

# Joint Entity and Relation Extraction With Set Prediction Networks

## 摘要

论文提出了一种用于实体与关系联合抽取的集合预测网络。该任务旨在从给定的句子中提取所有提及的关系三元组。为了解决这个问题，论文将实体与关系联合抽取视为直接的集合预测问题，使得抽取模型不需要预测多个三元组的顺序。为了解决这个集合预测问题，论文提出了一种基于非自回归并行解码的 transformer 网络。与自回归方法一次生成一个三元组的特点不同，所提出的网络能够直接输出最终的关系三元组集合。此外，论文还设计了一种基于集合的损失函数，通过二分图匹配得到唯一预测。与交叉熵损失函数相比，该二分图匹配损失对于预测的任何排列都是不变的，因此它可以通过忽略三元组的顺序，专注于关系类型和实体，为所提出的网络提供更准确的训练信号。在两个基准数据集上进行的各种实验证明了论文提出的模型明显优于当前最先进的模型。

**关键词：**二分图匹配；联合实体；关系抽取；非自回归解码器；集合预测

## 1 引言

关系三元组由两个实体通过一种语义关系连接而成，其形式为（主语，关系，宾语）。从非结构化的原始文本中抽取关系三元组是自动知识图谱构建的一项关键技术，近年来受到了广泛的关注。早期的研究，如 [1]，采用流水线的方式来提取实体和关系，首先识别实体，然后预测实体之间的关系。这种方法忽略了实体识别和关系预测的相关性 [2]，并且容易遭受严重的错误传播问题。为了显式地建模实体识别和关系预测之间的依赖关系，防止错误以管道的方式传播，后续研究提出了实体与关系联合抽取。这些研究大致可以归纳为三种主要范式。第一种范式，如 [3]，将联合实体和关系抽取任务视为端到端的填充问题。这些方法虽然使用共享参数来表示单个模型中的实体和关系，但是实体和关系是分开提取的，会产生冗余信息。第二种范式，如 [4]，将联合实体和关系抽取转化为序列标注。第三种范式，如 [5]，是由序列到序列（seq2seq）模型 [6] 直接生成关系三元组，这是一种处理重叠三元组的灵活框架，不需要人类专家的大量努力。论文遵循基于 seq2seq 的模型进行实体与关系联合抽取。尽管现有的基于 seq2seq 的抽取模型取得了成功，但它们仍然受到自回归解码器和交叉熵损失的限制。原因在于：句子中包含的关系三元组在形式上是集合，在本质上没有内在的顺序。然而，为了适应输出为序列的自回归解码器，在训练阶段，无序的目标三元组必须按照一定的顺序进行排序。同时，交叉熵是一个排列敏感的损失函数，其中，对预测出来的每一个三元组都会产生一个惩罚项。因此，目前基于 seq2seq 的模型不仅需要学习如何生成三元组，还需要考虑

多个三元组的提取顺序。在这项工作中，论文将实体与关系联合抽取任务建模为集合预测问题，避免了考虑多个三元组的顺序。为了解决集合预测问题，论文提出了一种具有非自回归并行解码和二分匹配训练损失的 transformer 为特征的端到端网络。具体而言，论文提出的集合预测网络 (Set Prediction Networks, SPNs) 包括三个部分：句子编码器、集合生成器和基于集合的损失函数。论文的主要贡献可以概括如下。1) 将实体与关系联合抽取任务建模为集合预测问题。2) 将非自回归并行译码与二分图匹配损失函数相结合来解决该问题。3) 提出的方法在两个基准数据集上获得了最先进的结果 (SoTA)，并进行了各种实验来验证方法的有效性。

