

# 基于序列到序列的中文AMR解析

黄子怡, 李军辉, 贡正仙

(苏州大学计算机科学与技术学院, 江苏苏州 215006)

20195227032@stu.suda.edu.cn, {lijunhui, zhxgong}@suda.edu.cn

## 摘要

抽象语义表示(Abstract Meaning Representation, 简称AMR)是将给定的文本的语义特征抽象成一个单根的有向无环图。AMR语义解析则是根据输入的文本获取对应的AMR图。相比于英文AMR, 中文AMR的研究起步较晚, 造成针对中文的AMR语义解析相关研究较少。本文针对公开的中文AMR语料库CAMR1.0, 采用序列到序列的方法进行中文AMR语义解析的相关研究。具体地, 首先基于Transformer模型实现一个适用于中文的序列到序列AMR语义解析系统; 然后, 探索并比较了不同预训练模型在中文AMR语义解析中的应用。基于该语料, 本文中文AMR语义解析方法最优性能达到了70.29的Smatch F1值。本文是第一次在该数据集上报告实验结果。

**关键词:** 中文AMR语义解析; 序列到序列模型; 预训练模型

## Chinese AMR Parsing based on Sequence-to-Sequence Modeling

HUANG Ziyi, LI Junhui, GONG Zhengxian

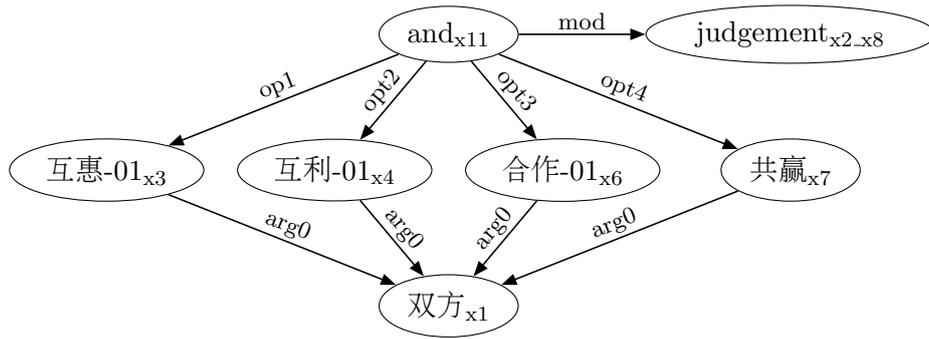
(School of Computer Science and Technology, Soochow University, Suzhou, Jiangsu 215006)

20195227032@stu.suda.edu.cn, {lijunhui, zhxgong}@suda.edu.cn

## Abstract

Abstract Meaning Representation (AMR) abstracts the semantic features of a given sentence into a single-rooted directed acyclic graph. AMR Parsing aims to transfer an input sentence into its corresponding AMR graph. Compared with English AMR, research on Chinese AMR is few. In this paper, we take advantage of the published Chinese AMR corpus CAMR 1.0 and formalize Chinese AMR parsing as a sequence-to-sequence learning. Specifically, we first apply the state-of-the-art seq2seq model, i.e., Transformer to Chinese AMR parsing. Then, we explore and compare various pretrained models to boost the performance of Chinese AMR parsing. Experimental results on CAMR 1.0 show that we achieves 70.29 Smatch F1 score. To our best knowledge, this is the first result reported on this dataset.

**Keywords:** Chinese AMR parsing, seq2seq modeling, pre-trained model



双方<sub>x1</sub> 是<sub>x2</sub> 互惠<sub>x3</sub> 互利<sub>x4</sub>、<sub>x5</sub> 合作<sub>x6</sub> 共赢<sub>x7</sub> 的<sub>x8</sub>。 <sub>x9</sub>

Figure 1: CAMR1.0中的一个中文AMR图示例。  
节点中“x+数字”代表着该节点所对应词在文本中所处的位置，  
如：“judgement<sub>x2\_x8</sub>”表示该节点对应“是...的”。

## 1 引言

抽象语义表示(Abstract Meaning Representation, AMR), 如图 1所示, 是一种新兴的语义解析表示形式, 旨在从句子中抽象出输入文本的语义特性, 利用图结构呈现句子的结构化语义信息。AMR将句子语义结构表示为一个单根的有向无环图, 并使用概念节点(concept)表示句子中的实体、属性、事件、状态等信息, 概念节点之间的边标签表示对应的语义关系(Banarescu et al., 2013; Zhou et al., 2016)。不同于其他语义标注体系的做法, 如语义角色标注只关注句子局部成分的含义、只对部分语义进行注释, AMR表示将词类(动词和名词)、词序、形态变异等句法概念抽离, 忽视文本的语态、时态等信息, 而侧重于强调整句自身的表意, 关注与语义密切相关的成分。近年来, AMR被成功应用于许多任务及场景中。如:Liao et al. (2018)将AMR语义解析应用于文本摘要中, 将一组相似的句子作为输入, 利用开源的AMR解析工具每个句子映射到一个AMR图中, 接着将所有AMR图合并到一个连接的源图中, 再通过结构化预测从源图中提取摘要图, 最后从中生成符合自然语言习惯的文本摘要; Song et al. (2019b)使用基于注意力的长短期记忆神经网络(Att-BiLSTM)和图递归网络(GRN)AMR编码器的模型对文本序列以及对应的AMR图进行编码, 将句子语义信息融入机器翻译的过程中; Bonial et al. (2019)对AMR标注方法进行了改进, 加入情景人机对话所需的时态、语态等信息, 并标注了适用于人机对话的AMR语料库, 为日后将AMR使用在人机对话中提供了可能性。

自Banarescu et al. (2013)于2013年公布了面向英语的AMR的标注规范及英语AMR语料库以来, 随后几年中又有学者陆续发布了其他语言的AMR语料库, 如巴西语(Anchieta and Pardo, 2018)、汉语(Li et al., 2016; Li et al., 2017)、韩语(Choe et al., 2019)等。鉴于英语AMR语料相较于其他语言的AMR语料更为成熟, 准确性更高, 大多数学者都是在英文AMR语料库上进行AMR语义解析的研究。但随着跨语言AMR语料库的丰富, 面向其他语言如中文的AMR解析任务开始得到大家的关注(Wang et al., 2018; 吴泰中et al., 2019)。

