软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software, [doi: 10.13328/j.cnki.jos.006207]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于多任务预训练的 AMR 文本生成研究*

徐东钦[1], 李军辉[1], 朱慕华[2], 周国栋[1]

[1](苏州大学 计算机科学与技术学院,江苏 苏州 215006)

[2](腾讯科技（北京）有限公司 腾讯新闻, 北京, 100001)

通讯作者: 李军辉, E-mail: jhli@suda.edu.cn

摘 要: 抽象语义表示(Abstract Meaning Representation,简称 AMR)文本生成的任务是给定 AMR 图,生成与其语义一致的文本.相关工作表明,人工标注语料的规模大小直接影响了 AMR 文本生成的性能.为了降低对人工标注语料的依赖,本文提出了基于多任务预训练的 AMR 文本生成方法.特别地,基于大规模自动标注 AMR 语料,本文提出与 AMR 文本生成任务相关的三个预训练任务，分别是 AMR 降噪自编码、句子降噪自编码以及 AMR 文本生成任务本身.此外,基于预训练模型,本文在朴素微调方法的基础上进一步提出了基于多任务训练的微调方法,使得最终模型不仅适用于AMR文本生成,同时还适用预训练任务.基于两个AMR标准数据集的实验结果表明,使用0.39M自动标注数据,本文提出的预训练方法能够大幅度提高 AMR 文本生成的性能,在 AMR2.0 和 AMR3.0 上分别提高了12.27和7.57个BLEU值,性能分别达到40.30和38.97.其中,在AMR2.0上的性能为目前报告的最优值,在AMR3.0上的性能达到了以往未曾达到的性能.

关键词: AMR;AMR 文本生成;多任务预训练;序列到序列模型

中图法分类号: TP391.1

中文引用格式: 徐东钦,李军辉,朱慕华,周国栋. 基于多任务预训练的 AMR 文本生成研究.软件学报. http://www.jos.org.cn/1000-9825/6207.htm

英文引用格式: Xu D, LI J, ZHU M, ZHOU G. Improving AMR2Text Generation with Multi-Task Pre-Training. Ruan Jian Xue Bao/Journal of Software, (in Chinese). http://www.jos.org.cn/1000-9825/6207.htm

Improving AMR-to-Text Generation with Multi-Task Pre-Training

XU Dong-Qin[1], LI Jun-Hui[1], ZHU Mu-Hua[2], ZHOU Guo-Dong[1]

[1](School of Computer Science and Technology, Soochow University, Suzhou 215006, China)

[2](Tencent News, Tencent Technology （Beijing） Co. Ltd., Beijing 100001, China)

Abstract: Given an AMR (Abstract Meaning Representation) graph, AMR-to-Text generation aims to generate text with the same meaning. Related studies show that the performance of AMR-to-Text severely suffers from the size of the manually annotated dataset. To alleviate the dependence on manually annotated dataset, in this paper we propose a novel multi-task pre-training for AMR-to-Text generation. In particular, based on large-scale automatic AMR dataset, we define three relevant pre-training tasks, i.e., AMR denoising auto-encoder, sentence denoising auto-encoder, and AMR-to-Text generation itself. In addition, to fine-tune the pre-training models, we further extend the vanilla fine-tuning method to multi-task learning fine-tuning, which enables the final model maintain performance on both AMR-to-Text and pre-training tasks. With automatic dataset of 0.39M sentences, detailed experimentation on two AMR benchmarks shows that the proposed pre-training approach significantly improves the performance of AMR-to-Text generation, with improvement of 12.27 BLEU on AMR2.0 and 7.57 on AMR3.0, respectively. This greatly advances the state-of-the-art performance with 40.30 BLEU on AMR2.0 and 38.97 on AMR 3.0, respectively. To our best knowledge, this is the best result achieved so far on AMR 2.0 while we first report AMR-to-Text generation performance on AMR 3.0.

* 基金项目: 国家重点研发计划项目(2017YFB1002101); 国家自然科学基金(61876120)
Foundation item: National Natural Science Foundation of China (61876120)
收稿时间: 2020-07-30; 修改时间: 2020-10-19; 采用时间: 2020-11-18; jos 在线出版时间: 2020-12-02

Key words: abstract meaning representation; AMR-to-Text generation; multi-task pre-training; sequence-to-sequence

抽象语义表示(Abstract Meaning Representation,简称 AMR)[1]是一种新型的语义表示方法,它将自然语言文本以句子为单位抽象成单根有向无环图.如图 1 所示,“word-01”和“tend-02”等图结点称作为概念(concept),自然语言中的实词被映射为 AMR 图中的概念结点;结点之间的边表示两个概念之间的语义关系,如“:ARG0”和“:op1”等. AMR 作为句子的语义表示已经被广泛应用于机器翻译[2],问答系统[3],事件抽取[4]等自然语言处理相关任务中.AMR 文本生成指给定 AMR 图,自动生成语义上一致的自然语言文本.如图 1 所示,该问题可以看作是一个图(即 AMR 图)到序列(即文本句子)的转换任务,在复述生成、机器翻译等自然语言处理相关任务中有着潜在的应用，同时对图到序列、树到序列任务的算法研究提供借鉴意义,近年来越来越受到人们的关注[2,3,4,5,6,7,8,9,10].

与其它自然语言处理任务相似,AMR 文本生成的性能受标注语料规模大小的影响.例如,基于相同的测试集,当训练集由包含 16,833 条训练样例的 AMR1.0 切换为包含 36,521 条训练样例的 AMR2.0 时,文献[10]在测试集上的性能由 25.50 显著提升至 27.43.这说明,通过增加标注语料的规模,能够进一步提高 AMR 文本生成的性能.为了降低对人工标注语料的依赖,一种可行的解决方案是借助预训练技术.先通过在大规模无标注文本语料上训练深层网络结构,从而得到一组模型参数;然后将预训练好的部分或全部模型参数应用到后续任务,如本文的 AMR 文本生成.然而,在通用领域的预训练模型,如 ELMo[11]、GPT-2[12]和 BERT[13]等模型并不能直接用于 AMR 文本生成.其主要原因在于,ELMo、GPT-2 和 BERT 等预训练模型皆训练自大规模自然语言文本,而在 AMR 文本生成任务中,源端是带结构化信息的 AMR 图.即使能够将 AMR 图线性化为序列,但该序列中存在着大量的 AMR 图标签,如图 1 所示的边标签“:ARG0”、“:mod”等和结点标签“word-01”、“tend-02”等.因此,已训练好的并公开的预训练模型并不适合直接用于本文 AMR 文本生成任务.

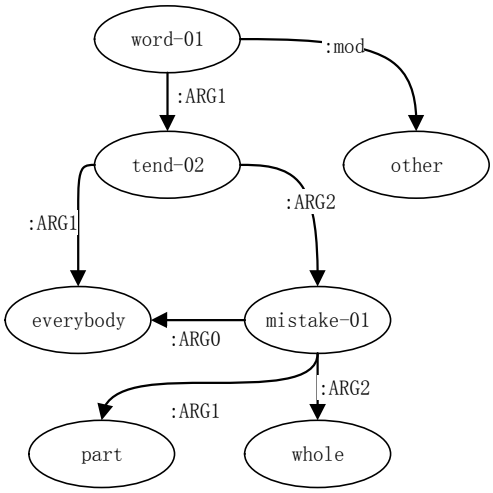


Fig. 1 AMR graph for sentence “In other words , everybody tends to mistake the part for the whole.”