## 2 相关工作

### 2.1 关系抽取

关系抽取是一项从自由文本中挖掘事实性知识，并且长期存在的自然语言处理任务。当给出一个带有标注实体的句子时，将退化为一个关系分类任务。一些研究利用卷积神经网络，循环神经网络或基于图的方法来解决关系分类任务。然而，这些方法必须依赖于预先抽取的实体，无法真正抽取关系三元组。当给定一个没有任何注释实体的句子时，各种方法，如 [7]，被提出来进行实体与关系联合抽取。现有关于实体和关系联合抽取任务的研究大致可以分为四种主要范式。第一种范式，首先识别实体 [8]，然后进行关系分类 [9]。尽管这种流水线结构在实际中得到了广泛的应用，但容易发生错误传播。第二种范式，用共享参数表示实体和关系。这种方法分别提取实体和关系，导致三元组构建过程中存在冗余信息。第三种范式，将任务视为序列标注问题。这类研究需要专家设计复杂的标注图式，但存在级联错误。第四种范式，基于 seq2seq 的方法，直接应用 seq2seq 模型生成关系三元组，不存在级联错误。但是，基于 seq2seq 的模型需要事先用一些启发式的全局规则将关系三元组的集合排序成一个序列，这破坏了三元组的自然集合结构。为了解决这个瓶颈，论文将实体与关系联合抽取视为一个集合预测问题，并提出了基于自回归并行解码和二分匹配损失函数的网络。

### 2.2 非自回归解码器

自回归解码器，如 [10]，根据先前生成的 token 序列生成每个 token。其中可能输出的分布，被分解为具有从左到右因果结构的条件概率链。

$$p(Y|X) = \prod_{i=1}^T p(y_i|y_{<i}, X) \quad (1)$$

式中：Y、X、T 分别表示输出序列、输入序列和输出序列的长度。与自回归解码器相比，非自回归解码器并行生成一个目标的所有 token，可以加快推理速度。非自回归解码器假设每个 token 都是在输入句子的条件下独立生成的。非自回归解码器对可能输出的分布进行如下建模：

$$p(Y|X) = p_L(T|X) \prod_{i=1}^T p(y_i|X) \quad (2)$$

其中，输出序列长度 T 用单独的条件分布  $p_L(Y|X)$  建模。由于自回归解码器在推理过程中具有较低的延迟，因此有丰富的文献对此进行了研究，如 [11]。如今，非自回归解码器已经在自

然语言和语音处理任务中被广泛探索，例如机器翻译和自动语音识别。同时，这是首次将非自回归模型应用于信息抽取领域。

### 2.3 集合预测

预测集的问题是元素的输出顺序是任意的。迄今为止，还没有专门设计的典型深度学习模型来直接预测集合。稠密连接网络足以进行固定规模的集合预测，但计算代价较高。另一种方法是利用自回归序列模型，如递归神经网络，逐个生成集合的所有元素。在所有情况下，损失函数应该通过预测的排列来保持不变，因为不能保证目标集合中的元素恰好与它们产生的顺序相同。基于指派的损失是一种流行的选择 [12]，可以保证每个目标元素都有唯一的匹配。此外，文献 [?] 提出了一种基于特征排序池化 (FSPool) 的方法。受集合预测研究的启发，由于句子中包含的关系三元组是无序的，论文将实体与关系联合抽取任务建模为集合预测问题。

## 3 本文方法

### 3.1 本文方法概述

论文提出的模型结构如下图 1

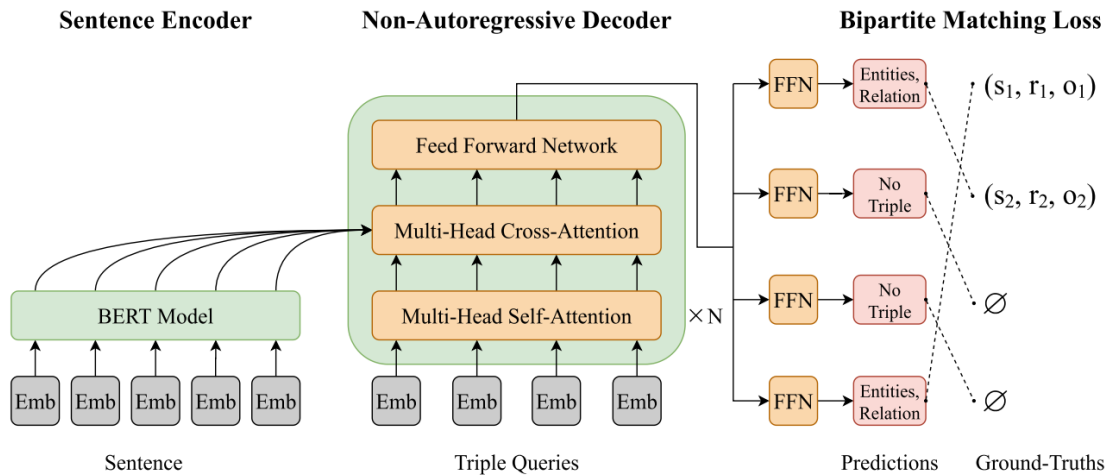


图 1. 模型结构图

网络中有三个关键组成部分。首先是句子编码器，它根据给定句子的上下文语义来表示该句子中的每个标记。然后是使用非自回归解码器生成一组三元组。最后，提出一种基于集合的损失，称为二分匹配损失，它迫使模型的输出与真实值之间存在唯一的匹配。