Wang et al. (2018)首先尝试对中文AMR进行语义解析, 利用基于转移的方法取得58.7的Smatch F1值。吴泰中et al. (2019)在此基础上进行了一系列的改进, 如使用Word2Vec预训练词向量、对模型中长短期记忆神经网络进行语言模型预训练以及利用依存句法来优化语义表示等, 让中文AMR解析性能得到进一步的提升, 最终达到了61的Smatch F1值, 是目前中文AMR语义解析的最好结果。但基于转移的模型有着潜在的问题: 该方法需要通过一系列的动作(action)来构建AMR图, 这使得研究者需要人为地为AMR解析设计复杂的动作, 并且为了实现该方法, 在编程时需要利用栈来对节点进行复杂的移进移出操作。相比之下, 序列到序列模型(Sequence-to-Sequence, Seq2Seq)无需人为地设计具体动作, 只需要将文本和对应AMR图交由模型学习, 模型可自行捕获两者之间的内在联系。

受启发于Ge et al. (2019)和Xu et al. (2020)在英语AMR解析的工作: 即便将AMR中的

结构信息线性化，模型依旧可以获得较好的AMR语义解析性能。同时Bevilacqua et al. (2021)使用Seq2Seq模型进行AMR语义解析，达到了目前在英语AMR语义解析中的最好结果84.5的F1值。因此，本文将探索Seq2Seq模型在中文AMR语义解析上的应用，同时为了缓解人工标注数据受限问题，还将探索并比较多种预训练在该问题上的应用。基于中文AMR语料库CAMR1.0的实验结果表明，通过融合中文预训练模型获取文本的上下文表示，基于Seq2Seq模型的中文AMR解析可以达到70.27的F1值。特别地，该方法将结构化的AMR图线性化，简化了AMR语义解析的过程，只需进行端到端的训练，省略了吴泰中et al. (2019)工作中的复杂模块。

本文的贡献包括：

- 探索Seq2Seq模型在中文AMR语义解析的应用，并分析不同子词化方法对中文AMR语义解析的影响；
- 探索并比较了多种中文预训练模型在中文AMR解析任务上性能的差异；
- 首次在公开的中文AMR数据集CAMR1.0(LDC2019T07)上实验并报告结果。

## 2 相关工作

### 2.1 AMR与中文AMR

目前常用的语义表示框架有基本依赖结构(Elementary Dependency Structures,EDS)、普遍概念认知注释(Universal Conceptual Cognitive Annotation,UCCA)、抽象语义表示(Abstract Meaning Representation, AMR)等，每种框架都使用图结构来表达文本的语义特征，其中EDS框架中图节点需要与句子中成分一一对应；UCCA框架中图节点与句子的成分存在部分对应关系；而在AMR中图节点和句子的成分之间没有显式的对应关系。AMR将句子语义表示为一个单根有向无环图，图中的概念节点表示句子中的语义概念，两节点之间的边表示两节点概念之间对应的语义关系。相比较其他语义表示方法，AMR具有以下的特点：

- AMR中的概念节点与句子成分无需一一对应，因此节省了在开始任何语义注释之前首先构建形态语法结构所需的时间；
- AMR的标注方法并不直接将句子成分作为节点概念，而是关注句子中对语义起作用的单词，因此句子中的虚词等语法上的功能词并不会被表示出来，同时也有一些概念节点并不应对句子中任何一个单词而是从上下文推断出来的；
- AMR避开了词序和形态句法标记等表面句法结构元素，而这些元素是不同语系及语言表达差异的主要来源，因此AMR使语义注释框架在不同语言间具有更强的可移植性。

Banarescu et al. (2013)首先提出了英文AMR的标注规则，其中包含了约100个语义关系，包含语义角色、一般语义关系、数量关系、日期实体(data-entities)关系、列表关系等。AMR采用从PropBank中提取的谓词框架词库，CAMR1.0在此基础上加入了Chinese Propbank框架(Song et al., 2019a)。Xue et al. (2014)是最早进行中文AMR相关研究的学者，他们利用AMR中不包含词序、词法和虚词等功能词，对不同语言语义影响较小的优势，以此探究英语、汉语和捷克语这三种语言之间的差异性和兼容性。最终结论表明相对于捷克语-英语而言，汉语-英语之间拥有更高的兼容性。但是将英语AMR标注体系迁移到中文语境并非易事，汉语具有特有语言现象，如动补结构、量词、重叠词、离合词、省略等，因此对中文的AMR标注是极其耗费人力和资源的一件事。Li et al. (2016)以《小王子》的中文翻译为例，对中文AMR语料库的标注规范进行了研究，提出了一个更适用于中文的AMR标注方法。随后Li et al. (2017)进一步对中文AMR标注规范进行改进，除了继承英文标注规范中的概念对齐外，也将AMR中关系与文本进行了对齐。CAMR1.0与AMR标注体系的主要区别之一对于时序和复合句之间关系的标注方法(李斌et al., 2017)，在AMR标注体系中对仅仅提供了少量抽象概念(如and, or, contrast.01)和语义关系(如:cause, :condition, :concession, :purpose)用于对复合句的标注，并通过这些概念和语义关系的组合来表达不同复合句之间的语句关系。但是这对于复杂的中文而言，这些是远远不够的，因此Li et al. (2019)设置了10个概念来表示句子之间的关系，使中文AMR标注体系能够更好地适用于复杂地中文语言环境。