图 1 句子“ In other words , everybody tends to mistake the part for the whole.”对应的 AMR 图

为此,针对 AMR 文本生成任务,本文提出了与该任务相关的三个预训练任务.同时,为了能够更好地捕获 AMR 和自然语言文本之间的关联,本文进一步提出了联合多任务的预训练.特别地,本文的预训练任务和 AMR 文本生成任务皆采用通用的序列到序列框架.于是,可以直接将预训练模型的所有参数应用于 AMR 文本生成任务.基于预训练模型,传统的微调方法仅使用 AMR 文本生成任务的训练集进一步优化模型参数.但该微调方法会导致模型出现“灾难性遗忘”问题,即微调后模型将会失去对预训练任务的预测能力,不再适用于原预训练任务.受 Li 和 Hoiem[14]的启发,本文提出多任务训练框架对 AMR 文本生成进行微调,使得微调后模型同时适用

于 AMR 文本生成和预训练任务.基于两个 AMR 标准数据集 AMR2.0 和 AMR3.0 的实验结果表明,借助于 0.39M 的自动标注句子, 本文提出的预训练方法能够大幅度提高 AMR 文本生成的性能,分别达到 40.30 和 38.97 BLEU 值.其中 AMR2.0 上的性能为目前报告的最高性能,AMR3.0 上的性能为目前为止首次报告的性能.

本文的主要贡献包括:

(1) 针对 AMR 文本生成任务,提出了三种相关的预训练任务和其联合预训练任务,并分析比较了各种预训练任务在不同微调方式下的性能;

(2) 基于预训练模型,分析比较了两种不同微调方式的性能;

(3) 基于 0.39M 的自动标注句子,本文在 AMR2.0 和 AMR3.0 上均得取目前的最优性能.

本文第 1 节对相关工作进行描述.第 2 节描述基于序列到序列的 AMR 文本生成.第 3 节给出本文提出的基于多任务预训练的 AMR 文本生成.第 4 章设计实验并通过实验分析表明本文方法的有效性.第 5 章从多方面进一步分析预训练对 AMR 文本分析性能的影响.最后总结全文,并对未来值得关注的研究方向进行初步探讨.

1 相关工作

本文的研究工作主要涉及 AMR 文本生成和模型预训练两个方面.因此,本节将从这两个角度来总结相关研究工作.

1.1 AMR文本生成

AMR 文本生成是一个典型的图到序列的任务.早期研究中多采用基于规则的方法来解决这个任务.Flanigan 等人[6]使用两阶段方法,根据重度结点(Reentrancy,即具有多个父亲结点的概念结点)将 AMR 图拆分成多个树结构后,再使用规则的方法将树结构翻译为文本序列.Song 等人[7]使用启发式提取算法来学习图到字符串(graph-to-string)规则.

目前更多的研究将 AMR 文本生成任务视为机器翻译任务,并通过深度优先遍历来获取线性化的 AMR 图.比如, Pourdamghani 等人[8]和 Ferreira 等人[9]使用基于短语的机器翻译模型将线性化 AMR 图翻译为自然语言文本;Konstas 等人[15]利用序列到序列的神经机器翻译模型将线性化 AMR 图转换为自然语言文本.Cao 和 Clark[16]则使用目标端的语法信息提高了基于序列到序列方法的 AMR 文本生成性能.

在某种程度上,AMR 图线性化得到的序列不可避免地会丢失原图中的结构化信息.为了减少线性化过程带来的信息损失,目前越来越多的研究提出基于图到序列(Graph2Seq)的神经网络模型.例如,Marcheggiani 和 Perez-Beltrachini[17]首次利用图神经网络(graph neural networks)来显式地编码图结构信息,并显著提高了文本生成性能.之后,研究者不断提出多种变体的图编码器模型,如基于图的 LSTM[18],门控图神经网络(gated graph neural networks,GGNN)[19]和图卷积神经网络(graph convolutional networks,GCN)[20].Guo 等人[21]使用了密集连通(dense connection)网络,允许不同层之间的信息进行交换.此外,为了更好地对 AMR 图进行编码,Ribeiro 等人[22]使用自上而下和自下而上的双向图表示方法.Zhu 等人[10]、Cai 和 Lam[23]提出了基于结构驱动的 Transformer[24]模型,对 AMR 结构信息进行编码.Zhao 等人[25]根据 AMR 中的概念与边关系,分别从概念图与关系图的角度对图结构进行编码.Song 等人[26]使用基于图结构的自编码,同时将编码后的图结构信息还原为序列化 AMR 与三元组关系,以减少对原 AMR 图结构的损失.

1.2 模型预训练

目前在自然语言处理任务的应用中,预训练模型在各种下游任务中的优越表现,使用预训练模型已经成为一种主流的做法.本文将预训练模型大致分成三类,第一类是学习静态词嵌入的预训练模型,如 word2vec[27]、GloVe[28]等,第二类是捕获上下文语境的预训练模型,如 CoVe[29], ELMo[11],GPT-2[12],和 BERT[13]等,第三类是基于序列到序列模型的预训练模型,如 PoDA[30],MASS[31],BART[32]等.

由于 AMR 需要对图结构进行编码,同时 AMR 中包含许多特殊符号,使得基于自然语言文本的预训练模型无法直接应用到 AMR 文本生成中.目前基于预训练的 AMR 文本生成研究较少.因此,如何针对 AMR 文本生成

任务进行模型预训练是一个亟待研究的课题.Mager 等人[33]和 Harkous 等人[34]首次在 AMR 文本生成任务中引入预训练模型,显著提升了文本生成的性能.

本文将 AMR 文本生成任务看作序列到序列任务,根据 AMR 文本生成任务的特点,提出了多种针对 AMR 文本生成的预训练任务,并使用目前效果显著的 Transformer 作为预训练模型.与 Mager 等人[33]和 Harkous 等人[34]所使用的预训练模型不同的是,本文使用的预训练模型为序列到序列结构.

2 基于序列到序列的 AMR 文本生成

本文使用目前综合性能最佳的 Transformer 序列到序列模型作为 AMR 文本生成的基准模型.Transformer 模型由编码器与解码器组成,而编码器与解码器又分别由多个堆叠的编码器层与解码器层组成.编码器层包括自注意力层(Self-Attention Layer)和全连接前馈神经网络(Position-wise Feed-Forward Networks, FFN),解码器则包括自注意力层、编码器与解码器注意力层(Encoder-Decoder Attention Layer)和全连接前馈神经网络.每两子层之间使用残差连接(Residual Connection)及层级正则化(Layer Normalization)处理子层间传递的数据.与基于循环神经网络和基于卷积神经网络的序列到序列模型相比,带多头注意力机制 Transformer 模型不仅能够捕获序列之间的长距离依赖,同时还能够并行处理数据.目前在机器翻译、句法树解析等任务中,Transformer 均取得了较好的性能.有关 Transformer 模型的更多细节,可以参阅 Vaswani 等[24]的论文.

