

融合目标端句法的AMR-to-Text生成

朱杰, 李军辉

苏州大学, 计算机科学与技术学院/ 江苏省苏州市
zhujie951121@gmail.com, lijunhui@suda.edu.cn

摘要

抽象语义表示到文本 (AMR-to-Text) 生成的任务是给定AMR图, 生成相同语义表示的文本。可以把此任务当作一个从源端AMR图到目标端句子的机器翻译任务。目前存在的一些方法都在探索如何更好的对图结构进行建模。然而, 它们都存在一个未限定的问题, 因为在生成阶段许多句法的决策并不受语义图的约束, 从而忽略了句子内部潜藏的句法信息。为了明确考虑这一不足, 该文提出一种直接而有效的方法, 显式的在AMR-to-Text生成的任务中融入句法信息, 并在Transformer和目前该任务最优性能的模型上进行了实验。实验结果表明, 在现存的两份标准英文数据集LDC2018E86和LDC2017T10上, 都取得了显著的提升, 达到了新的最高性能。

关键词: AMR-to-Text生成; 句法决策; 语义约束; 融入句法信息

AMR-to-Text Generation with Target Syntax

Jie Zhu, Junhui Li

School of Computer Science and Technology, Soochow University / Suzhou, Jiangsu
zhujie951121@gmail.com, lijunhui@suda.edu.cn

Abstract

The task of AMR-to-text generation is to generate text with the same semantic representation given an AMR graph. This task can be viewed as a translation task from the source AMR graph to the target sentence. Some existing methods are currently exploring how to better model the graph structure. However, they all have an unrestricted problem, because many syntactic decisions in the generation phase are not constrained by the semantic graph, thus ignoring the syntactic information hidden within the sentence. In order to clearly consider this shortcoming, this paper proposes a direct and effective method, which shows the integration of syntactic information in the task generated by AMR-to-Text, and has conducted experiments on Transformer and the current model of the optimal performance of the task. The experimental results show that on the two existing standard English data sets LDC2018E86 and LDC2017T10, both have achieved significant improvements and reached new state-of-the-art.

Keywords: AMR-to-text generation, Syntactic decision, Semantic constraints, Incorporate syntactic information

1 引言

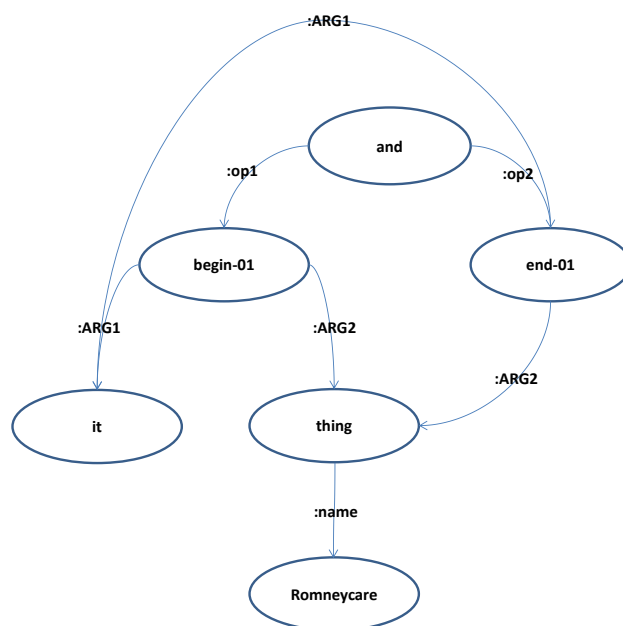


图 1. “It begins and ends with Romneycare .” 抽象成AMR图的一个例子

抽象语义表示 (Abstract Meaning Representation, 简称AMR) (Banarescu et al., 2013) 是一种新型的语义表示方法, 它是从文本中抽象出来捕捉核心的“谁对谁做了什么”的语义结构, 形式上是一种单根有向无环图的结构。图 1 给出了一个AMR图示例, 它是由句子 “It begins and ends with Romneycare .” 抽象而成的。文本中的实词被抽象成AMR图中的概念节点 (concept), 如图中 “begin-01” 和 “thing” 等节点称作为概念。概念之间的相互关系则被抽象为边 (edge), 表示两个概念之间存在的语义关系, 比如 “:ARG0” 和 “:op1” 等。AMR图在语义表示中已经得到了广泛的应用, 并且在机器翻译 (Tamchyna et al., 2015), 问答系统 (Mittra and Baral, 2016), 事件抽取 (Li et al., 2015) 等自然语言处理相关任务也得到了实践。与此同时, AMR-to-Text生成在近年来也受到了越来越多的关注。