## 2.2 AMR语义解析及中文AMR语义解析

目前常用的AMR语义解析的方法大致可以分为以下三种：基于转移的方法(transition-based parsing)、基于图的方法(graph-based parsing)以及基于序列到序列的方法(Seq2Seq-based parsing)。基于转移的方法在早期工作比较常见，通过人工规定一系列动作(action),从缓冲区中读取单词并逐步将它们组合成堆栈中的语法结构的一系列决策。基于图的方法在维持AMR的图结构基础上进行概念识别和关系预测。最后一种方法是基于序列到序列的方法，通过去除AMR图中的结构信息，将AMR图线性化为AMR序列，使用序列到序列的模型进行AMR语义解析，常见的被用于AMR语义解析的序列到序列模型有Transformer(Ge et al., 2019; Xu et al., 2020; Xu et al., 2021)、BART(Bevilacqua et al., 2021)等。使用序列到序列的模型最大的优点就是简化了训练过程，减少AMR语义解析预处理和训练过程中人工的参与。Barzdins and Gosko (2016)首次利用序列到序列模型进行AMR语义解析任务。他们在预处理过程中，通过复制AMR中共用的概念节点、删除wiki链接和变量，获得了简化的AMR序列，对比原始AMR图和通过后处理恢复被删除成分的AMR图，发现这种方法获得的简化AMR序列与原始AMR的差别并不大，极大简化了输出序列的长度。Ge et al. (2019)首先使用Transformer进行AMR语义解析，并在此基础上通过在源端序列中以一定规则加入句法和语义信息，帮助模型更好的学习句子语义信息；Xu et al. (2020)尝试利用大规模外部语料获得预训练模型，进一步提升AMR性能；Bevilacqua et al. (2021)首先使用编码器-解码器的预训练模型进行AMR语义解析，将使用不同遍历方法:广度优先遍历(BFS)和深度优先遍历(DFS)获得的序列用于AMR语义解析中，得出结论使用DFS获得序列更适用于AMR语义解析，并得到了目前在英语AMR语义解析中最优性能84.5的F1值。

相较于英文AMR语义解析，中文AMR语义解析相关研究起步较晚，Wang et al. (2015)在英语AMR语料上进行基于转移的AMR语义解析，设计了一序列具有语言直觉性的动作，借助句子依赖结构与AMR之间映射的规律，将Text2AMR任务转变为Text2Dependency、Dependency2AMR两个过程。Wang et al. (2018)又将上述方法首次针对未公开的中文AMR语料进行AMR语义解析，在基于转移的模型中，中文AMR语义解析获得了58.7的Smatch F1值。此后吴泰中et al. (2019)在此基础上通过对依存句法建模，加入了词路径和关系路径，并引入了预训练词向量Word2Vec，并通过预训练语言模型获取中文词的ELM(embeddings from language model)，使基于转移的中文AMR解析最优性能达到61的Smatch F1值。

## 3 面向中文的AMR解析

给定一个句子 $x$ ，AMR解析需要找到一个与其相匹配的AMR表示 $y$ ，如1所示。本文使用Transformer作为中文AMR语义解析基线模型，端到端地进行 $x \rightarrow y$ 的转换。但由于Seq2Seq模型无法对AMR进行结构化建模，我们需要首先通过数据预处理将AMR图序列化为易于模型处理的形式。当得到模型输出后，再通过后处理来修正输出中的结构化错误、删除重复节点，将模型输出还原为符合标准的AMR图。

### 3.1 序列到序列模型

Transformer依赖self-attention机制来计算输入和输出表示。对于RNN和CNN中比较明显的短板，如无法实现并行计算、计算冗余而低效、进行顺序计算时信息会丢失导致RNN缺乏长距离依赖等问题，Transformer都可以得到较好的解决。Transformer本质上是一个由编码器和解码器组成的端到端结构，可以实现原始数据到目标任务结果的直接输出。给定包含 $n$ 个单词的输入 $x = x_1, x_2, \dots, x_n$ ，首先通过查表获得每个输入 $x_i$ 所对应的向量表示 $\mathbf{e}_i^x$ ：

$$\mathbf{e}_i^x = \text{Embedding}(x_i) \quad (1)$$

随后，将表示向量序列 $\mathbf{e}_1^x, \mathbf{e}_2^x, \dots, \mathbf{e}_n^x$ 输入编码器中，获得每个单词的上下文相关表示 $\mathbf{h}_i$ ：

$$\mathbf{h}_i = \text{Encoder}([\mathbf{e}_1^x, \mathbf{e}_2^x, \dots, \mathbf{e}_n^x], i) \quad (2)$$

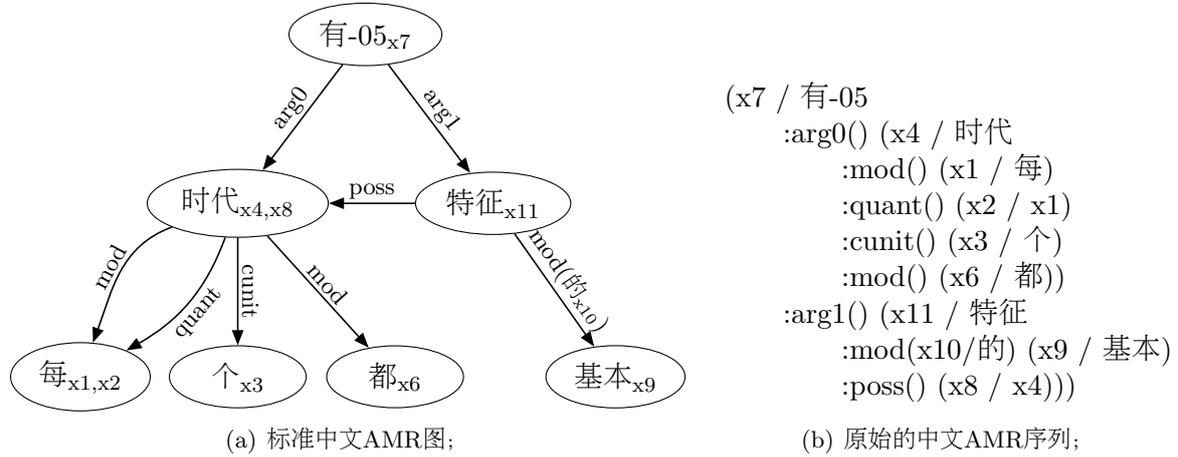
这些单词的表示捕获了输入文本 $x$ 中所蕴含的语义信息。我们将它们输入解码器中，得到输出AMR序列。其中，第 $j$ 个AMR序列的输出表示 $\mathbf{o}_j$ 为：

$$\mathbf{o}_j = \text{Decoder}([\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_n], [\mathbf{e}_1^y, \mathbf{e}_2^y, \dots, \mathbf{e}_{j-1}^y], j) \quad (3)$$