2.1 AMR图线性化预处理

原始的句子:
In other words , everybody tends to mistake the part for the whole .

原始的 AMR 图:

```
(w2 / word-01
        :ARG1 (t / tend-02
                :ARG1 (e / everybody)
                :ARG2 (m / mistake-01
                        :ARG0 e
                        :ARG1 (p / part)
                        :ARG2 (w / whole)))
   :mod (o / other))
```

线性化处理后的 AMR 图:
word :arg1 (tend :arg1 everybody :arg2 (mistake :arg0 everybody :arg1 part :arg2 whole))

Fig. 2 An example of linearized AMR
图 2 一个 AMR 图线性化示例

本文使用了 Zhu 等人[10]实验中基准系统所使用的预处理方法.首先,使用深度优先遍历方法遍历 AMR 图,获得线性化 AMR 序列.其次,删除序列中的变量标签、谓词语义(predicate senses)后缀、wiki 链接标签和概念结点两侧的引号等,对所有单词小写化获取简化的线性化 AMR.特别地,(a)若 AMR 图中存在重度结点,则在删除重度结点的变量前,在该变量的位置复制一个重度结点,使其从 AMR 图变为 AMR 树;(b)若概念结点两侧仅有括号,则删除该位置两侧的括号;(c)删除线性化 AMR 图最外层的一对括号.如图 2 所示,简化的线性化 AMR 为一个符号序列,该序列中包括了原 AMR 图中的去除谓词语义标签的概念名称、概念之间的语义关系以及用于表示结构的括号.

2.2 子词化处理

在有限的数据规模下,数据稀疏问题显得尤为明显,该问题直接影响了深度学习中模型对特征的学习能力.为了解决数据稀疏问题,Konstas 等人[15]使用匿名机制来缓解实体和数字的稀疏,但这需要人为的制定规则,耗时耗力.Song 等人[18]使用了字 LSTM 加复制机制,但这增加了模型本身学习解决问题的复杂度.受启发于机器翻

译领域解决低频词翻译的方法,本文采用字节对编码(Byte Pair Encoding,简称 BPE)[35]方法.通过将低频词拆分为更小粒度的高频子词单元,该方法能够显著提升包括机器翻译在内的众多自然语言处理任务中模型对低频词的建模能力.

对于 AMR 文本生成任务,由于源端和目标端存在着大量相同的单词.如图 2 所示,线性化 AMR 图和原句子中均含有 mistake、part、whole 等相同的单词,也包含如 tend 和 tends、word 和 words 等共源的单词.于是,本文将源端和目标端一同进行子词化处理;并在处理完后,共享源端和目标端的词汇表,以建立起源端和目标端单词之间的联系.

3 基于多任务预训练的 AMR 文本生成

为了降低 AMR 文本生成性能对标注数据集规模的依赖,相关研究在借助大规模的自动标注数据,即给定未标注的大规模(英文)文本数据集,通过预先构建的 AMR 解析器将该数据集中的每个句子转换为其相应的 AMR 图.一旦获得自动标注数据集,传统的做法大体上可以分为两类,一种方式是混合自动标注数据和人工标注数据集得到新的训练数据集[36],另一种方式是先使用自动标注数据集训练得到预训练模型,然后再使用人工标注数据集进行微调[15,18,21].不同于以上两种传统做法,为了更有效地利用自动标注数据集,本文提出了基于单任务预训练和基于多任务预训练的 AMR 文本生成方法.

基于自动标注数据集,本文提出与 AMR 文本生成任务相关的三个预训练任务,分别是 AMR 降噪自编码、句子降噪自编码以及 AMR 文本生成任务本身.由于前两个预训练任务皆基于降噪自编码,本节将首先介绍基于序列到序列的降噪自编码预训练,然后再详细描述本文的单任务预训练方法以及多任务预训练方法,最后介绍本文实验中使用到的两种微调(Fine-Tuning)方法. 需要注意的是,本文所有的序列到序列模型均采用 Transformer 模型.此外,模型的损失函数均采用相对熵损失(Relative Entropy Loss)的方法计算.

3.1 基于序列到序列的降噪自编码预训练

本文针对 AMR 文本生成任务,提出基于序列到序列模型的降噪自编码[37]预训练任务.降噪自编码模型被广泛应用于基于序列到序列的自然语言处理任务中[30],表明降噪自编码模型即能够较好地捕获源端语言的特征表示, 同时也能够较好地生成目标端语言序列.

降噪自编码模型分为编码器与解码器两部分.给定单词序列 $x=\{x_i\}_{i=1}^n$,在预处理阶段首先对该单词序列进行噪声化,得到包含噪声的单词序列 $x'=\{x'_i\}_{i=1}^n$.编码器以 x' 作为输入,并编码得到其对应的隐藏状态序列 $h=\{h_i\}_{i=1}^n$;解码器试图消除噪声,根据隐藏状态序列 h 恢复为原本的单词序列 x.因此,降噪自编码模型可以看作是从带噪声序列恢复为无噪声序列的过程.

受 Devlin 等人[13]工作的启发,单词序列的噪声化使用了三种不同的噪声方式,分别为(1)随机重新设置序列中每个单词的位置,并限制新位置与其原始位置相距不超过 3 个位置.(2)以 10%的概率,将单词替换为[mask]标记.(3)以 10%的概率,将单词丢弃.

降噪自编码模型在预测单词 x_i 时,将噪声序列 x' 的上下文以及预测序列 x 的上文作为条件,即计算条件概率 $p(x\mid x')$,该概率可以分解为 $p(x\mid x')=\prod_{i=1}^{n}p(x_i\mid x',x_{<i})$.

通过降噪自编码预训练的噪声编码、降噪解码两个部分,可以使模型学习到文本序列的上下文语境的特征表示,同时不会对模型的语言理解能力造成影响.这一过程同样可以看作是训练生成式语言模型,通过有效利用该语言模型可以对给定文本序列进行文本纠错.

3.2 单任务预训练

根据 AMR 文本生成任务的特点,利用其源端(即 AMR 端)和目标端(即文本端)序列的特征,本文设计了三种不同的单任务预训练方案.

1. **基于源端序列的降噪自编码模型**(Denoising Auto-Encoder based on source sequence,以下简称 DAE(S)).

利用源端线性化处理后的 AMR 序列,预训练基于序列到序列的 AMR 降噪自编码模型.通过对 AMR 文本生成模型的编码器(encoder)进行训练,提高编码器对线性化 AMR 序列图结构的语言特征表示的捕获能力.不难看出,该预训练任务的源端与 AMR 文本生成任务的源端是相匹配的.因此,该预训练任务将能够提高 AMR 文本生成任务源端编码器的表示能力.

2. **基于目标端序列的降噪自编码模型**(Denoising Auto-Encoder based on target sequence,以下简称 DAE(T)).根据目标端输入的单词序列,预训练基于序列到序列的句子降噪自编码模型.通过对 AMR 文本生成模型的解码器(decoder)进行训练,提高解码器的文本生成能力.类似地,该预训练任务的目标端与 AMR 文本生成任务的目标端是相匹配的.该预训练任务能够提高文本生成的能力.
3. **基于自动标注数据的 AMR 文本生成**(AMR-to-Text,以下简称 A2T).利用大规模自动标注的 AMR 语料,预训练 AMR 文本生成任务.为了获取大规模自动标注的 AMR 语料,本文借助了 Ge 等人[38]实验中 AMR 解析基准模型作为 AMR 解析器,利用该 AMR 解析器获得大规模自动标注的 AMR 语料,经过后处理即可成为本文实验中所需的自动标注 AMR 语料.有关详细内容,可以参阅 Ge 等人[38]等的论文.

