SUDA-HUAWEI 系统报告 - CAMRP2022 评测

周仕林[1]✉, **夏庆荣**[2], **李扬**[1], **王喆锋**[2], **李正华**[1], **怀宝兴**[2], **张民**[1]
1. 苏州大学 计算机科学与技术学院，江苏 苏州
2. 华为云，浙江 杭州
(✉ 联系邮箱 slzhou.cs@outlook.com)

摘要

本文介绍了我们在第二十一届中国计算语言学大会中文抽象语义表示解析评测中提交的参赛系统。抽象语义表示（Abstract Meaning Representation，AMR）以图的方式表示一句话的语义。本次评测任务针对中文抽象语义表示（Chinese Abstract Meaning Representation，CAMR），参赛系统除了需要对常规的 AMR 图信息进行预测，还需要预测中文 AMR 特有的概念节点对齐、虚词、同指信息。我们基于 PERIN (Samuel and Straka, 2020) 以非自回归的方式生成 AMR 语义图，并设计了相应的后处理方式来预测概念节点对齐、虚词、同指信息。此外，我们还利用多图聚合的策略来进一步提高预测的精度。最终，我们在 closed 和 open 两个赛道，两个测试集上均取得第一名。

关键词： 抽象语义表示；非自回归；图聚合

System Report of SUDA-HUAWEI for CAMRP2022 Evaluation

Shilin Zhou[1]✉, **Qingrong Xia**[2], **Yang Li**[1], **Zhefeng Wang**[2],
Zhenghua Li[1], **Baoxing Huai**[2], **Min Zhang**[1]
1. School of Computer Science and Technology, Soochow University, Suzhou, China
2. Huawei Cloud, Hangzhou, China
(✉ Corresponding Email: slzhou.cs@outlook.com)

Abstract

This paper introduces the system we submitted in the shared task of Chinese Abstract Meaning Representation (CAMR) at the Twenty-first China National Conference on Computational Linguistics. The participating systems need to predict not only conventional AMR graph , but also node alignment, function words, and coreference information unique to CAMR. We generate AMR graph based on PERIN (Samuel and Straka, 2020) in a non autoregressive manner, and design the corresponding post-processing methods to predict node alignment, function words, and coreference. In addition, we also use the strategy of multi-graph ensembling to further improve the accuracy of prediction. In the end, we won the first place in both closed track and open track and in both test sets.

Keywords: Abstract Meaning Representation , Non Autoregressive , Graph Ensembling

1 任务介绍

抽象语义表示（Abstract Meaning Representation，AMR）以有向无环图的方式来表示一句话的语义 (Banarescu et al., 2013)。图中的节点可以是句中的词，也可以是由词抽象得到的概

念，节点之间的关系通过图中的边来表示。AMR 作为自然语言处理领域中的重要任务，已经被广泛应用于机器翻译 (Song et al., 2019) 和知识问答 (Kapanipathi et al., 2020) 等下游任务。

本次评测主要针对中文 AMR。和英文 AMR 不同，中文 AMR 增加了节点对齐信息，模型需要预测出与节点对齐的词。此外，虚词信息（关系对齐）和同指信息也是本次评测任务目标。在中文语句中，虚词指没有实际词汇含义的词，如“的”、“把”、“被”等 (Dai et al., 2020)，但虚词有着丰富的语义信息，可以表达概念和情态意义。同指信息指在一句话的 AMR 图中有着相同抽象意义的概念节点。虚词和同指信息对句子的理解都有着很大的帮助，所以本次评测也将虚词和同指纳为了评价对象，这也是本次评测任务的难点。