其中 $e_k^y$ 表示AMR序列 $y$ 中第 $k$ 个位置的符号的向量表示。最后通过一个全连接层获得模型在第 $j$ 位置所预测的AMR单词输出概率。

$$p(y_j|x, y_1, y_2, \dots, y_{j-1}) = \text{softmax}(\text{Linear}(\mathbf{o}_j)) \quad (4)$$

有关在Transformer中运用到的注意力(Attention)机制及其他更多细节，可以参阅论文(Vaswani et al., 2017)。



(有-05 :arg0 (时代:mod (每) :quant (每) :cunit (个) :mod (都)) :arg1 (特征:mod (基本) :poss (时代)))

(c) 线性化得到的中文AMR序列;

每<sub>x1</sub> 一<sub>x2</sub> 个<sub>x3</sub> 时代<sub>x4</sub> , <sub>x5</sub> 都<sub>x6</sub> 有<sub>x7</sub> 其<sub>x8</sub> 基本<sub>x9</sub> 的<sub>x10</sub> 特征<sub>x11</sub> 。 <sub>x12</sub>

(d) 中文源端句子;

Figure 2: 一个对CAMR进行数据预处理的示例。

### 3.2 数据预处理

我们借鉴Zhu et al. (2019)和Xu et al. (2020)在英文AMR解析中的预处理方式对中文AMR图进行预处理，如删除AMR图中大量的不影响AMR的语义的而只是用于指示共同引用节点的变量，在不改变句子语义的情况下获得线性化的AMR图。虽然CAMR1.0大体上遵循英文AMR的标注框架，但在规范在一些必要的地方针对中文特有的现象进行了调整，如复合句的处理、叠词的处理等。此外中文AMR图还对源端句子中出现的单词进行了关系对齐，如对于中文句子“每一个时代，都有其基本的特征。”其中“的”并不属于语义概念的范畴，但在中文AMR图中同样对这样的词进行了对齐，即对“基本”与“特征”的关系进行标注时也会将“的”与关系标签进行对齐，即图 2(a)中“特征<sub>x11</sub>  $\xrightarrow{\text{mod(的x10)}} \text{基本}_{x9}$ ”。但是它并不能提供更多的语义概念，因此我们在数据处理时删除了这些关系对齐。图 2给出了中文源端句子、对应的标准中文AMR图以及先行处理之后的结果的例子。

### 3.3 数据后处理

模型生成的AMR序列是一个不包含共用节点、wiki变量、包含冗余和重复信息的线性化AMR序列，需要通过一系列的后处理来恢复AMR图，如图 2(c)。针对英文线性化AMR的后处理包括恢复共用节点和wiki变量，修复模型生成的不完整的概念等。但是由于中文AMR并不包含wiki链接，因此仅需恢复变量以及共用节点。为方便起见，本文在van Noord and Bos (2017)提供的预处理和后处理脚本<sup>0</sup>的基础上针对中文的特点进行修改，如添加删除中文AMR中的关系对齐等功能。

<sup>0</sup><https://github.com/RikVN/AMR>

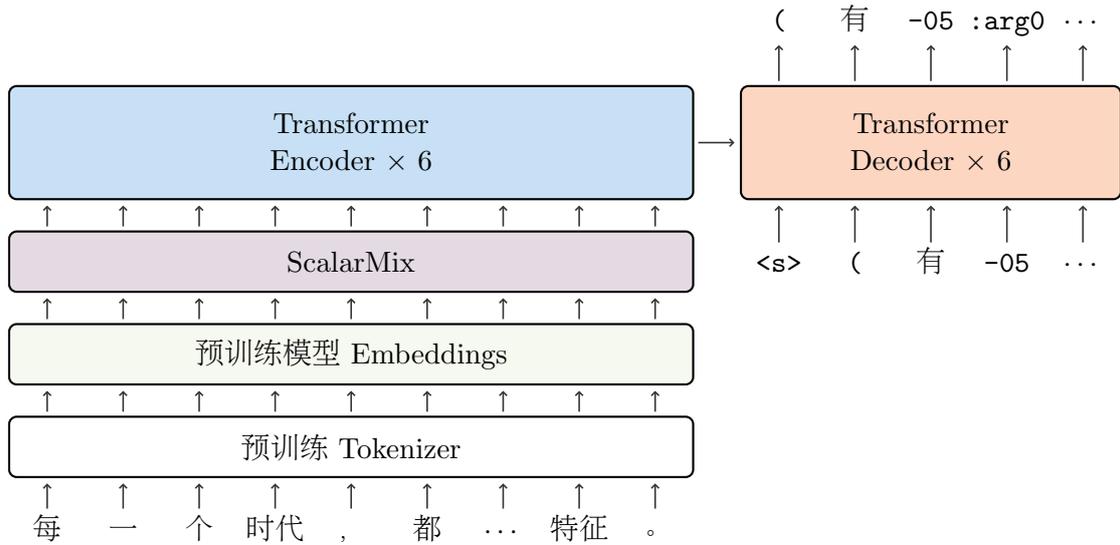


Figure 3: 序列到序列中文AMR解析模型结构。

## 4 结合中文预训练模型的中文AMR语义解析

### 4.1 预训练模型的使用

预训练模型的使用方法有很多，如使用预训练模型对输入序列提取语言特征，用预训练模型部分替换训练模型，将预训练模型作为特征提取器为模型增加额外的序列特征等。Xu et al. (2020)在英文数据集中对以上几种方式进行了比较，结果发现使用预训练模型获得文本对应的词向量可以获得最好AMR语义解析性能，因此本文也采用相同的方法。如图 3所示，本文获取预训练模型各层的输出向量，并将他们加权求和后得到的上下文相关表示作为后续编码器的输入。给定输入文本 $x$ ，利用预训练Tokenizer重新切分后的得到输入 $t = t_1, t_2, \dots, t_n$ 。首先通过预训练模型获得每个输入 $t_i$ 所对应的 $K$ 层上下文表示 $\mathbf{G}_i = [\mathbf{g}_i^0, \mathbf{g}_i^1, \mathbf{g}_i^2, \dots, \mathbf{g}_i^K]$ ：