AMR-to-Text生成是在给定AMR图条件下, 自动生成相同语义的文本。该任务现存的一些方法 (Flanigan et al., 2016; Konstas et al., 2017; Song et al., 2016; Song et al., 2018; Beck et al., 2018; Damonte and Cohen, 2019; Zhu et al., 2019) 都着重在考虑如何对图关系进行建模, 从而忽略了生成时存在的句法约束。

最初的工作是采用基于统计的方法 (Pourdamghani et al., 2016; Song et al., 2017; Flanigan et al., 2016), 随后 Konstas (2017) 将该任务引入到了序列到序列 (sequence-to-sequence, 简称S2S) 模型上, 使用双向长短时记忆网络 (Bi-LSTM) 进行编码。但是S2S模型需要将AMR图进行序列化去适应模型的输入, 这样会损失大量的图结构信息。因此, 为了更好的对图关系进行建模, Beck等 (2018), Song等 (2018), Damonte等 (2019), Guo等 (2019), Zhu等 (2019) 提出了图到序列 (graph-to-seq, 简称G2S) 的框架, 使用图模型来对AMR图进行建模。然而, 他们的工作都将句子表示为单词序列, 并没有考虑到句子中潜在的句法信息。最近的一些研究也表明, 即使百万级的平行语料, 模型仍然无法从中捕获深层的句法信息 (Li et al., 2017)。

针对上述存在的问题, 本文提出一种显示的方法来融入句法信息, 从而给定生成时一些句法约束, 并且不需要对模型本身进行任何修改。为了更好的证明本文方法的有效性, 本文选取了S2S中最优的Transformer模型和G2S中现存最优的模型 (Zhu et al., 2019) 进行了实验。最终, 在两份标准的英文数据集LDC2015E86和LDC2017T10上都取得了显著的提升。

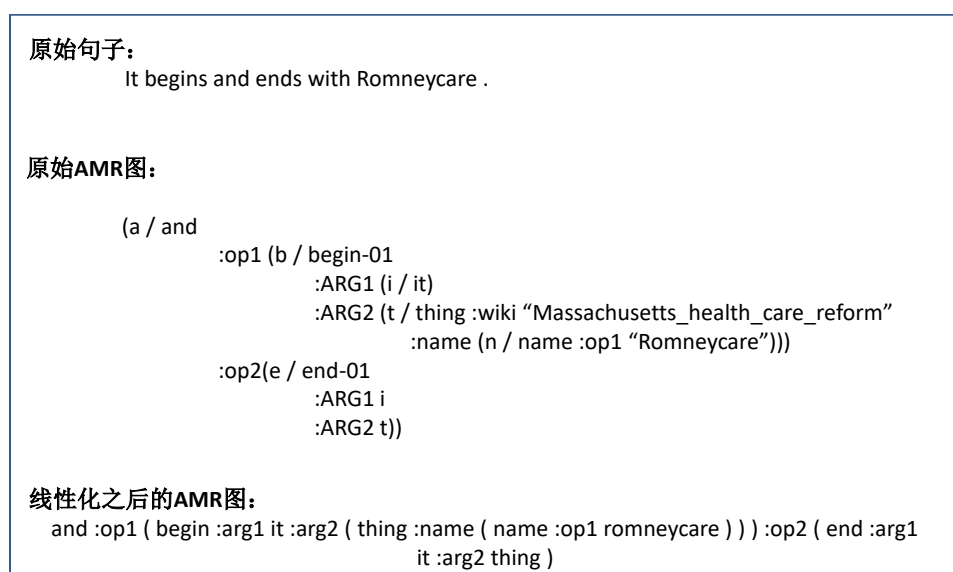


图 2. 一个AMR图线性化示例

2 相关工作

目前AMR-to-Text生成的任务大致可以分为两类：基于统计的方法和基于神经网络的方法。而基于神经网络的方法，现在又可以分为seq2seq和graph2seq两类。