AMR 自被提出以来便引起了学术界的广泛关注和研究。现阶段主流的 AMR 方法主要分为自回归方法和非自回归方法两种。自回归的方法首先按照深度遍历或者广度遍历的顺序将 AMR 图中的节点和关系排序，Zhang et al. (2019) 和 Cai and Lam (2020) 根据得到的线性顺序依次预测节点以及节点之间的关系。Bevilacqua et al. (2021) 进一步将排序后的节点和关系序列化为字符串，然后通过 seq2seq 的方法解析 AMR 图。不同于自回归的方法，非自回归的方法不需要按照某个顺序生成节点，所有的节点均是同时生成，这也更符合 AMR 图的结构。Samuel and Straka (2020) 将 AMR 节点对齐到句子中的词，以词为单位去同时生成节点。此外还有一类基于转移的方法 (Wang et al., 2018)，他们通过预先定义的栈、缓冲区以及转移动作来逐步构建 AMR 图。在本次评测中，我们基于 PERIN 模型 (Samuel and Straka, 2020)，以非自回归的方式生成 AMR 图。我们的代码发布在 https://github.com/zsLin177/camr.

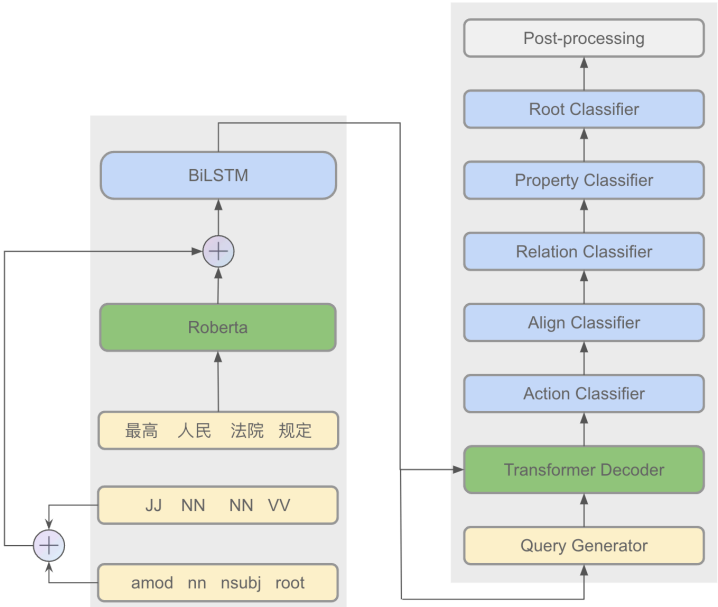

Figure 1: 模型结构图

2 模型

本节介绍了在这次评测中我们所使用模型的结构，我们借鉴 PERIN (Samuel and Straka, 2020)，先使用非自回归的方式生成 AMR 语义图中的节点，再基于生成的节点预测对齐信息，以及节点之间的关系信息。模型主要分为编码器和解码器两个部分，编码器部分使用 Roberta (Cui et al., 2020) 编码输入句子，并通过 BiLSTM 融入词性和依存句法信息。解码器包括动作预测、对齐预测、关系预测、属性判断和根节点预测。最后为了输出中文 AMR 中特有的节点对齐、虚词、共指信息，我们设计了相应的后处理操作。图 1展示了我们模型的整体框架。

2.1 编码器

编码器主要由 Roberta 和 BiLSTM 两部分组成。我们利用预训练语言模型 Roberta 来编码输入句子 X，来得到每个词 x_i 词级别的向量表示 $\mathbf{r}_i$：

$$\mathbf{r}_i = \text{Roberta}(x_i) \tag{1}$$

因为 Roberta 是以子词为单位进行编码，而我们这里的词是以空格分隔得到的完整词，所以我们直接将属于同一个完整词的子词的表示相加，得到最终的词表示 $\mathbf{r}_i$。

我们利用词性标签和依存句法中依存弧的标签来加入句法信息到我们的模型中，即对于每一个输入词 x_i，我们可以分别得到它对应的词性向量表示 $\mathbf{e}_i^{pos}$ 和句法向量表示 $\mathbf{e}_i^{syn}$。最后我们将得到的 $\mathbf{r}_i$、$\mathbf{e}_i^{pos}$、$\mathbf{e}_i^{syn}$ 拼接后输入一层 BiLSTM 得到最终的编码器的输出向量 $\mathbf{h}_i$。

$$\mathbf{h}_i = \text{BiLSTM}(\mathbf{r}_i \oplus \mathbf{e}_i^{pos} \oplus \mathbf{e}_i^{syn}) \tag{2}$$

