_____, _____ 笑。
(1)春雨碧连营，柳暗门巷到。宫殿步辇踏寒冰，百媚生春俏。正日烘晴春，花雾春风报。一点山花醉眼明，半倚阑干笑。
(2)雪雨弄轻暖，新涨绿波到。楼上对面湿翠冰，一夜风花俏。晓日烘长春，睡起春宵报。枕畔山花落叶声，不是梦魂笑。

得益于 masking 的训练策略，我们的模型可以进行生成结果的精修，例如生成了一些结果之后，我们可以删掉不满意的部分，让模型重新进行局部补全生成，上面的结果可以看出，效果还是比较令人满意的。

(3) 歌词生成

正文开始我们提到可以通过 SongNet 进行歌曲配词，我们也试验了将《桥边姑娘》这首歌进行重新配词，模型可以轻松的产生成千上万首新的歌词，选择部分结果如下：

夜风扬/残月如霜/又见她在梳妆/一缕柔情凝结成冰凉/谁在空中遥想/她的影子浮现眼眶/孤独飘忽迷茫/怎不愿把这份思量/悄悄藏在心房/天涯两殇/你可曾相望/花落情殇/难道是梦乡/爱过恨过之往/秋水悠长/我却为你痴狂/岁月已成沧桑/寂寞守望/繁华盛放/只有对酒品尝/伴着忧伤流淌/夜风扬/好凄凉/醉看霓虹绽放/谁能看透这几度惆怅/苦涩与绝唱/都变得愈演愈勉强/痛彻愁肠帐/让时光停止流浪/当作路程远航/江山未央/回忆旧模样/地老天荒/何处话凄凉/值得我去吟唱/日出朝阳/映照整个过往/陪看红尘万丈/无力遗忘/寂寞依墙/等到白发苍苍/唯剩满眼悲怆