2.1 基于统计的方法

早期神经网络未普及时候，在AMR-to-Text生成上的工作大都使用基于统计的方法。Flanigan等(2016)将AMR图转换为合适的生成树，并应用树-串（tree-to-string）转换器生成文本。Song等(2016)将一个AMR图拆分成了许多小的片段，并生成所有片段的翻译，最终通过采用非对称广义旅行商问题解法来对片段确定其顺序。Song等(2017)使用同步节点替换语法来对AMR图进行解析，并生成相应的句子。Pourdamghani等(2016)采用基于短语的机器翻译模型来对线性化AMR图进行建模。

2.2 基于神经网络的方法

随着神经网络的兴起，最近的研究都是使用神经网络来生成。在Sutskever等(2014)证明了深度神经网络的优越性之后，Konstas等(2017)提出使用序列到序列（S2S）模型来生成文本，利用双向LSTM来对线性化的AMR图进行编码。为了限制生成的文本具有更合理的句法，Cao等(2019)将AMR-to-Text生成的任务拆分成两个步骤，先使用句法模型去预测最优的目标端句法结构，再利用预测的句法信息去辅助生成模型更好的生成句子。但是也相应的损失了深度神经网络端到端的特性，并且和本文方法相比更加复杂，增加了网络的复杂度和参数。

随后，为了解决seq2seq模型将AMR图线性化之后信息损失的问题，大家的研究热点都着重在研究图神经网络上。图到序列（Graph-to-Sequence）模型常优于序列到序列（S2S）模型，包括图状态LSTM(2018),GGNN(2018)等。图状态LSTM通过每步的迭代交换相邻节点的信息来更新节点。同时也对每个节点增加一个向量单元保存历史信息。GGNN是一个基于门控的图神经网络，将AMR图结构完整的融入模型中，并且将边信息也转化为节点，解决了参数爆炸问题的同时，也给了解码器更丰富的信息。着重于解决AMR图中重入节点的问题，Damonte等(2019)提出了一种堆栈式的编码器，由图卷积神经网络和双向LSTM堆栈而成。Guo等(2019)提出了一种深度连接图卷积网络（GCN）更好的获取局部与非局部信息。Zhu等(2019)在Transformer的基础上，受到(Shaw et al., 2018)对相对位置建模的启发，提出了一种Structure-Aware Self-Attention的编码方法可以对图结构中任意两两节点进行完整的建模（不论节点之间是否直接相连），在该任务上取得了最高的性能。

3 方法

本文采用了两种方法作为基准模型 (Baseline)。

1. Transformer, 最先进的seq2seq模型, 最初使用于神经机器翻译和句法分析任务(Vaswani et al., 2017)。
2. Zhu等(2019)提出的Structure-Aware Self-Attention模型, 目前在AMR-to-Text生成的任务上取得了最高的性能。

3.1 基准模型1 (Baseline1)

3.1.1 Transformer

Baseline1是使用的Transformer模型, 它采用了编码器-解码器 (Encoder-Decoder) 的架构, 由许多编码器和解码器堆栈组成。每一个编码器都存在两个子层: 自注意力机制层 (self-attention) 后面紧接着前馈神经网络层 (position-wise feed forward)。自注意力层使用了多个注意力头 (attention head), 将每个注意力头的结果进行连接和转换之后, 形成自注意力机制层的输出。每个注意力头使用点乘注意力机制 (scaled dot-product) 来计算输入一个序列 $x = (x_1, \dots, x_n)$, 得到一个同样长度的新的序列 $z = (z_1, \dots, z_n)$:

$$z = \text{Attention}(x) \quad (1)$$

其中 $x_i \in R^{d_x}$, $z \in R^{n \times d_z}$ 。每一个输出元素 z_i 是输入元素的线性变换的加权和:

$$z_i = \sum_{j=1}^n \alpha_{ij} (x_j W^V) \quad (2)$$

其中 $W^V \in R^{d_x \times d_z}$ 是一个可学习的参数矩阵。公式2中的向量 $\alpha_i = (\alpha_{i1}, \dots, \alpha_{in})$ 是通过自注意力机制模型得到的, 该机制捕获了 x_i 和其它元素之间的对应关系。具体来说, 每个元素 x_j 的自注意力权重 α_{ij} 是通过一个softmax函数计算得到:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^n \exp(e_{ik})} \quad (3)$$

其中

$$e_{ij} = \frac{(x_i W^Q)(x_j W^K)^T}{\sqrt{d_z}} \quad (4)$$

是一个对齐函数, 它用来度量输入元素 x_i 和 x_j 的匹配程度。 $W^Q, W^K \in R^{d_x \times d_z}$ 是可学习的参数矩阵。

3.1.2 线性化预处理

因为Transformer是seq2seq模型, 输入只支持序列化的输入, 所以需要AMR图进行线性化的预处理。本文采用Konstas等(2017)提出的深度优先遍历的线性化方法来对AMR图进行预处理, 从而得到简化版的AMR图。在线性化之前, 首先移除了图中的变量、wiki链接和语义标签。图2展示了一个AMR图线性化示例。

3.2 基准模型2 (Baseline2)

3.2.1 Structure-Aware Self-Attention

Zhu等(2019)扩展了传统的自注意力机制框架, 提出了一种新颖的结构化的注意力机制, 在对齐函数中显式地对元素对 (x_i, x_j) 之间的关系进行编码, 用公式5替换公式4。

$$e_{ij} = \frac{(x_i W^Q)(x_j W^K + r_{ij} W^R)^T}{\sqrt{d_z}} \quad (5)$$

x_i	x_j	结构标签序列
begin-01	and	:ARG1↑
begin-01	Romneycare	:ARG2↓ :name↓
begin-01	begin-01	None

表 1. 图1中一些概念对之间的结构路径示例。

其中 $W^R \in R^{d_z \times d_z}$ 是一个参数矩阵. 然后, 再相应地更新公式2, 将结构信息传播到子层的输出。

$$z_i = \sum_{j=1}^n \alpha_{ij} (x_j W^V + r_{ij} W^F) \quad (6)$$

其中, $W^F \in R^{d_z \times d_z}$ 是一个参数矩阵. $r_{ij} \in R^{d_z}$ 代表了元素对 (x_i, x_j) 之间的关系, 它是通过3.2.2学习到的一个向量表示。

3.2.2 学习图概念 (concept) 对之间的向量表示 r

上述的Structure-Aware Self-Attention机制可以用来获取到图中任意两两概念对 (concept pairs) 之间的图结构关系。定义使用沿着概念 x_i 到 x_j 之间边标签 (edge label) 组成的一条路径当作概念对之间的图结构关系⁰。同时, 为了区分方向, 也给每条边标签相应的增加了方向符号。Table 1展示了图 1中的几个概念对之间的结构标签序列。

现在已经给定了一个结构标签路径 $s = s_1, \dots, s_k$, 然后获取到它的向量表示 $l = l_1, \dots, l_k$, 最后本文使用基于卷积神经网络(Kalchbrenner et al., 2014) (CNN-based)¹ 的方法来获得公式 5和公式 6中的向量表示 r_{ij} 。