2.2 解码器

在解码阶段，节点由句中的词以非自回归的方式生成，每个词最多可以生成 K 个节点。我们为词 x_i 生成 K 个 query 表示:$\mathbf{q}_i^1, ..., \mathbf{q}_i^j, ..., \mathbf{q}_i^{\text{K}}$，并为 $\mathbf{q}_i^j$ 预测动作标签 a_i^j，来决定由词 x_i 生成什么语义节点。具体地，我们先将编码器的输出向量 $\mathbf{h}_i$ 输入到 Query Generator 中。Query Generator 通过 K 个线性层以及 tanh 激活函数生成词 x_i 对应的 K 个 query 表示：$q_i^1, ..., q_i^j, ..., q_i^{\text{K}}$，在实验中，我们设置 $\text{K} = 3$。

$$\mathbf{q}_i^j = \tanh(\text{MLP}^j(\mathbf{h}_i)) \tag{3}$$

接下来，将生成的 query 向量和编码器的输出向量 $\mathbf{h}_i$ 输入到三层 Transformer Decoder (Vaswani et al., 2017) 中得到 $\mathbf{v}_i^j$。最终解码器根据 $\mathbf{v}_i^j$，使用动作分类器（Action Classifier），对齐分类器（Align Classifier），关系分类器（Relation Classifier），属性分类器（Property Classifier）和根节点分类器（Root Classifier），来预测 $\mathbf{v}_i^j$ 在 AMR 图中作为什么语义节点，该节点对齐到哪些词，该节点与其他节点的关系，该节点是否是一个属性节点，以及该节点是否是根节点。

- 动作预测：动作分类器使用线性层打分来预测 $\mathbf{v}_i^j$ 应该生成什么节点，目标动作包括“copy”(拷贝词 x_i 来生成节点)、“add-01”（在词 x_i 后添加“-01”，来生成节点 $x_i - 01$)、“generate [and]”（生成节点“and”）等。此外还有一个特殊动作 [NULL]，表示 $\mathbf{v}_i^j$ 不生成节点，如果 $\mathbf{v}_i^j$ 预测出的动作是 [NULL]，这就表示 $\mathbf{v}_i^j$ 对应的 query 不在 AMR 图中，也就不参与之后的对齐等预测。在这次评测中，我们共统计有 1810 种动作标签。

- 对齐预测：因为一个语义节点可以对齐到多个词，所以对齐分类器使用 Biaffine attention (Dozat and Manning, 2017) 来判断 $\mathbf{v}_i^j$ 与每个词是否有对齐关系，得到对齐词集合 AlignSet_i^j。

- 关系预测：对于预测出的所有节点，我们分别使用两个 Biaffine attention 来判断一对节点之间是否存在关系，和存在什么关系。因为在中文 AMR 中存在着对应实词间的关系意义的虚词 (Dai et al., 2020)，我们将此类虚词和关系标签组合，形成复合标签，如“arg0+被”、“domain+ 是”、“location+ 的”等。我们为了减少标签空间，节省显存占用，只保留了那些出现次数大于 1 的标签，最终在我们的模型中共有 1042 种关系标签。

- 属性预测：在我们的模型中，我们将节点的属性（如“op1”、“op2”）也处理成了该节点的孩子节点。属性分类器需要对每个节点进行二分类来判断该节点是否是属性节点，若该节点是属性节点则将该节点从图中删去，并处理成其父亲节点的属性。

- 根节点预测：根节点分类器使用线性层打分来预测哪个节点是 AMR 图的根节点。

2.3 后处理

后处理部分主要分为三个部分，分别是规范化对齐、匹配虚词的对齐位置、同指节点的处理。

- 规范化对齐：因为在我们的模型中，节点的对齐是词语级别的，而在中文 AMR 中，某些节点是对齐到词语中的某几个字符的，如节点“30”，对齐到词“30 余”的第一和第二个字符。我们通过后处理的方式，处理这种字符级的对齐。具体地，在由对齐预测得到对齐词集合 AlignSet_i^j 后，我们将对齐词集合中的词按照顺序拼接得到对齐词字符串，若节点字符串是对齐词字符串的子串，我们就在对齐词字符串中从左往右匹配节点字符串，找到第一个匹配位置，并返回字符级的对齐。