$$\mathbf{G}_i = \text{PretrainedModel}(t, i) \quad (5)$$

随后将得到的 $K$ 层表示向量 $\mathbf{g}_i^0, \mathbf{g}_i^1, \mathbf{g}_i^2, \dots, \mathbf{g}_i^K$ 进行加权求和获得对应的上下文相关表示，并将输出 $\mathbf{e}_i$ 作为公式 2的输入。

$$\mathbf{e}_i = \text{ScalarMix}(\mathbf{G}_i) \quad (6)$$

其中Scalar mix通过两个可学习的参数 $\omega \in \mathbb{R}^k$ 和 $\gamma \in \mathbb{R}$ 来实现对预训练模型的输出加权求和。其中 $\omega$ 用来确定每一层输出的权重， $\gamma$ 则用来调整最终输出向量的方差，具体计算公式如下：

$$\mathbf{s} = \text{softmax}(\omega) \quad (7)$$

$$\text{ScalarMix}(\mathbf{G}) = \gamma \sum_{k=0}^K \mathbf{s}^k \mathbf{g}^k \quad (8)$$

### 4.2 中文预训练模型

目前许多预训练框架如BERT、XLnet、NEZHA都提供了在中文文本上进行预训练的模型。大多数预训练模型都是以BERT为基础从模型结构(如位置编码方式)、训练策略以及掩码方式(Mask)这几个角度进行优化衍生而来的。如传统位置编码方式可以分为：绝对位置编码和相对位置编码，绝对位置编码对单词在句子中的位置生成位置向量，相对位置编码则以当先单词为基准计算上下文的相对位置。而将位置编码表示融入模型的方法又分为：函数式和参数式，前者利用函数生成每个位置的单词的位置编码，后者利用参数让模型自动学习位置的嵌入向量，典型的例子就是原始的Transformer模型和BERT分别采用了函数式和

预训练模型	掩码方法	位置编码方式	训练策略
BERT	字级掩码	参数式绝对位置	MLM & NSP
BERT-wwm	全词掩码	参数式绝对位置	MLM & NSP
NEZHA	全词掩码	函数式相对位置	混合精度训练 & LAMB
XLNet	无	函数式相对位置	广义自回归训练

Table 1: 多种不同预训练模型的对比

参数式的绝对位置编码方式。以及使用不同的掩码策略对训练数据进行处理。下面我们将对BERT、NEZHA、XLNet进行简要介绍和横向对比，并在表 1中提炼了各个预训练模型改进之处。最后我们还介绍一个利用额外的中英翻译对模型进行预训练的方法。

**BERT预训练模型** ELMo模型是有两个方向相反的LSTM网络组合而成单向的语言模型进行预训练，而OpenAI的GPT采用单向的上下文感知结构，BERT(Devlin et al., 2018)是第一个对上下文双向表征的预训练模型，采用词向量、位置编码、句子编码相加之后的向量作为模型的输入,利用Transformer的编码器来学习文本中序列中单词之间的上下文关系。BERT涉及两个预训练任务，一是MLM，将单词序列中15%的单词被替换为[MASK]标记输入BERT模型，然后模型将根据序列中其他未屏蔽单词提供的上下文，尝试预测被屏蔽单词。在中文预训练中原始的BERT的上述掩码操作是以字为单位进行的。为了强化预训练模型对中文词的理解，Cui et al. (2019)将前人在英文预训练模型中使用的全词掩码(Whole Word Mask)运用到中文预训练中，即以词为单位对输入进行连续的掩码操作，并发布了新的中文预训练模型(BERT-wwm)。二是NSP模型接收成对的句子作为输入，并学习预测成对的第二句话是否是原始文档中的后续句子。BERT作为语言表征模型可以为下游任务提供更好的语言表征。

**NEZHA预训练模型** 该模型(Wei et al., 2019)基于BERT，以汉语文本为训练对象,在编码方式、训练策略、掩码方法、优化器都针对BERT进行了改进。NEZHA中使用了完全函数式的相对位置编码，没有任何需要学习的参数,并且通过剥离实验证明该方法相比于其他的位置编码的有效性。此外NEZHA预训练模型同样使用了全词掩码来代替BERT中的随机字级掩码操作，通过对掩盖原句中更多内容来提升任务难度使模型学到更多语义信息。此外预训练模型的参数通常为单精度浮点数(FP32)，但是NEZHA采用混合精度进行训练，在向前和向后传播过程中会将权重、梯度、激活函数转换为半单精度浮点数(FP16),但在参数更新过程中恢复为单精度浮点数(FP32)，此举可以有效提高预训练速度，节约内存空间，从而使用更大的批次大小进行训练，并使用LAMB优化器。

**XLNet预训练模型** 在对BERT进行预训练时，需要将一定数量的输入进行掩码操作。而这些掩码标注在真实文本或微调任务中是不存在的，这样一来BERT模型的预训练与运用场景之间的输入分布存在着不一致，而这有可能会损害模型的性能。而利用自回归的预训练模型如GPT系列，虽然预训练与运用场景之间的输入分布是相同的，但都只能利用单向的上下文信息。为此Yang et al. (2019)提出利用广义自回归(Generalized Autoregressive)的方式来对模型进行训练。不同于原始的自回归方法只关注从左到右或从右到左的概率分解。XLNet对序列概率分解的次序进行全排列，使得模型可以摆脱原始自回归预训练模型只能利用单向信息的缺点；同时让模型在不需要对输入进行掩码操作的情况下也能学习到上下文信息，使得预训练与运用场景的输入分布保持一致。最终，XLNet在多个数据集上取得了优于BERT的结果。值得一提的是，XLNet使用Transformer-XL作为模型骨架，其中的相对位置编码与段循环(Segment Recurrence)机制，能让XLNet在长序列上也保持不错的效果。

**中英翻译预训练模型(Translation)** 上述涉及的预训练模型都是通过在大规模中文语料上训练获得更好的上下文表示，来提升模型对AMR语义解析中源端和目标端之间隐式关系的学习能力。由于中文AMR解析源端是中文文本，目标端是包含中英的AMR序列，因此除了使用已经训练好的中文预训练模型外，本文尝试利用中英翻译语料进行预训练，预训练的源端为中文，目标端为英文数据，最终使用正确AMR语料对模型进行微调(fine-tuning)。