CNN-based

使用CNN来卷积标签序列 l 获得一个向量 r :

$$\begin{aligned} conv &= Conv1D(kernel_size = (m), \\ &\quad strides = 1, \\ &\quad filters = d_z, \\ &\quad input_shape = d_z \\ &\quad activation = 'relu') \end{aligned} \quad (7)$$

$$r = conv(l) \quad (8)$$

实验中 m 的大小常设置为4。

3.3 数据稀疏性

在训练AMR-to-Text模型的时候, 因为语料数量的限制, 常常会受到数据稀疏性的影响。为了解决这个问题, 前人的工作有采用匿名化的方法来删除命名实体和罕见词(Konstas et al., 2017), 或者使用复制机制(Gulcehre et al., 2016)来学习, 使模型可以学会从源端输入复制未登录词到目标端。在本文中, 我们提出使用字节对编码 (BPE) (Sennrich et al., 2016)将未登录词拆分成更细粒度, 更高频的单词。再根据该任务的特性考虑, 共享了源端和目标端的词表。Zhu等(2019)也在实验中证明了该方法的有效性。

3.4 融合句法信息

前人的工作都是使用平行语料来进行训练, 输入源端AMR图去生成对应的句子。他们大都是将句子视为单词序列, 但是却忽略了句子本身的一些外部知识, 没有考虑到句子中潜藏的句

⁰当同时存在多条路径组合时, 默认选择最短的那一条。

¹(Zhu et al., 2019)使用了多种方法来学习图结构表示方法, 本文选择了CNN-based这一方法作为基线模型。

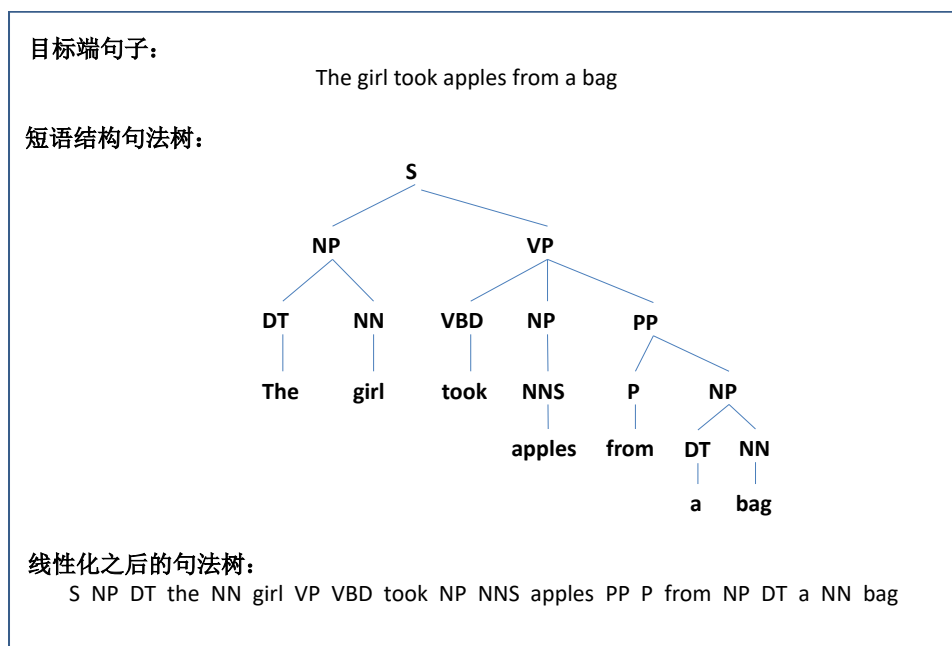


图 3. 一个目标端句子解析的短语结构句法树及其线性化示例

法信息。为了使模型能够学习到目标端句子的句法信息和内部结构，本文提出一种显式的方法来融入目标端的句法信息。

融合目标端句法信息的基本思想是将目标端句子经过解析得到句法树，之后再通过深度优先遍历得到最终的句法标签序列。句法的标注形式大致有两种，短语结构句法树和依存句法树。本文在训练时选择使用线性化的短语结构句法树(Vinyals et al., 2015)来替换目标端的句子。如图3所示，给出了一个目标端句子解析的短语结构句法树和其线性化结果。之所以选择短语结构句法树，是因为与依存树相比，它具有良好的线性化顺序的优点。此外，短语结构句法树也更容易实现，因为它们有效的对应句子中单词的顺序。在解码阶段，只需要将句法标签去除之后，就是最终预测生成的句子。