### 3.2 句子编码器

该模块是获得输入句子中每个 token 的上下文感知表示。由于 Transformer 在语言模型训练上的出色表现，论文使用 Transformer 的双向编码器表示 (BERT) 作为句子编码器。在 BERT 编码器中，输入句子通过字节对编码进行 token 分割，然后输入到编码器中。在 Transformer 模块中，有两个主要组成部分：一个自注意力子层和一个位置前馈神经网络。

### 3.3 用于三元组生成的非自回归解码器

论文将实体与关系联合抽取看作一个集合预测问题，使用基于 Transformer 的非自回归解码器直接生成三元组集合。以前的研究，将三元组集合转化为三元组序列，然后利用自回归解码器逐个生成三元组。这样，目标三元组的条件概率被修改为

$$p(Y|X; \theta) = \prod_{i=1}^n p(y_i|X, Y_{j<i}; \theta) \quad (3)$$

然而，论文使用非自回归解码器来指导模型。与以往基于 seq2seq 的方法相比，非自回归解码器不仅可以避免学习多个三元组的抽取顺序，而且可以基于双向信息生成三元组，而不仅仅是左右信息。

1) 输入：在解码开始之前，解码器需要知道目标集合的大小；在这项工作中，论文通过要求非自回归解码器为每个句子生成固定大小的  $m$  个预测集合，将  $p_L(n|X)$  简化为一个常数，其中  $m$  被设置为显著大于句子中三元组的数量。解码器的输入不是从编码器端复制 token，而是由  $m$  个可学习的嵌入初始化，称之为三元组查询。注意到所有句子共享相同的三元组查询。

2) 解码器结构：非自回归解码器由  $N$  个完全相同的 Transformer 模块堆叠而成。在每个模块中，有一个多头自注意力子层建模三元组之间的关系，有一个多头交叉注意力子层来融合给定句子的信息。与多头自注意力子层不同的是，多头交叉注意力子层中使用的查询由上一个多头交叉注意力子层的输出进行投影，而密钥和值则由 BERT 编码器的输出进行投影。

通过非自回归解码器，将  $m$  个三元组查询转化为  $m$  个输出嵌入，记为  $H_d \in R^{m \times d}$ 。然后，通过前馈网络 (FFNs) 将输出嵌入  $H_d$  独立地解码为关系类型和实体，从而产生  $m$  个最终的预测三元组。具体地，给定  $H_d$  中的一个输出嵌入  $h_d \in R^d$ ，则预测的关系类型为对预测实体 (主体与客体) 进行解码

$$p^r = \text{softmax}(W_r h_d) \in R^t \quad (4)$$

分别对起始和终止指标进行预测，得到 4 个类别的分类器

$$p^{s-start} = \text{softmax}(v_1^T \tanh(W_1 h_d + W_2 H_e)) \in R^1 \quad (5)$$

$$p^{s-end} = \text{softmax}(v_2^T \tanh(W_3 h_d + W_4 H_e)) \in R^1 \quad (6)$$

$$p^{o-start} = \text{softmax}(v_3^T \tanh(W_5 h_d + W_6 H_e)) \in R^1 \quad (7)$$

$$p^{o-end} = \text{softmax}(v_4^T \tanh(W_7 h_d + W_8 H_e)) \in R^1 \quad (8)$$

其中  $W_r \in R^{t \times d}$ ， $\{W_i \in R^{d \times d}\}_{i=1}^8$ ， $\{v_i \in R^d\}_{i=1}^4$  为可学习参数， $t$  为关系类型总数 (包括一个特殊的关系类型  $\circ$  表示没有三元组)， $l$  为句子长度， $H_e$  为 BERT 模型的输出。