## 5 实验

### 5.1 数据及子词化处理

本文使用CAMR1.0 (LDC2019T07)进行实验。CAMR1.0从CTB8.0中选取的10,149个句子,对这些句子进行AMR标注,其中训练集、验证集、测试集分别包含7,610、1,263、1,276个句子以及对应的中文AMR图。为了避免在有限的数据集下,低频词在训练中对模型性能造成影响,我们对源端和目标端文本进行子词化处理。我们使用BERT提供的子词化工具对数据集进行子词化处理,同时我们在后续的实验中也与目前比较常用的BPE工具<sup>1</sup>进行了比较。同时我们在子词化时,不对目标端中的英文概念和关系标签进行切分。表2中给出了使用不同子词化工具对源端句子和对应的目标端AMR进行处理的例子。

	子词化工具	举例
src	NoTokenizer	麦田 先生 变脸 功夫 实在 比 川剧 艺术 还 了 得。
	BPE	麦@@ 田 先生 变脸 功夫 实在 比 川@@ 剧 艺术 还 了 @@ 得。
	BertTokenizer	麦田 先生 变脸 功夫 实在 比 川剧 艺术 还 了 得。
tgt	NoTokenizer	(了得-01 :mod (实在) :arg0 (功夫 :mod (变脸) :poss (先生 :name (name :op1 "麦田" ))) :mod (还) :compared-to (艺术 :mod (川剧)))
	BPE	(了得-01 :mod (实在) :arg0 (功夫 :mod (变脸) :poss (先生 :name (name :op1 "麦田" ))) :mod (还) :compared-to (艺术 :mod (川@@ 剧)))
	BertTokenizer	(了得-01 :mod (实在) :arg0 (功夫 :mod (变脸) :poss (先生 :name (name :op1 "麦田" ))) :mod (还) :compared-to (艺术 :mod (川剧)))

Table 2: 不同子词化预处理方式示例

### 5.2 模型设置

本实验所使用的代码是以OpenNMT-py为基础的Transformer模型。实验中Transformer编码器和解码器的层数皆设置为6层,将多头注意力机制的头数设置为8。词嵌入向量和隐藏层的维度都设置为512,前馈神经网络层的维度为2048。实验中还使用了Adam优化器对参数更新进行优化,其中 $\beta_1 = 0.9$ 、 $\beta_2 = 0.98$ 、批次大小为8192个形符(Token)。除此之外,实验中的预热(Warm up)步数、学习率、丢弃(Dropout)率、标签平滑分别设为16000、2.0、0.3、0.1。在解码过程中,采用束搜索(Beam search),束的大小设置为5。在模型的选择时,选用在开发集上性能最好的模型进行测试。AMR解析的评测使用Smatch(Cai and Knight, 2013)及其他细粒度的评判指标<sup>2</sup>。

在预训练模型的实验中所采用的模型参数为: nezha-cn-base (NEZHA)<sup>3</sup>、bert-base-chinese (BERT)<sup>4</sup>、hfl/chinese-bert-wwm (BERT-wwm)<sup>5</sup>和hfl/chinese-xlnet-base (XLNet)<sup>6</sup>。使用Scalar Mix对预训练模型输出进行加权求和时所用的输出层数 $K$ 均设为12层。

### 5.3 实验分析

#### 5.3.1 实验结果

表3给出并比较了使不使用预训练模型以及使用不同预训练模型在中文AMR语义解析上的性能。对比基线模型和使用预训练模型的实验结果发现,在模型中加入预训练模型可以让解析器的Smatch和细粒度的指标都得到显著提升。以BERT为例,使用BERT的实验中,AMR语义解析的Smatch F1值由57.90提升到69.96。其中细粒度指标: Unlabeled、No WSD、Concepts、Named Ent.、Negations、Reentrancies、SRL分别有11.15、12.15、11.38、8.73、12.41、10.9、13.2个F1值的提升。这充分说明,通过预训练模型获取的上下文表征可以帮助模型提升中文AMR语义解析的性能。此外本文还使

<sup>1</sup><https://github.com/rsennrich/subwordnmt.git>

<sup>2</sup><https://github.com/mdtux89/amr-evaluation>

<sup>3</sup>[https://github.com/lonePatient/NeZha\\_Chinese\\_PyTorch](https://github.com/lonePatient/NeZha_Chinese_PyTorch)

<sup>4</sup><https://github.com/huggingface/transformers>

<sup>5</sup><https://github.com/ymcui/Chinese-BERT-wwm>

<sup>6</sup><https://github.com/ymcui/ChineseXLNet>

预训练模型	None			BERT			BERT-wwm		
	P	R	F1	P	R	F1	P	R	F1
Smatch	65.63	51.80	57.90	76.16*	64.70*	69.96*	76.38*	65.09*	<b>70.29*</b>
Unlabeled	72.90	57.07	64.02	82.19	69.25	75.17	82.40	69.64	<b>75.48</b>
No WSD	66.45	52.45	58.63	77.05	65.46	70.78	77.28	65.86	<b>71.12</b>
Concepts	71.98	58.53	64.56	81.45	71.13	75.94	82.00	71.62	<b>76.46</b>
Named Ent.	73.90	45.41	56.25	80.61	54.43	64.98	81.45	54.67	<b>65.43</b>
Negations	53.97	51.31	52.61	67.79	62.46	<b>65.02</b>	65.38	60.98	63.10
Reentrancies	34.30	32.00	33.10	43.00	45.10	44.00	42.80	46.00	44.40
SRL	59.70	49.80	54.30	72.10	63.40	67.50	71.70	63.80	67.50
预训练模型	NEZHA			XLNet			Translation		
	P	R	F1	P	R	F1	P	R	F1
Smatch	77.04*	64.08*	69.96*	73.49*	61.20*	66.78*	71.50*	61.89*	66.35*
Unlabeled	82.74	68.25	74.80	79.86	65.95	72.24	78.11	67.05	72.16
No WSD	77.94	64.82	70.78	74.29	61.86	67.51	72.50	62.75	67.28
Concepts	82.05	70.22	75.68	78.94	67.37	72.69	79.50	70.74	74.86
Named Ent.	82.46	54.03	65.28	80.02	54.03	64.51	80.27	51.64	62.85
Negations	66.31	61.31	63.71	65.27	58.85	61.90	62.48	57.87	60.09
Reentrancies	44.00	45.20	<b>44.60</b>	40.50	42.20	41.40	38.40	41.80	40.10
SRL	73.00	63.10	<b>67.70</b>	69.50	59.80	64.30	66.20	59.90	62.90

Table 3: 不同预训练模型细粒度性能, 其中\*表示在显著性测试中 $p < 0.005$ , 该预训练模型的性能较基线模型性能具有显著性提高.