不幸的是，AMR标注数据并没有发放句法标注数据。因此，本文使用斯坦福解析器(Stanford Parser)(Manning et al., 2014)解析训练集和验证集语料，从而获得对应结构语法树的银语料(Silver-Standard)。

4 实验

4.1 数据集

为了评估方法的有效性，本文使用LDC发行的现存的两份标准英文语料集进行实验，分别是LDC2015E86和LDC2017T10。两份语料集分别包含了16,833和36,521条训练数据，并且共享了1,368条验证集和1,371条测试集。训练集和验证集使用斯坦福解析器²获取到目标端句子所对应的Penn treebank-style风格的结构句法树。

4.2 实验设置

本文分别通过使用10K和20K的操作数来对LDC2015E86和LDC2017T10两份语料进行BPE操作。从BPE处理之后的训练集中根据词频建立词汇表，参考Ge等(2019)的工作，共享了源端和目标端的词汇表。为了公平的对比，模型中的词向量使用随机初始化的方式。

本文使用OpenNMT(Klein et al., 2017)的框架作为Transformer的基准模型³。在超参数的设置上，模型的编码器和解码器为6层。在优化器方面，本文使用 $\beta_1=0.1$ (Kingma and Ba, 2015)的Adam优化算法。自注意头的数量设置为8。此外，模型中向量和隐藏状态的维度位置为512，批处理大小(batch size)设置为4096。为了模型计算速度考虑，限定路径标签的最大

²<https://nlp.stanford.edu/software/lex-parser.html>

³<https://github.com/OpenNMT/OpenNMT-py>

System	LDC2015E86			LDC2017T10		
	BLEU	Meteor	chrF++	BLEU	Meteor	chrF++
(Konstas et al., 2017)*	22.00	-	-	-	-	-
(Cao and Clark, 2019)*	23.5	-	-	26.8	-	-
(Song et al., 2018) [†]	23.30	-	-	-	-	-
(Beck et al., 2018) [†]	-	-	-	23.3	-	50.4
(Damonte and Cohen, 2019) [†]	24.40	23.60	-	24.54	24.07	-
(Guo et al., 2019) [†]	25.7	-	-	27.6	-	57.3
(Zhu et al., 2019)(CNN-based) [†]	29.10	35.00	62.10	31.82	36.38	64.05
(Song et al., 2016) [‡]	22.44	-	-	-	-	-
Baseline1	25.13	33.08	59.36	26.98	34.36	61.05
+ Syntax	26.39	33.63	59.84	28.13	34.82	61.73
Baseline2	28.64	34.83	61.89	31.10	36.07	63.87
+ Syntax	29.71	35.49	62.52	32.28	36.81	64.61

表 2. 本文方法在LDC2015E86和LDC2017T10测试集上的实验结果及与其它模型的比较。* 代表seq2seq模型, † 代表graph2seq模型, ‡ 代表其它模型。

长度为 4。解码时, 默认的额外长度从50增加至150, 该值表示模型解码时允许生成句子长度是源端最大长度加150。在所有实验中, 以学习率 0.5 在 Tesla P40 GPU上训练 300K 步停止。本文实验代码已开源公布至<https://github.com/Amazing-J/structural-transformer>。

为了更好的体现本文方法的有效性, 采用了BLEU(Papineni et al., 2002), Meteor(Banerjee and Lavie, 2005; Denkowski and Lavie, 2014), chrF++(Popović, 2017)三种评测指标。BLEU是基于语料级的评估, 后两者是基于句子级的评估。相对来说, 后两者的分数更接近于人工评测。

4.3 实验结果

表2给出了本文两个基准模型在融合了目标端句法信息前后AMR-to-Text生成的性能对比。从表2可以看出, 融合目标端句法信息之后, AMR-to-Text生成的性能有着显著的提升。在两个基准模型上, 分别提高了1.26和1.07 (LDC2015E86), 1.15和1.18 (LDC2017T10) BLEU。这也有力的证明, 在目标端融入句法信息, 可以帮助模型学习到句子中潜藏的一些知识, 从而在生成时考虑到句法信息的约束。该方法与融合源端句法和语义角色信息的机器翻译方法类似(Li et al., 2013), 本文进一步证明了在生成任务中目标端融入句法信息同样可以有着显著的提升。