值得注意的是,预训练任务和微调任务使用相同的词表.

3.3 多任务预训练

通常来说,模型可以通过训练不同任务来获得捕获不同语言特征的能力.为了使模型在预训练过程中得到更加充分训练,本文组合多个单任务预训练从而提出多任务预训练方法.受多语言神经机器翻译中零次学习(zero-shot)[39]方法的启发,如表 1 所示,为了区分不同的预训练任务,本文在源端与目标端序列前加入不同的序列起始符号.

在序列中加入序列起始符号后,多任务预训练可以简单地使用 Transformer 模型依次迭代每个任务的批次数据,实现同时进行多个任务的训练.例如,模型在训练完第一个任务一个批次后,更新模型参数,再训练第二个任务一个批次,并更新模型参数,如此迭代.通过该方法,可以获取 4 种多任务预训练模型,分别为 DAE(S) + DAE(T)、DAE(S) + A2T、DAE(T) + A2T 和 DAE(S) + DAE(T) + A2T.

Table 1 Symbols of beginning of sentence for the three proposed pre-training tasks

表 1 多任务预训练设置的源端和目标端起始符号

预训练任务	源端起始符	目标端起始符
DAE(S)	<AMR_BOS>	<AMR_BOS>
DAE(T)	<TXT_BOS>	<TXT_BOS>
A2T	<AMR_BOS>	<TXT_BOS>

3.4 基于预训练模型的微调

本文使用的两种微调方法,分别是朴素微调方法(Vanilla Fine-Tuning,以下简称 Vanilla 微调)和多任务微调方法(Multi-Task-Learning Fine-Tuning,以下简称 MTL 微调).

Vanilla 微调方法是基于预训练模型,通过直接在 AMR 训练语料上进行训练、优化模型的参数,获取 AMR 文本生成模型.微调会对模型的共享参数进行调整,令 AMR 文本生成更具有区分性.该方法通常会使用较低的学习率,间接保留预训练模型所捕获到的某些语言特征信息.然而,受限于 AMR 语料的规模,Vanilla 微调方法在 AMR 训练语料上训练、优化预模型的参数时,存在潜在的过拟合问题.

受 Li 和 Hoiem[14]的启发,本文提出了一种基于多任务的微调方法 MTL,能够在准确优化 AMR 文本生成任务的同时,维持预训练模型在预训练任务上的性能.在该方法中,维持预训练模型在预训练任务上的性能可以看作是模型对 AMR 文本生成训练的正则化处理.与多任务预训练方法相似,MTL 方法利用序列起始符号区分不同任务,结合 AMR 文本生成与多个预训练任务,依次迭代每个任务的数据对模型进行训练微调.如图 3 所示,为对 DAE(S) + DAE(T) + A2T 预训练模型的 MTL 微调方法的示例,在微调 DAE(S) + DAE(T) + A2T 预训练模型

时,首先训练 DAE(S)任务一个批次的数据,并更新模型的参数,接着再训练 DAE(T)任务一个批次的数据并更新模型的参数,最后训练 A2T 任务(即 AMR 文本生成人工标注数据)一个批次的数据并更新模型的参数.如此对三个任务进行依次迭代,直至模型收敛,即可获得一个拥有较好性能的 AMR 文本生成模型.

特别地,由于 AMR 文本生成预训练任务与微调的下游任务是相同的,本文 MTL 方法不适用于该预训练任务.例如, 在使用 MTL 方法进行微调预训练模型 DAE(S) + DAE(T) + A2T 时,仅会保持训练 DAE(S) 和 DAE(T)的预训练任务,而 A2T 任务替换为原本的 AMR 文本生成任务.此外,在使用 MTL 微调模型进行 AMR 文本生成推理时,源端的输入起始符必须为<AMR_BOS>、目标端的输入起始符必须为<TXT_BOS>,与 AMR 文本生成预训练任务一致

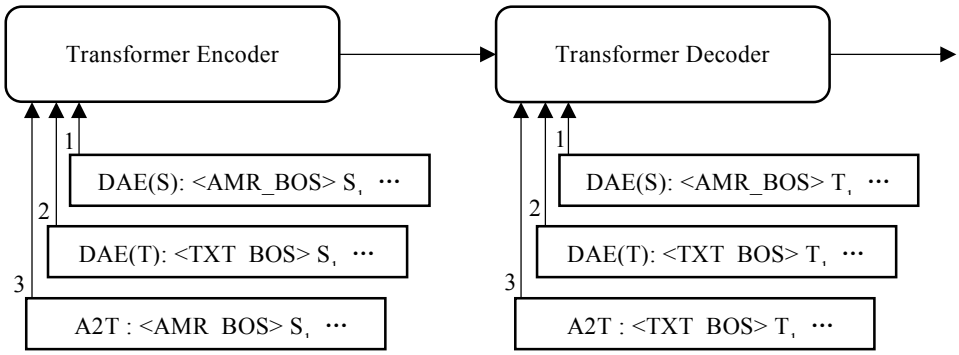


Fig. 3 Illustration of MTL Fine-Tuning for multi-task pre-training

图 3 MTL 微调方法示意图(以多任务预训练为例)

4 实验

本节首先介绍实验所用数据集以及评价标准,然后设计实验对本文的方法进行验证并对实验结果进行分析讨论.

4.1 实验数据集

预训练数据集 为了训练预训练模型,本文使用 WMT14 英语到德语翻译的 news-commentray-v11 数据集中的英语部分.该数据集共包括 390,961 英语句子.使用斯坦福分词工具进行分词后,平均每个句子包含 25 个单词.为了得到这些句子的自动 AMR 图,本文分别基于 AMR2.0 和 AMR3.0 训练得到两个基于序列到序列的 AMR 分析器.这两个 AMR 分析器在 AMR2.0 和 AMR3.0 的测试集上的性能分别是 73.21 和 74.65 Smatch F1 值 .紧接着,合并 AMR 人工标注数据集和自动标注数据集,使用 BPE 工具并设置操作数 2 万以获取子词词表,最后将人工标注数据和自动标注数据按照此词表进行子词化处理.

AMR 数据集 为了评测 AMR 文本生成的性能,本文使用两个 AMR 数据集,分别是 AMR2.0 和 AMR3.0.该两个数据集的统计如表 2 所示.值得注意的是,AMR3.0 中的训练集、开发集和测试集均包含了 AMR2.0 的相应部分.本文根据 2.1 节对 AMR 图进行线性化,同时对文本端使用斯坦福分词工具进行分词.然后,使用预训练数据集的子词词表,将线性化后的 AMR 图和分词后的文本进行子词化处理.