用BERT-wwm、NEZHA、XLNet来提取源端的上下文表征, 不同的预训练模型使用的模型结构和训练策略各不相同, 因此不同预训练模型AMR解析任务上表现各不相同。对比BERT、BERT-wwm、NEZHA、XLNet的性能可以发现, 在模型中融合BERT-wwm的上下文表征在中文AMR语义解析任务上的性能最好, 最优性能可以达到70.29的F1值, 且在Unlabeled、No WSD、Concepts、Named Ent.等细粒度指标的性能都超越了使用其他预训练模型获得的性能。

此外, 图4中给出了在使用不同预训练模型时Scalar mix在不同层输出上的归一化权重s的可视化结果: 所有四个预训练模型的第一层输出表示的权重都是最大的, 均达到了20%以上, 其次为前4层和中部7-10层, 而最后两层的输出的权重则较小。不同层的表示的重要程度在整体上体现出了先下降后上升再下降的趋势。

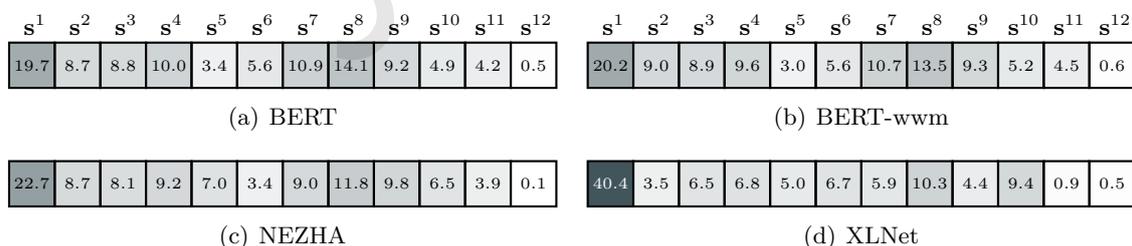


Figure 4: 不同预训练模型的Scalar Mix权重可视化结果, 数值为使用softmax进行归一化后的权重s(数值中的%已被省略)。

### 5.3.2 不同的子词化方法对中文AMR解析的影响

无论是BPE还是BERT-Tokenizer, 都是通过将句子中的低频词切分为更细粒度的子词, 从而使模型学习到拥有相同前缀或后缀的词之间的联系。但是通过观察BERT-Tokenizer子词化之后的结果, 发现对于中文来说该子词化方法并不会像英文一样根据词出现的频率进行切分, 而

是以字为单位切分中文句子。在本节中我们给出使用不同的子词化方法的中文AMR解析性能，探究不同的子词化方法对解析效果的影响。

	NoTokenizer			BPE			BERT-Tokenizer		
	word	avg. len.	avg. oov	word	avg. len.	avg. oov	word	avg. len.	avg. oov
src	16272	22.120	7.841%	14889	24.621	3.715%	3599	32.996	0.404%
tgt	18788	78.404	2.710%	17438	81.861	1.326%	3882	100.600	0.151%

Table 4: 基于不同子词化方式的数据统计对比

表 4给出了使用不同的子词化方法的词表大小、平均句长和平均未登录词(oov)占比。从中可知在使用BERT-Tokenizer对数据进行处理之后，模型的词表大小相较于不进行子词化方法或是使用BPE进行分词将大大降低，仅为不进行子词化处理的词表的1/4。且未登录词的数量在源端和目标端分别由7.841%、2.710%下降到0.404%、0.151%。由此可见使用以字为单位的子词化方法可以更好的减少中文AMR语料中的未登录词的数量，缩小源端和目标端词表规模。

表 5为不同分词在中文AMR语义解析上的实验结果。相比其他两种方法，使用BERT-Tokenizer进行子词化可以让模型取得更高的P和F1值，并且在所有细粒度指标上都取得最好的结果，其中在Concepts、Named Ent.和Negations上都提高了1%以上。尤其是在Negations指标，即对否定的识别语义的上分别比以词为单位和BPE子词化的结果高出了4.22%和4.67%。有趣的是较之于不进行子词化处理，BPE方法在否定语义的识别上表现得更差。如我们预想的那样，使用切分粒度更细的数据进行中文AMR语义解析的性能更好。

	NoTokenizer			BPE			BERT-Tokenizer		
	P	R	F1	P	R	F1	P	R	F1
Smatch	63.66	51.63	57.02	62.37	53.05*	57.34	65.63*	51.80	57.90*
Unlabeled	70.81	56.95	63.13	69.31	58.47	63.43	72.90	57.07	64.02
No WSD	64.39	52.23	57.68	63.13	53.70	58.04	66.45	52.45	58.63
Concepts	69.37	57.34	62.78	68.31	59.24	63.45	71.98	58.53	64.56
Named Ent.	69.38	44.85	54.48	64.26	47.65	54.72	73.90	45.41	56.25
Negations	51.48	45.74	48.44	50.36	45.74	47.94	53.97	51.31	52.61
Reentrancies	31.70	34.60	33.10	30.80	34.20	32.40	34.30	32.00	33.10
SRL	58.00	49.10	53.20	57.00	50.00	53.30	59.70	49.80	54.30