### 3.4 二分匹配损失

训练的主要困难是将预测的三元组相对于真实值进行打分。采用交叉熵损失函数来衡量两个集合之间的差异是不合适的，因为交叉熵损失对预测的排列很敏感。受运筹学中指派问题的启发，论文提出了一个集合预测损失，它可以在预测三元组和真实三元组之间产生一个最优的二分匹配。

1) 用  $Y = \{Y_i\}_{i=1}^n$  表示真实三元组的集合,  $\hat{Y} = \{\hat{Y}_i\}_{i=1}^m$  表示  $m$  个预测三元组的集合, 其中  $m > n$ 。真值集合中的每个元素  $i$  看成  $Y_i = (r_i, s_i^{start}, s_i^{end}, o_i^{start}, o_i^{end})$ , 其中  $r_i$  为目标关系类型 (这可能是:  $\odot$ ),  $s_i^{start}, s_i^{end}, o_i^{start}, o_i^{end}$  分别为主体  $s$  或客体  $o$  的起止下标。预测三元组集合中的元素  $i$  记为  $\hat{Y}_i = (p_i^r, p_i^{s-start}, p_i^{s-end}, p_i^{o-start}, p_i^{o-end})$ 。

2) 损失函数: 计算二分匹配损失函数的过程分为两个步骤: 寻找最优匹配和计算损失函数。寻找真实三元组集合  $Y$  与预测三元组集合之间的最优匹配:

$$\pi^* = \arg \min_{\pi \in \Pi(m)} \sum_{i=1}^m C_{match}(Y_i, \hat{Y}_{\pi(i)}) \quad (9)$$

式中:  $\Pi(m)$  为所有  $m$  长排列的空间。  $C_{match}(Y_i, (\hat{Y})_{\pi(i)})$  为真值  $Y_i$  与指标为  $\pi(i)$  的预测三元组之间的成对匹配代价。综合关系类型的预测和实体跨度的预测, 论文定义  $C_{match}(Y_i, (\hat{Y})_{\pi(i)})$  为:

$$C_{match}(Y_i, \hat{Y}_{\pi(i)}) = -l_{r_i \odot} [p_{\pi(i)}^r(r_i) + p_{\pi(i)}^{s-start}(s_i^{start} + p_{\pi(i)}^{s-end}(s_i^{end} + p_{\pi(i)}^{o-start}(o_i^{start} + p_{\pi(i)}^{o-end}(o_i^{end})))] \quad (10)$$

将最优分配  $\pi$  建模为运筹学中的分配问题。可以把基本真理  $Y$  的集合看作是人的集合, 即集合对预测的三元组  $(\hat{Y})$  为一组作业)。用  $Y_j$  (工作  $j$ ) 分配  $(\hat{Y})_j$  (人) 的成本定义为  $C_{match}(Y_i, (\hat{Y})_{\pi(i)})$ 。总成本最小的最优分配可以通过经典的匈牙利算法计算, 时间复杂度为  $O(m^3)$ 。第二步是计算上一步中所有匹配对的损失函数。定义损失如下:

$$L(Y, \hat{Y}) = \sum_{i=1}^m \{-\log p_{\pi(i)}^r(r_i) + C_{match}(Y_i, \hat{Y}_{\pi(i)})\} \quad (11)$$

## 4 复现细节

### 4.1 与已有开源代码对比

论文提出的模型有在 github 开源代码。论文对于二分匹配损失函数使用匈牙利算法, 复现时采用 Kuhn-Munkres 算法。它通过构建增广路径来寻找最佳匹配, 是匈牙利算法的改进版。Adam 算法使用自适应学习率, 但在某些情况下, 学习率可能过大或过小, 导致训练不稳定或收敛速度慢。因此, 采用学习率衰减策略进行优化。

### 4.2 实验环境搭建

Python: 3.7

PyTorch:  $\geq 1.5.0$

Transformers: 2.6.0

### 4.3 界面分析与使用说明

采用命令行进行训练。python main.py 通过更改命令行指令如:-bert\_directory 和-max\_epoch 等修改 bert 预训练参数与最大训练次数。命令示例如下:

```
python -m main-bert_directory BERT_DIR
        -num_generated_triples 15
        -na_rel_coef 1
        -max_grad_norm 1
        -max_epoch 100
        -max_span_length 10
```