- 匹配虚词的对齐位置：在关系预测中，我们将虚词和关系标签组合成复合标签。在模型预测出复合标签，我们得到该关系的虚词后，我们通过匹配的方式去寻找该虚词的匹配位置。具体地，我们寻找距离该关系对应的父亲节点的对齐词最靠近的位置。特殊地，如果该句中没有匹配的位置，则返回头节点的对应词的后面的一个位置。

- 同指节点的处理：在训练阶段，我们首先随机选取同指节点中的一个节点作为核心节点，并将其他同指节点用核心节点的编号代替，如“x1”，“x2”。在预测阶段，若解码得到的 AMR 图中存在标签为编号的节点，则用该编号对应的核心节点的标签代替原节点标签，并将编号相同的节点归为同指节点。

3 图聚合

模型聚合是指利用多个模型来得到一个相比使用单个模型更稳定、更精确的预测结果 (Domingos, 2000)，这项技术在各种领域的科研和竞赛中被广泛利用，并取得了非常亮眼的表现 (Chen and Guestrin, 2016)。在本次评测中我们首先尝试了较为简单的基于参数的聚合和基于分数的聚合，均不能达到预期的目标。最终我们基于 Hoang et al. (2021) 提出的图聚合，并针对本次评测需要预测对齐的特点做出改进，达到了不错的效果。

给定由 m 个模型预测得到的图集合 $S = \{g_1, \cdots, g_m\}$，图聚合的核心思想是得到这 m 个图的最大公共子图，将该最大公共子图作为最终的聚合图 g_e。但该问题是 NP-Hard 的，Hoang et al. (2021) 基于 Cai and Knight (2013) 提出的得到两个图之间的最优匹配的近似算法，提出一个启发式算法来得到多个图的聚合图 g_e。图聚合算法在每轮迭代中首先选择 $S = \{g_1, \cdots, g_m\}$ 中的一个图 g_i 作为该轮迭代的核心图，由该核心图初始化得到节点标签表，边标签表，接着得到图 g_i 与剩余 m-1 个图的最优匹配，根据匹配过程去更新节点标签表和边标签表。在完成更新后删去表中得票低的标签，并选择得票最高的标签作为该节点（边）的标签，得到以 g_i 为核心图的聚合图 g_e^i。每一轮的过程可以看作是在对该轮的核心图中的节点的标签和边标签进行投票，并保留票数最高的标签。最终在完成 m 轮迭代后，选择 $\{g_e^1, \cdots, g_e^m\}$ 中总票数最高的聚合图作为最终的 g_e。具体的算法过程我们建议读者阅读原文 Hoang et al. (2021)。

在本次评测任务中，我们需要输出 AMR 图中每个节点的对齐信息，而原本的图聚合算法并没有考虑到这一点，我们改进了图聚合算法，在每一轮的迭代中也对每个节点的对齐词集合进行投票，保留得票最高的对齐词集合，以提高对齐的准确率。

4 实验

4.1 实验设置

本次评测共分为 closed 和 open 两个赛道，我们在两个赛道中均只使用官方提供的训练集和开发集进行模型的训练。模型基于 pytorch 框架开发，核心代码和参数设置参考 PERIN (Samuel and Straka, 2020) 开源模型。在 closed 赛道，我们仅微调 Roberta (Cui et al., 2020)[0]来提高编码器能力，在 open 赛道，除了 Roberta 模型外，我们也利用了另外两种预训练语言模型 MacBERT[1]和 PERT (Cui et al., 2022)[2]来训练模型，增强不同模型之间的差异性，获得更多样性的结果，以在图聚合的时候得到更稳定的结果。

另外，本次评测共有两个测试集 TestA 和 TestB，TestA 提供了词性和依存句法信息来帮助 AMR 的预测，TestB 不提供词性和依存句法信息。所以在 closed 赛道，我们并没有利用词性和依存句法信息来帮助对 TestB 的解析。在 open 赛道我们使用训练集，分别基于 SAS [3] 和