Table 5: 不同子词化方式性能差异，其中\*表示在显著性测试中 $p < 0.005$ ，相应模型性能较NoTokenizer模型的性能具有显著性提高。

推断可能的原因为：中文词汇按照构词语素，可以分为单纯词和合成词。单纯词中只包含一个语素，如“仿佛”和“巧克力”，组成他们的字并不单独表意。包含了多个语素的词语称为合成词，如“统治”和“论文”，他们由多个语素按照一定的规律结合进行表意。合成词在中文词汇中占有很大的比例，也就是说大部分词可以拆分成多个语素组合的形式。因此将字作为模型输入的基本单位，可以让模型更好得通过对语素的学习来把握词语整体的意思。同时这也能解释为什么以字为单位进行子词化能够让模型更好地识别否定语义。中文中否定语义主要通过“不”、“非”、“没”等否定语素进行表达。但是在进行分词时，这些否定语素往往会与其他语素进行结合构成合成词，如“不利”、“非法”和“没有”等。这些词又往往在文本中是作为常见词出现的，在进行BPE时他们将倾向于作为一个整体出现。因此模型难以学习到“不”、“非”、“没”这些否定语素的含义，在面对与否定语素进行构词而表示否定语义的罕见词和未登录词时模型也就显得心有余而力不足了。

## 5.4 总结

本文使用序列到序列的模型进行中文AMR语义解析，并在CAMR1.0 (LDC2019T07)上获得了57.90的F1值。并在此基础上使用中文预训练模型，通过中文预训练模型获取输入文本的上下文表征，并使用Scalar Mix获得上下文表征的最优性能，使用该方法获得了70.29的F1值。在

下一步的工作中，我们将研究如何运用概念对齐的CAMR1.0语料，在模型训练过程中增加更多的概念对齐信息来进一步提升中文AMR语义解析的性能。

## 参考文献

- Rafael Anchieta and Thiago Pardo. 2018. Towards amr-br: A sembank for brazilian portuguese language. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*.
- 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 the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.
- Guntis Barzdins and Didzis Gosko. 2016. Riga at semeval-2016 task 8: Impact of smatch extensions and character-level neural translation on amr parsing accuracy. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1143–1147.
- Michele Bevilacqua, Rexhina Biloshmi, and Roberto Navigli. 2021. One SPRING to rule them both: Symmetric AMR semantic parsing and generation without a complex pipeline. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*.
- Claire Bonial, Lucia Donatelli, Stephanie Lukin, Stephen Tratz, Ron Artstein, David Traum, and Clare Voss. 2019. Augmenting abstract meaning representation for human-robot dialogue. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 199–210.
- Shu Cai and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 748–752.
- Hyonsu Choe, Jiyeon Han, Hyejin Park, and Hansaem Kim. 2019. Copula and case-stacking annotations for korean amr. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 128–135.
- Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Ziqing Yang, Shijin Wang, and Guoping Hu. 2019. Pre-training with whole word masking for chinese bert. *arXiv preprint arXiv:1906.08101*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Donglai Ge, Junhui Li, Muhua Zhu, and Shoushan Li. 2019. Modeling source syntax and semantics for neural amr parsing. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 4975–4981.
- Bin Li, Yuan Wen, Weiguang Qu, Lijun Bu, and Nianwen Xue. 2016. Annotating the little prince with chinese amrs. In *Proceedings of the 10th Linguistic Annotation Workshop held in conjunction with ACL 2016*, pages 7–15.
- B Li, Y Wen, L Song, L Bu, W Qu, and N Xue. 2017. Construction of chinese abstract meaning representation corpus with concept-to-word alignment. *Journal of Chinese Information Processing*, 31(6):93–102.
- Bin Li, Yuan Wen, Li Song, Weiguang Qu, and Nianwen Xue. 2019. Building a chinese amr bank with concept and relation alignments. In *Linguistic Issues in Language Technology, Volume 18, 2019-Exploiting Parsed Corpora: Applications in Research, Pedagogy, and Processing*.
- Kexin Liao, Logan Lebanoff, and Fei Liu. 2018. Abstract meaning representation for multi-document summarization. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1178–1190.
- Li Song, Yuan Wen, Sijia Ge, Bin Li, and Weiguang Qu. 2019a. An easier and efficient framework to annotate semantic roles: Evidence from the chinese amr corpus. In *Proceedings of the 20th Chinese Lexical Semantics Workshop*, pages 474–485.
- Linfeng Song, Daniel Gildea, Yue Zhang, Zhiguo Wang, and Jinsong Su. 2019b. Semantic neural machine translation using amr. *Transactions of the Association for Computational Linguistics*, 7:19–31.

- Rik van Noord and Johan Bos. 2017. Neural semantic parsing by character-based translation: Experiments with abstract meaning representations. *Computational Linguistics in the Netherlands Journal*, 7:93–108.
- 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 the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010.
- Chuan Wang, Nianwen Xue, and Sameer Pradhan. 2015. A transition-based algorithm for amr parsing. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 366–375.
- Chuan Wang, Bin Li, and Nianwen Xue. 2018. Transition-based chinese amr parsing. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 247–252.
- Junqiu Wei, Xiaozhe Ren, Xiaoguang Li, Wenyong Huang, Yi Liao, Yasheng Wang, Jiashu Lin, Xin Jiang, Xiao Chen, and Qun Liu. 2019. Nezha: Neural contextualized representation for chinese language understanding. *arXiv preprint arXiv:1909.00204*.
- Dongqin Xu, Junhui Li, Muhua Zhu, Min Zhang, and Guodong Zhou. 2020. Improving amr parsing with sequence-to-sequence pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 2501–2511.
- Dongqin Xu, Junhui Li, Muhua Zhu, Min Zhang, and Guodong Zhou. 2021. Xlpt-amr: Cross-lingual pre-training via multi-task learning for zero-shot amr parsing and text generation. In *Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*.
- Nianwen Xue, Ondrej Bojar, Jan Hajic, Martha Palmer, Zdenka Uresova, and Xiuhong Zhang. 2014. Not an interlingua, but close: Comparison of english amrs to chinese and czech. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, pages 1765–1772.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32, pages 5753–5763.
- Junsheng Zhou, Feiyu Xu, Hans Uszkoreit, Weiguang Qu, Ran Li, and Yanhui Gu. 2016. Amr parsing with an incremental joint model. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 680–689.
- 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 the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 5459–5468.
- 吴泰中, 顾敏, 周俊生, 曲维光, 李斌, and 顾彦慧. 2019. 基于转移神经网络的中文amr 解析. *中文信息学报*, 33(4):1–11.
- 李斌, 闻媛, 宋丽, 卜丽君, 曲维光, and 薛念文. 2017. 融合概念对齐信息的中文amr 语料库的构建. *中文信息学报*, 31(6):93–102.