表 2也给出了与其它现存模型在该任务上的性能比较。值得注意的是, LDC2015E86和LDC2017T10的验证集和测试集是相同的, 区别只是训练集的数量相差了一倍左右。从表2可以看到, 与seq2seq模型相比, 本文的baseline1就已经显著的超越了它们, 并且在融入句法信息 (+Syntax) 之后, 性能依然有着明显的提升。目前最高的性能是Zhu等(2019)提出的Structure-Aware Self-Attention模型, 本文在它们的基础之上也同样有着有效的提高, 创造了新的最高的性能 (SOTA)。可以证明本文的方法无论用在seq2seq模型或者graph2seq模型上都有效。

4.4 参数数量和训练时间

本文融合目标端句法信息的方法是将目标端句子替换为线性化结构句法树, 不会对模型进行任何修改, 也就意味着并不会给模型增加参数, 这也是本文方法的一大优点。但是, 目标端句子替换成线性化结构句法树之后, 它的序列长度会相应的变长, 这就会导致训练的时间略微增加。据统计, 本文baseline1基准模型在LDC2015E86上进行训练, 完场一轮训练的时间大概需要288秒 (约4.80分钟), 而融合目标端句法信息之后, 大概花费345秒 (约5.75分钟)。

4.5 融合不同形式句法信息的影响

从实验结果可以得到融合目标端句子的句法信息可以显著的提升AMR-to-Text生成的性能, 但是为了探究哪种形式的句法信息对生成性能最为有效, 本文做了进一步的实验分析。

一	(S (NP (DT the)DT (NN girl)NN (VP (VBD took)VBD (NP (NNS apples)NNS (PP (P from)P (NP (DT a)DT (NN bag)NN)NP)PP)NP)VP)NP)S
二	S NP the girl VP took NP apples PP from NP a bag
三	S NP DT the NN girl VP VBD took NP NNS apples PP P from NP DT a NN bag

表 3. 三种线性化结构句法树的示例

	一	二	三
BLEU	24.84	25.89	26.39

表 4. 三种线性化结构句法树在LDC2015E86测试集的性能对比

如表3所示，本文探索了三种线性化短语结构句法树对生成性能的影响。第一种形式包括了一个完整的句法树，不仅保留了句法树的所有节点，还给相应节点增加了结束标签，例如)NN,)NP,)S等。第二种形式则没有增加节点的结束标签，并且为了缩减句子的长度，把句法树中的词性标签删除，仅保留句法树的主干成份，剔除如DT, NN, VBD等词性标签。第三种形式则是在第二种形式的基础上保留了单词的词性标签。

本文在LDC2015E86的测试集上，对上述三种句法树的形式做了实验。从表4可以看出，当使用第一种形式时候，性能最差。因为它会将目标端的句子长度几倍的增长，极大地增加了模型的学习难度。从第二种和第三种形式得到的性能可以看出，保留短语结构句法树中的词性标签信息是对生成有着明显的贡献，有着0.5BLEU值的提升。本文则是使用的第三种短语结构句法树的形式。

5 结论

本文提出了一种直接而有效的方法融合目标端句子的句法信息，并且在最优的seq2seq模型Transformer上以及AMR-to-Text生成任务中最优的模型上都进行了实验。实验结果表明，使用该方法可以有效的对目标端句子的句法信息进行学习，从而提高AMR-to-Text生成的性能。在该任务最优模型的基础上同样也有着 1.07 和 1.18 BLEU值的提升，创立了新的最高性能。未来的工作中，由于句法分析和生成任务有着比较高的关联性，所以将会探索句法分析与AMR-to-Text生成任务之间的联合学习方向。