Table 2 Statistics on AMR2.0 and AMR3.0

表 2 AMR2.0 和 AMR3.0 统计信息

数据集	训练集	开发集	测试集
AMR2.0	36,521	1,368	1,371
AMR3.0	55,635	1,722	1,898

4.2 模型设置

本文使用 Open-NMT 作为 Transformer 模型的实现.模型参数参照文献[24]的 Transformer-base 模型的参数.在该参数设置中,编码器和解码器的层数均设置为 6,多头注意力机制的头数设置为 8,词向量大小和隐藏状态大小均设置为 512 维,同时前向神经网络中间层状态大小设置为 2048 维.本文模型中使用 Adam 优化器[40],并且设置β_1和β_2值分别为 0.9 和 0.98.学习率和预热步数(Warm up step)分别设置为 2.0 和 16000.以子词为单位的批处理大小设置为 4096.所有的模型皆训练 25 万次迭代,并且根据 news-commentray-v11 数据集的开发集选择最优模型.

在微调模型时,模型参数与如上的预训练模型参数设置一致.

4.3 评价方法

Table 3 AMR-to-Text performance on AMR2.0 and AMR3.0

表 3 AMR 文本生成在 AMR2.0 和 AMR3.0 测试集上的性能

#	预训练	微调	AMR2.0				AMR3.0			
			BLEU	Meteor	chrF++	BERTScore	BLEU	Meteor	chrF++	BERTScore
1	None	None	28.03	34.08	61.35	93.19	31.40	36.48	64.20	94.00
2	DAE(T)	Vanilla	31.38	35.84	63.62	93.80	32.15	36.71	65.04	94.03
3		MTL	33.59	36.98	65.51	94.28	34.58	37.71	66.61	94.61
4	DAE(S)	Vanilla	31.12	36.02	63.59	93.79	32.54	37.09	65.23	94.19
5		MTL	34.53	37.98	66.40	94.48	34.72	38.07	66.79	94.41
6	A2T	Vanilla	38.77	39.86	67.12	94.65	37.64	39.20	66.60	94.63
7	DAE(S) + DAE(T)	Vanilla	32.95	36.78	64.80	94.10	34.42	37.64	66.28	94.45
8		MTL	34.58	37.76	66.13	94.52	35.41	38.42	67.12	94.73
9	DAE(S) + A2T	Vanilla	39.63	40.37	68.48	94.97	38.17	39.59	66.86	94.74
10		MTL	39.82	40.34	68.51	95.03	39.23	40.20	67.12	**94.84**
11	DAE(T) + A2T	Vanilla	39.21	39.97	68.58	94.52	37.46	39.27	65.85	94.56
12		MTL	39.37	40.06	68.31	94.97	37.35	39.26	66.06	94.50
13	DAE(S) + DAE(T) + A2T	Vanilla	**40.35**	40.57	68.52	95.04	38.11	39.65	65.00	94.37
14		MTL	40.30	**40.66**	**68.82**	**95.11**	**38.97**	**40.10**	**67.07**	94.73

为评估生成文本的性能,本文使用多个评测指标,包括 BLEU[41]、Meteor[42]、chrF++[43]和 BERTScore[44].与前三者不同的是,BERTScore 的计算并不直接依赖于生成的文本和正确文本之间相同的词形,即通过 BERT 获

取自动生成的文本和正确文本的语义表示向量,然后再通过计算向量之间的相似度来获取两文本之间的 BERTScore 值,而不直接依赖于文本中的相同词和词串.这一点与 AMR 文本生成任务非常贴切,因为同一个 AMR 图可以表示为多个不同但意义相同的句子.此外,BLEU 是基于语料级的评测,而后三者是基于句子级的评测.

4.4 实验结果

表 3 给出了各预训练模型在 AMR2.0 和 AMR3.0 测试集上的性能.从实验结果可以看出:

- 基于单任务的预训练(#2~#6)显著提高了 AMR 文本生成的性能,这说明基于序列到序列的预训练模型有助于 AMR 文本生成.其中,A2T 的预训练,即基于大规模自动 AMR 文本分析语料,提升幅度最大,在 AMR2.0 上提高了 10.74 个 BLEU 值.难能可贵的是,虽然 DAE(S)和 DAE(T)这两个预训练任务与 AMR 文本生成任务并不直接相关,通过把已训练好的模型参数迁移到 AMR 文本生成模型来帮助后者训练,使得后者不用像大多数模型那样从零学习.
- 两个或更多任务上的联合预训练较单任务预训练,进一步提高了 AMR 文本生成的性能.例如,相比于 A2T 单任务,联合 DAE(S)和 DAE(T)之后,AMR2.0 上的性能 BLEU 值由 38.77 提高至 40.30.
- MTL 微调取得比 Vanilla 微调更好的性能.例如,基于 DAE(S)和 DAE(T)两个单任务,MTL 微调较之 Vanilla 微调,提高 2~3 个 BLEU 值.然而,基于两个或更多预训练任务,随着 AMR 文本生成的性能的进一步提高,MTL 微调的优势逐渐变得不明显.
- 由于 AMR3.0 较 AMR2.0 有更多的训练语料和测试数据,基准系统在 AMR3.0 上的性能明显高于 AMR2.0.但随着预训练模型的使用和性能的不断提升,AMR3.0 上的性能优势逐渐变得不明显.

4.5 与相关工作的比较

目前 AMR 文本分析的相关工作仍然聚焦于设计更优的图到序列(Graph2Seq)模型.作为最新工作的代表,Song 等人[26]在 Zhu 等人[10]的图模型基础上,进一步提出了从目标端构造线性化 AMR,在 AMR2.0 上取得了 34.13 的性能,远高于本文基准系统的性能.

表 4 比较了本文工作与相关工作的性能. 其中 Reconstructor 的性能指 Song 等人[26]的 Loss 2: Reconstructing Linearized 方法.我们使用 Song 等人[26]的开源代码,首先使用本文 0.39M 自动标注语料进行预训练,然后再使用 AMR2.0 语料进行微调后获取性能. Mager 等人[33]和 Harkous 等人[34]分别使用了大规模预训练模型 GPT-2[12]和 RoBERTa[45].从表中可以看出:

Table 4 Performance comparison of our approach and related studies

表 4 本文方法与相关工作的比较

	额外资源	BLEU	Meteor	chrF++	BERTScore	#参数
Our	0.39M	**40.30**	**40.66**	**68.82**	**95.11**	54M
Reconstructor[26]	0.39M	38.27	38.47	66.08	94.20	62M
Mager 等人[33]	GPT-2	33.02	37.68	63.89	-	762M+
Harkous 等人[34]	RoBERTa	35.6	37.3	-	-	355M+

- 在仅使用 AMR2.0 的人工标注数据下,基于图结构的 Reconstructor 方法虽然较本文基准系统取得更好的性能,然而,随着大规模自动标注语料的使用,本文基于多任务预训练的方法较 Reconstructor 方法提升 2.03 个 BLEU 值,可见复杂图模型在大规模语料情况下的优势变得不明显.同时,相比于本文的序列到序列基准模型,复杂图模型可能对自动标注语料的质量有着更高的要求.
- 相比于 Mager 等人[33]和 Harkous 等人[34],虽然本文预训练模型的数据规模远低于 GPT-2 和 RoBERTa 所使用的预训练数据,本文较两者分别提高了 7.28 和 4.7BLEU 值.这说明针对 AMR 文本生成任务本身,制定合适的预训练任务是有必要的.
- 在模型参数方面,本文使用的词表大小虽然是 Reconstructor 词表的两倍,但由于 Reconstructor 方法本

身较为复杂,本文方法模型参数要低于后者约 8M.此外,本文方法的模型参数要远低于基于大规模预训练模型的 Mager 等人[33]和 Harkous 等人[34]的方法.