[0]https://huggingface.co/hfl/chinese-roberta-wwm-ext-large
[1]https://huggingface.co/hfl/chinese-macbert-large
[2]https://huggingface.co/hfl/chinese-pert-large
[3]https://github.com/zsLin177/syntax_aware_seqtag

Team	TestA			TestB		
	P	R	F_1	P	R	F_1
closed						
SUDA&HUAWEI	**81.83**	**78.25**	**80.00**	**75.16**	**70.28**	**72.64**
PKU	78.61	76.54	77.56	71.53	69.96	70.74
NJU	65.25	62.20	63.69	57.87	56.79	57.32
ECNU	67.86	59.03	64.14	60.84	52.64	56.45
ECUST	41.91	26.10	32.17	32.86	21.51	26.00
open						
SUDA&HUAWEI	**82.16**	**78.20**	**80.13**	**75.52**	**71.79**	**73.61**
ECNU	73.83	66.05	69.72	66.01	57.71	61.58
BUPT	50.41	42.55	46.15	49.95	42.72	46.05

Table 1: 所有参赛队伍的 AlignSmatch 得分

Model Num	P	R	F_1	
1	79.03	78.83	78.93	
3	82.10	76.62	79.27	+0.34
6	81.83	78.25	80.00	+1.07
9	**82.16**	**78.20**	**80.13**	+1.20

Table 2: 不同数量模型进行图聚合在 TestA 上的结果

Supar[4] 训练了一个 CRF 序列标注模型和一个 Biaffine 依存句法模型，来预测 TestB 的词性标签和依存句法标签。

4.2 评测结果

本次评测主要使用 AlignSmatch 作为评价指标，相较于 AMR 中常用的 Smatch，AlignSmatch 将中文 AMR 包含的概念节点对齐信息和虚词对齐信息也纳为了评价对象 (Xiao et al., 2022)。表 1列出了本次评测任务中所有参赛队伍的 AlignSmatch 得分。可以看到，在 closed 和 open 两个赛道的两个测试集上，我们的方法均取得了最佳的成绩。具体地，在 closed 赛道领先第二名约百分之二的 F_1 值，在 open 赛道领先第二名约百分之十的 F_1 值。

4.3 结果分析

4.3.1 图聚合

在本次评测中我们发现通过图聚合的方式相比单个模型可以带来稳定提升。表 2给出了在 TestA 上，我们使用不同数量的模型进行图聚合的结果对比，其中单个模型的结果为我们所训练的多个单模型中表现最好的模型结果。从表中可以看出，图聚合所使用的模型数量越多，性能表现也就越好。使用 3 个模型进行聚合可以带来 0.34 的 F_1 提升，使用 6 个模型则可以进一步带来 0.7 的 F_1 提升。

但是当使用更多的模型进行聚合时，图聚合带来的提升也在渐渐减少。与使用 6 个模型进行聚合相比，使用 9 个模型只能带来 0.13 的 F_1 提升。此外，我们还发现图聚合大幅提高了预测的准确率（P），但是召回率（R）有略微降低。这也反映了图聚合算法的本质是综合考虑不同模型的预测结果，用投票的思想来提高模型的精度。

最终在本次评测中，在 closed 赛道我们使用 6 个模型来进行图聚合，6 个模型使用的预训练模型均为 Roberta。我们在 open 赛道使用 9 个模型进行图聚合，除了 closed 赛道用到了 6 个基于 Roberta 的模型，我们另外使用了两个基于 MacBERT 的模型和一个基于 PERT 的模型。

	TestA			TestB		
	P	R	F_1	P	R	F_1
w/ pos&syn	78.53	79.06	78.79	71.92	71.94	71.93
w/o pos&syn	78.26	77.59	77.92	70.93	70.56	70.74