参考文献

- Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of 7th Linguistic Annotation Workshop & Interoperability with Discourse*, pages 178–186.
- Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of ACL*, pages 65–72.
- Daniel Beck, Gholamreza Haffari, and Trevor Cohn. 2018. Graph-to-sequence learning using gated graph neural networks. In *Proceedings of ACL*, pages 273–283.
- Kris Cao and Stephen Clark. 2019. Factorising amr generation through syntax. In *Proceedings of NAACL*, pages 2157–2163.
- Marco Damonte and Shay B. Cohen. 2019. Structural neural encoders for AMR-to-text generation. In *Proceedings of NAACL*, pages 3649–3658.
- Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of WMT*, pages 376–380.
- Jeffrey Flanigan, Chris Dyer, Noah A. Smith, and Jaime Carbonell. 2016. Generation from abstract meaning representation using tree transducers. In *Proceedings of NAACL*, pages 731–739.
- DongLai Ge, Junhui Li, Muhua Zhu, and Shoushan Li. 2019. Modeling source syntax and semantics for neural amr parsing. In *Proceedings of IJCAI*, pages 4975–4981.

- Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. In *Proceedings of ACL*, pages 140–149.
- Zhijiang Guo, Yan Zhang, Zhiyang Teng, and Wei Lu. 2019. Densely connected graph convolutional networks for graph-to-sequence learning. *Transactions of the Association of Computational Linguistics*, 7:297–312.
- Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of ACL*, pages 655–665.
- Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of ICLR*.
- Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. In *Proceedings of ACL, System Demonstrations*, pages 67–72.
- Ioannis Konstas, Srinivasan Iyer, Mark Yatskar, Yejin Choi, and Luke Zettlemoyer. 2017. Neural AMR: Sequence-to-sequence models for parsing and generation. In *Proceedings of ACL*, pages 146–157.
- Junhui Li, Resnik Philip, and Daumé III Hal. 2013. Modeling syntactic and semantic structures in hierarchical phrase-based translation. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pages 540–549.
- Xiang Li, Thien Huu Nguyen, Kai Cao, and Ralph Grishman. 2015. Improving event detection with abstract meaning representation. In *Proceedings of the first workshop on computing news storylines*, pages 11–15.
- Junhui Li, Deyi Xiong, and Zhaopeng Tu. 2017. Modeling source syntax for neural machine translation. In *Proceedings of ACL-2017*, pages 688–697.
- Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of ACL-2014*, pages 55–60.
- Arindam Mitra and Chitta Baral. 2016. Addressing a question answering challenge by combining statistical methods with inductive rule learning and reasoning. In *Thirtieth AAAI Conference on Artificial Intelligence*.
- Kishore Papineni, Salim Roukos, Ward Todd, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of ACL*, pages 311–318.
- Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of WMT*, pages 612–618.
- Nima Pourdamghani, Kevin Knight, and Ulf Hermjakob. 2016. Generating english from abstract meaning representations. In *Proceedings of the 9th International Natural Language Generation conference*, pages 21–25.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of ACL*, pages 1715–1725.
- Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of NAACL*, pages 464–468.
- Lin Feng Song, Yue Zhang, Xiaochang Peng, Zhiguo Wang, and Daniel Gildea. 2016. Amr-to-text generation as a traveling salesman problem. In *Proceedings of EMNLP*, pages 2084–2089.
- Lin Feng Song, Xiaochang Peng, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2017. Amr-to-text generation with synchronous node replacement grammar. In *Proceedings of ACL*, pages 7–13.
- Lin Feng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2018. A graph-to-sequence model for AMR-to-text generation. In *Proceedings of ACL*, pages 1616–1626.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

- Aleš Tamchyna, Chris Quirk, and Michel Galley. 2015. A discriminative model for semantics-to-string translation. In *Proceedings of the 1st Workshop on Semantics-Driven Statistical Machine Translation (S2MT 2015)*, pages 30–36, Beijing, China, July. Association for Computational Linguistics.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of NIPS*, pages 5998–6008.
- Oriol Vinyals, Lukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar as a foreign language. In *Advances in Neural Information Processing Systems 28*, pages 2773–2781.
- Jie Zhu, Junhui Li, Muhua Zhu, Longhua Qian, Min Zhang, and Guodong Zhou. 2019. Modeling graph structure in transformer for better amr-to-text generation. In *Proceedings of EMNLP-2019*, pages 5458–5467.

JCL 2020