5 分析与讨论

本节本文以 AMR2.0 为例,从多方面进一步分析预训练对 AMR 文本生成性能的影响.其中,A2T 预训练任务使用 Vanilla 微调方法, 而其他预训练任务均使用 MTL 微调方法.

5.1 预训练数据集大小对AMR文本分析性能的影响

从 0.39M 的预训练数据集中,随机抽取 20%、40%、60%和 80%作为预训练模型的数据集,然后再在预训练模型的基础上,使用微调方法训练 AMR 文本生成模型.图 4 给出了预训练数据集大小对 AMR 文本分析性能影响的折线图.从图 4 可以看出:

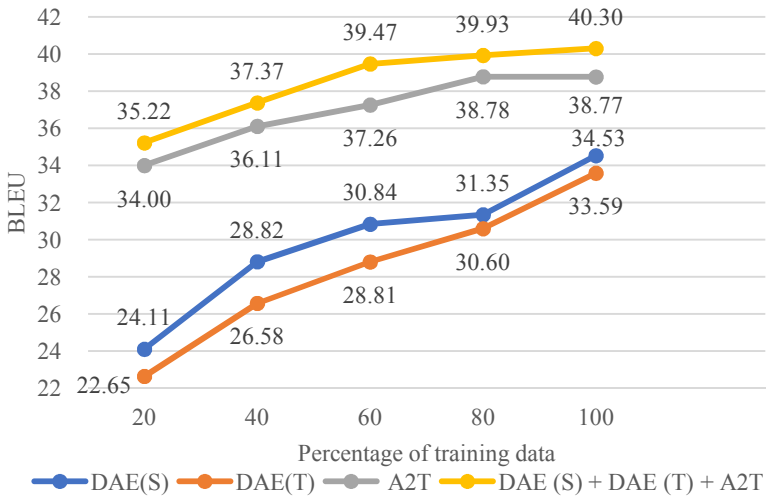


Fig. 4 Learning curve over the number of instances in pre-training datasets

图 4 预训练数据集大小对 AMR 文本分析性能的影响折线图

- 对于单预训练任务和多预训练任务,随着预训练数据集规模的扩大,AMR 文本生成的性能也逐渐提高.此外,在本文使用的预训练数据集基础上,如果继续扩大数据的规模,AMR 文本生成性能仍具有提升空间,特别是 DAE(S)和 DAE(T)这两个预训练任务.
- 基于各个不同规模的预训练集,模型的性能趋势是一致的,即基于三个联合预训练任务的 AMR 文本生成性能最佳,随后分别是 A2T、DAE(T)和 DAE(S)三个单预训练任务.
- 当预训练数据集规模较小时(如图中的 20%和 40%),DAE(S)和 DAE(T)两个预训练任务的 AMR 文本生成性能甚至低于基准系统的性能.这说明,当训练语料较小时,预训练模型容易产生过拟合现象,反而负面影响了后续任务.

5.2 AMR自动标注语料质量对AMR文本分析性能的影响

从第 4 节的实验结果可以看出,基于 AMR 自动标注语料的预训练能够大幅度提高 AMR 文本生成的性能(见表 3 中的#1 和#6).因此,本节分析在预训练语料中,AMR 自动标注语料质量对后续 AMR 文本生成性能的影响.为了获取质量更佳的 AMR 自动标注语料,本文使用类似融合大规模自动标注语料的 AMR 分析器,该分析器在 AMR2.0 测试集上的性能达到 81.40 Smatch F1,远高于本文第 4 节中使用的 AMR 分析器.

表 5 给出了在使用不同质量的 AMR 自动标注语料时,AMR 文本生成取得的性能.从中可以看出,得益于自

动标注语料质量的提升,基于本文预训练任务的 AMR 文本生成性能也得到了进一步的提升.例如,基于三个联合预训练任务,AMR 文本分析性能 BLEU 值由 40.30 提高至 42.22.

Table 5 Performance comparision of AMR-to-Text generation when using pre-training datasets of different qualities

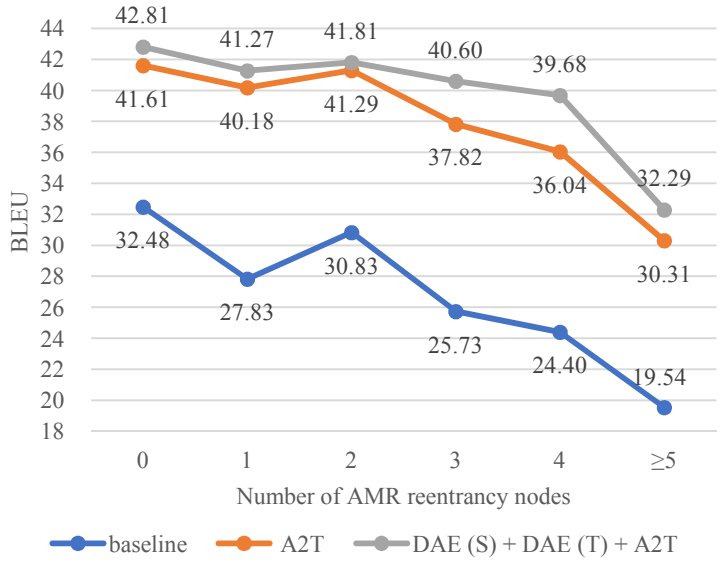
表 5 在使用不同质量 AMR 自动标注语料时,AMR 文本生成性能的比较

预训练	AMR 分析器	BLEU	Meteor	chrF++	BERTScore
A2T	73.21	38.77	39.86	67.12	94.65
	81.40	40.14	40.55	67.45	94.73
DAE(S) + DAE(T) +A2T	73.21	40.30	40.66	68.82	95.11
	81.40	**42.22**	**41.49**	**69.35**	**95.13**

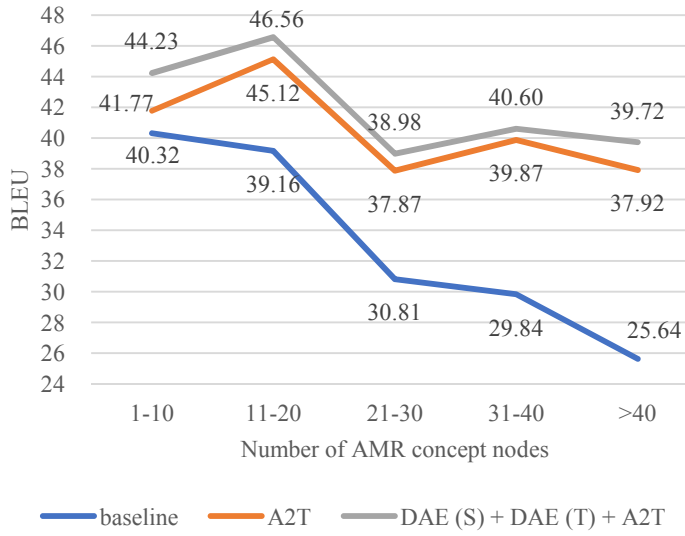
注:其中 73.21 和 81.40 分别指对应的 AMR 分析器在 AMR2.0 测试集上的性能 Smatch F1 值.