Table 3: 词性和依存句法信息对结果的影响

4.3.2 词性和依存句法

在实验中，我们也发现词性和依存句法信息对 AMR 解析有稳定帮助。表 3给出了词性和句法对我们模型的影响（这里给出是我们单个模型的结果，没有使用图聚合）。可以看到在 TestA 上，词性和句法可以带来 0.87 的 F_1 提升。在 TestB 上，词性和句法可以带来 1.19 的 F_1 提升。可以看出在语体风格和训练集差异较大的测试集上，词性和句法可以带来更为明显的提升。

5 结语

在本次 CAMR2022 评测任务中，我们使用非自回归的方法生成 AMR 图，并通过后处理的方式输出中文 AMR 特有的节点对齐、虚词、同指信息。另外，我们使用了多图聚合的方法来进一步提高预测的性能，并探究了词性和句法信息对 AMR 解析的帮助，最终我们在 closed 和 open 赛道的两个测试集上均位列第一。

但是，我们的系统依然有不足之处。例如，在 open 赛道，我们使用的预训练语言模型的结构基本一致，且仅仅使用了官方提供的训练集和开发集进行系统的开发，未来可以尝试探究基于 seq2seq 架构的预训练语言模型对 AMR 的影响，以及使用大规模伪数据对模型的帮助。此外，对于虚词的对齐以及同指的预测，我们是通过后处理的方式进行，后续可以探究不同的模型结构，以更好地支持对这两者的预测。

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 the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

Michele Bevilacqua, Rexhina Blloshmi, and Roberto Navigli. 2021. One spring to rule them both: Symmetric amr semantic parsing and generation without a complex pipeline. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 12564–12573.

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.

Deng Cai and Wai Lam. 2020. AMR parsing via graph-sequence iterative inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1290–1301, Online, July. Association for Computational Linguistics.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for Chinese natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 657–668, Online, November. Association for Computational Linguistics.

Yiming Cui, Ziqing Yang, and Ting Liu. 2022. Pert: Pre-training bert with permuted language model.

Yuling Dai, Rubing Dai, Minxuan Feng, Bin Li, and Weiguang Qu. 2020. Representation and analysis of abstract meaning of chinese function words based on relation alignment. *Journal of Chinese Information Processing*, 34(4):21.

[4]https://github.com/yzhangcs/parser

Pedro Domingos. 2000. A unified bias-variance decomposition. In *Proceedings of 17th international conference on machine learning*, pages 231–238. Morgan Kaufmann Stanford.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *Proceedings of International Conference on Learning Representations*.

Thanh Lam Hoang, Gabriele Picco, Yufang Hou, Young-Suk Lee, Lam Nguyen, Dzung Phan, Vanessa López, and Ramon Fernandez Astudillo. 2021. Ensembling graph predictions for amr parsing. *Advances in Neural Information Processing Systems*, 34:8495–8505.

Pavan Kapanipathi, Ibrahim Abdelaziz, Srinivas Ravishankar, Salim Roukos, Alexander Gray, Ramon Astudillo, Maria Chang, Cristina Cornelio, Saswati Dana, Achille Fokoue, et al. 2020. Question answering over knowledge bases by leveraging semantic parsing and neuro-symbolic reasoning. *arXiv preprint arXiv:2012.01707*.

David Samuel and Milan Straka. 2020. ÚFAL at MRP 2020: Permutation-invariant semantic parsing in PERIN. In *Proceedings of the CoNLL 2020 Shared Task: Cross-Framework Meaning Representation Parsing*, pages 53–64, Online, November. Association for Computational Linguistics.

Linfeng Song, Daniel Gildea, Yue Zhang, Zhiguo Wang, and Jinsong Su. 2019. Semantic neural machine translation using AMR. *Transactions of the Association for Computational Linguistics*, 7:19–31.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

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, New Orleans, Louisiana, June. Association for Computational Linguistics.

Liming Xiao, Bin Li, Zhixing Xu, Kairui Huo, Minxuan Feng, Junsheng Zhou, and Weiguang Qu. 2022. Align-smatch: A novel evaluation method for chinese abstract meaning representation parsing based on alignment of concept and relation.

Sheng Zhang, Xutai Ma, Kevin Duh, and Benjamin Van Durme. 2019. AMR parsing as sequence-to-graph transduction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 80–94, Florence, Italy, July. Association for Computational Linguistics.