5.3 不同复杂度AMR的文本生成性能的影响

一般来讲,AMR 图越简单,其文本生成的性能越好,反之越差.为了进一步分析预训练模型对不同复杂度 AMR 的文本生成性能的影响,本文简单地分别以 AMR 结点数和 AMR 重度结点数的多少作为 AMR 复杂度的衡量.对测试集中的样例,根据 AMR 结点数和重度结点数进行分组,并评估各组的性能 BLEU 值.图 5 给出了不同复杂度 AMR 的文本生成性能.



(a)



(b)

Fig. 5 Performance (in BLEU) on the test set with respect to the reentrancy numbers (a) and node numbers (b).

图 5 (a) 不同 AMR 重度结点数的 AMR 文本生成性能; (b) 不同 AMR 结点数的 AMR 文本生成性能.

从图 5(a)中可以看出,随着 AMR 中重度结点数量的增加,三个系统的性能均呈现不同程度的下降.当重度结点数从 0 增加到 2 时,模型的性能虽有下降(除重度结点数 2 外),但幅度不大.而当重度结点数从 2 变为更多时,模型性能下降非常明显.相对而言,本文最优模型受重度结点数的影响要较另两个模型更加缓和.值得注意的是,即使是对于重度结点超过 5 的 AMR 图,本文最优模型得取的性能与基准模型在重度结点为 0 的 AMR 图上的性能相当(32.29 vs. 32.48).

从图 5(b)中可以看出,当 AMR 中结点数量在 20 以内时,基准系统的性能较为稳定.但当结点数量超过 10 后,模型的性能呈现断崖式地下降,且随着结点数量的增加,基准系统的性能持续下降.例如,当结点数超过 10 时,AMR 文本生成性能由 BLEU 值 40.32 下降到 25.64.相比于基准系统,A2T 与 DAE(S) + DAE(T) + A2T 系统的性能要稳定得多.例如,当 AMR 中结点数量大于 20 时,模型的性能趋于稳定,保持较高的水准.

经过以上对图 5 的分析可知,预训练模型的性能相比基准系统的性能有明显的提升,且可以在 AMR 结点数量较多时保持较为稳定的性能,但重度结点问题依旧是其面临的一道挑战.重度结点过多引起性能下降的主要原因有两方面,其一,由于本文仅通过复制概念结点解决 AMR 重度结点序列化问题,会令其损失部分的图结构信息; 其二,随着重度结点的增加,AMR 图结构会随之复杂,线性化会导致原本图结构位置相近的概念(如重度结点、兄弟概念结点)在线性化 AMR 中的距离变大,增加模型捕获图结构的难度.

5.4 案例分析

本节通过具体的案例来进一步对比各模型的性能,并且选取 Song 等人[26]基于 0.39M 自动标注语料的 Reconstructor 系统、本文基准系统以及本文使用外部语料的最优系统生成的样例进行分析,每个例子中包括 AMR 图,参考文本,Reconstructor 系统、本文基准系统以及最优系统的生成文本.

如表 6 所示,在例(1)中,基准系统错误地将语义角色":ARG2"解析为介词"from",同时错误地生成专有名词词组"nuclear nonproliferation group";Reconstructor 方法和本文方法生成的文本均在语义上与参考文本近乎一致.在例(2)中,Reconstructor 方法成功识别出了 AMR 图中的重度结点,但重复生成"no"和"nuclear",使得文本连贯性有所降低;基准系统同样有此问题,并且在生成的文本中还额外引入了其他词,如"fact"等;相比而言,本文方法生成的文本在语义上表达更清晰、语言表达更连贯.在例(3)中,Reconstructor 方法与基准系统均出现了不同

程度的实体生成错误,而本文方法能够较好地表达出 AMR 图的语义.总体而言,与基于图模型的 Reconstructor 系统和基准系统相比,本文方法生成的文本更为连贯,语义更为明确.

Table 6 Case studies on AMR-to-Text generation

表 6 AMR 文本生成案例分析

(1)	(d / distribute-01 :ARG1 (d2 / document :ARG0-of (o / outline-01 :ARG1 (p / propose-01 :ARG0 (c / country :wiki "Israel" :name (n / name :op1 "Israel"))))) :ARG2 (p2 / person :quant 45 :ARG0-of (h / have-org-role-91 :ARG1 (o2 / organization :wiki "Nuclear_Suppliers_Group" :name (n2 / name :op1 "Nuclear" :op2 "Suppliers" :op3 "Group")) :ARG2 (m / member))) :time (d3 / date-entity :year 2007 :month 3))	
	Reference	documents outlining the israeli proposal were distributed to the 45 - member nuclear suppliers group (nsg) in march 2007 .
	Reconstructor	in march 2007 a document outlining israel 's proposal was distributed to 45 members of the nuclear suppliers group .
	Baseline	in march 2007 an document outlining israel 's proposal was distributed from 45 members of the nuclear nonproliferation group .
	Our	documents outlining the israeli proposal were distributed to 45 members of the nuclear suppliers group in march 2007 .
(2)	(s / say-01 :ARG0 (a / accord-03) :ARG1 (r / remain-01 :ARG1 (a2 / and :op1 (i / issue-02 :polarity - :ARG0 p) :op2 (a3 / ambiguity :polarity -) :ARG0-of (r2 / regard-01 :ARG1 (a4 / and :op1 (p / program :mod (n / nucleus) :poss (c2 / country :wiki "Iran" :name (n2 / name :op1 "Iran"))) :op2 (a5 / activity-06 :ARG0 c2 :mod n))))))	
	Reference	the accord said that there were no remaining issues and ambiguities regarding iran 's nuclear program and activities .
	Reconstructor	according to the accord , there remains no issue and no ambiguities regarding iran 's nuclear program and nuclear activities .
	Baseline	the accord said it remains a fact that the program did not issue and is not ambiguity regarding iran 's nuclear program and iran 's nuclear activities .
	Our	according to the accord , there remains no issue and ambiguity regarding iran 's nuclear program and activities .
(3)	(t / tell-01 :ARG0 (n / newspaper :wiki "Richmond_Times-Dispatch" :name (r / name :op1 "Richmond" :op2 "Times-Dispatch")) :ARG1 (t2 / tale :topic (i / impact-01 :ARG1 (r2 / road :wiki - :name (h / name :op1 "Huguenot" :op2 "Trail") :location (c / county :wiki "Powhatan_County,_Virginia" :name (p / name :op1 "Powhatan" :op2 "County"))))) :time (t3 / today))	

Reference	the richmond times - dispatch tells the tale today about the impact on the huguenot trail in powhatan county .
Reconstructor	today , the riche timeline tells a tale of the impact on the huffington trail in politburo county .
Baseline	richmond times-dispatch newspaper today told that the tale about the impact of huguenot trail at polt an county .
Our	today , the richmond times-dispatch tells a tale about the impact of the huguenot trail in powhatan county .

6 总结与未来工作

本文首次将序列到序列预训练任务引入到 AMR 文本生成任务中,提出了三种简单有效的预训练任务和两种微调方法.实验结果表明,本文的方法能够有效地提升 AMR 文本生成的性能,并且在两份 AMR 数据集中均达到了目前最优结果,其中当使用质量较高的自动标注语料时,实验中在 AMR2.0 中的最高结果达到了 42.22.

在使用预训练模型后,本文模型在有较多结点的 AMR 图中依旧可以达到较高的性能,但在重度结点的数量较多时,本文模型还有很大的提升空间.在未来的工作中,将尝试解决线性化 AMR 导致重度结点结构信息丢失的问题.

References:

[1] Banarescu L, Bonial C, Cai S, et al. Abstract Meaning Representation for Sembanking. Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse. 2013: 178-186.

[2] Tamchyna A, Quirk C, Galley M. A discriminative model for semantics-to-string translation. Proceedings of the 1st Workshop on Semantics-Driven Statistical Machine Translation. 2015: 30-36.

[3] Mitra A, Baral C. Addressing a question answering challenge by combining statistical methods with inductive rule learning and reasoning. Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. 2016: 2779-2785.

[4] Li X, Nguyen T H, Cao K, et al. Improving event detection with abstract meaning representation. Proceedings of the first workshop on computing news storylines. 2015: 11-15.

[5] Xu K, Wu L, Wang Z, et al. Graph2seq: Graph to sequence learning with attention-based neural networks. arXiv preprint arXiv:1804.00823, 2018.

[6] Flanigan J, Dyer C, Smith N A, et al. Generation from abstract meaning representation using tree transducers. Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2016: 731-739.

[7] Song L, Peng X, Zhang Y, et al. AMR-to-Text Generation with Synchronous Node Replacement Grammar. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). 2017: 7-13.

[8] Pourdamghani N, Knight K, Hermjakob U. Generating English from Abstract Meaning Representations. Proceedings of the 9th International Natural Language Generation conference. 2016: 21.

[9] Ferreira T C, Calixto I, Wubben S, et al. Linguistic realisation as machine translation: Comparing different MT models for AMR-to-Text generation. Proceedings of the 10th International Conference on Natural Language Generation. 2017: 1-10.

[10] Zhu J, Li J, Zhu M, et al. Modeling Graph Structure in Transformer for Better AMR-to-Text Generation. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. 2019: 5462-5471.

[11] Peters M, Neumann M, Iyyer M, et al. Deep Contextualized Word Representations. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). 2018: 2227-2237.

[12] Radford A, Wu J, Child R, et al. Language models are unsupervised multitask learners. OpenAI Blog, 2019, 1(8): 9.

[13] Devlin J, Chang M W, Lee K, et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). 2019: 4171-4186.

[14] Li Z, Hoiem D. Learning without forgetting. IEEE transactions on pattern analysis and machine intelligence, 2017, 40(12): 2935-2947.

[15] Konstas I, Iyer S, Yatskar M, et al. Neural AMR: Sequence-to-Sequence Models for Parsing and Generation. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2017: 146-157.

[16] Cao K, Clark S. Factorising AMR generation through syntax. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). 2019: 2157-2163.

[17] Marcheggiani D, Perez-Beltrachini L. Deep Graph Convolutional Encoders for Structured Data to Text Generation. Proceedings of the 11th International Conference on Natural Language Generation. 2018: 1-9.

[18] Song L, Zhang Y, Wang Z, et al. A Graph-to-Sequence Model for AMR-to-Text Generation. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2018: 1616-1626.

[19] Beck D, Haffari G, Cohn T. Graph-to-Sequence Learning using Gated Graph Neural Networks. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2018: 273-283.

[20] Damonte M, Cohen S B. Structural Neural Encoders for AMR-to-Text Generation. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). 2019: 3649-3658.

[21] Guo Z, Zhang Y, Teng Z, et al. Densely Connected Graph Convolutional Networks for Graph-to-Sequence Learning. Transactions of the Association for Computational Linguistics, 2019, 7: 297-312.

[22] Ribeiro L F R, Gardent C, Gurevych I. Enhancing AMR-to-Text Generation with Dual Graph Representations . Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. 2019: 3183–3194

[23] Cai D, Lam W. Graph Transformer for Graph-to-Sequence Learning . Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence. 2020: 7464-7471.

[24] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017: 6000-6010.

[25] Zhao Y, Chen L, Chen Z, et al. Line Graph Enhanced AMR-to-Text Generation with Mix-Order Graph Attention Networks. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020: 732-741.

[26] Song L, Wang A, Su J, et al. Structural Information Preserving for Graph-to-Text Generation. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020: 7987-7998.

[27] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality. Proceedings of the 26th International Conference on Neural Information Processing Systems-Volume 2. 2013: 3111-3119.

[28] Pennington J, Socher R, Manning C D. Glove: Global vectors for word representation. Proceedings of the 2014 conference on empirical methods in natural language processing. 2014: 1532-1543.

[29] McCann B, Bradbury J, Xiong C, et al. Learned in translation: contextualized word vectors. Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017: 6297-6308.

[30] Wang L, Zhao W, Jia R, et al. Denoising based Sequence-to-Sequence Pre-training for Text Generation. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. 2019: 3994-4006.

[31] Song K, Tan X, Qin T, et al. MASS: Masked Sequence to Sequence Pre-training for Language Generation. International Conference on Machine Learning. 2019: 5926-5936.

[32] Lewis M, Liu Y, Goyal N, et al. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension . Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020: 7871–7880.

[33] Mager M, Astudillo R F, Naseem T, et al. GPT-too: A language-model-first approach for AMR-to-Text generation . Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020: 1846–1852

[34] Harkous H, Groves I, Saffari A. Have Your Text and Use It Too! End-to-End Neural Data-to-Text Generation with Semantic Fidelity. arXiv preprint arXiv:2004.06577, 2020.

[35] Sennrich R, Haddow B, Birch A. Neural Machine Translation of Rare Words with Subword Units. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2016: 1715-1725.

[36] Hu J, Xia M, Neubig G, et al. Domain Adaptation of Neural Machine Translation by Lexicon Induction. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019: 2989-3001.

[37] Vincent P, Larochelle H, Lajoie I, et al. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of machine learning research, 2010, 11(12).

[38] Ge D, Li J, Zhu M, et al. Modeling source syntax and semantics for neural AMR parsing. Proceedings of the 28th International Joint Conference on Artificial Intelligence. AAAI Press, 2019: 4975-4981.

[39] Johnson M, Schuster M, Le Q, et al. Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. Transactions of the Association for Computational Linguistics, 2017, 5: 339-351.

[40] Kingma D P, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

[41] Papineni K, Roukos S, Ward T, et al. BLEU: a method for automatic evaluation of machine translation. Proceedings of the 40th annual meeting of the Association for Computational Linguistics. 2002: 311-318.

[42] Banerjee S, Lavie A. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, 2005: 65.

[43] Popović M. chrF++: words helping character n-grams. Proceedings of the Second Conference on Machine Translation. 2017: 612-618.

[44] Zhang T, Kishore V, Wu F, et al. BERTScore: Evaluating Text Generation with BERT. International Conference on Learning Representations. 2020.

[45] Liu Y, Ott M, Goyal N, et al. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692, 2019.