### 4.4 创新点

(1) 使用 Kuhn-Munkres 算法替换匈牙利算法。以解决二分匹配损失函数。在二分匹配问题中, 传统的匈牙利算法在处理大规模数据时存在计算复杂度高的问题。而 Kuhn-Munkres 算法, 也称为匈牙利算法的改进版, 通过优化匹配过程, 减少了计算时间, 并且能够处理更大规模的数据。通过将 Kuhn-Munkres 算法应用于二分匹配损失函数, 可以提高模型的训练效率和准确性。这种改进使得模型在处理大规模数据时更加高效和可靠, 同时也提升了模型对于复杂任务的处理能力。

(2) 使用学习率衰减策略。学习率衰减是一种常用的优化策略, 通过在训练过程中逐渐减小学习率, 可以使模型更好地收敛到最优解。在训练初期使用较大的学习率可以加快收敛速度, 而在训练后期逐渐减小学习率可以细化模型的调整, 提高模型的泛化能力。

## 5 实验结果分析

实验中使用 BERT 的 cased base 版本作为句子编码器, 该编码器包含 110M 个参数。BERT 的初始学习率设置为 0.00001, 非自回归解码器的初始学习设置为 0.00002。非自回归解码器中堆叠的双向 transformer 的数量设置为 3。使用 Dropout 策略来缓解过拟合, 其中 Dropout 率设置为 0.1。同时, 应用梯度裁剪来防止梯度爆炸。通过 AdamW 更新规则, 在小批量上通过随机梯度下降最小化定义的损失函数来训练 SPN。评价指标设置为准确率, 召回率, F1 分数。实验在 RTX 3090 上进行。



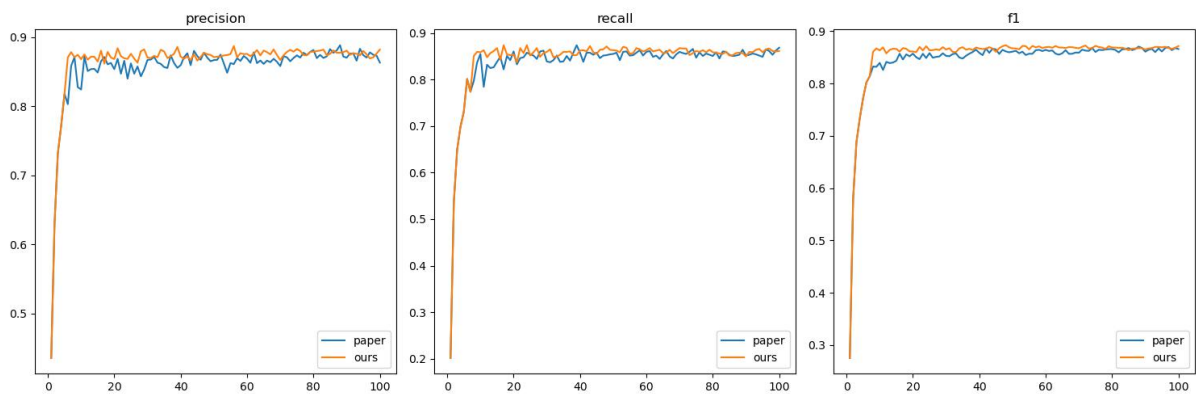


图 2. 实验结果示意

如图 2 所示，可以看出，使用学习率衰减策略后，模型收敛更快。从准确率、召回率、f1 分数等指标可以看出，数值在 90% 左右徘徊，说明模型效果较为不错。

## 6 总结与展望

Transformer 模型在句子中三元组提取任务中展现出了强大的能力。通过自注意力机制和多头注意力机制，Transformer 能够捕捉句子中的上下文信息，并将其应用于三元组提取任务中。Transformer 模型的编码器部分能够有效地对输入句子进行编码，提取句子中的关键信息。而解码器部分则可以生成与输入句子相关的三元组。

Transformer 模型在三元组提取任务中的优势在于：1. 上下文建模能力：Transformer 模型能够对句子中的上下文信息进行全局建模，不仅仅关注局部的词语，而是将整个句子作为一个整体进行处理，从而更好地理解句子的语义和结构。2. 多头注意力机制：Transformer 模型通过多头注意力机制，能够同时关注不同位置和不同方面的信息，从而更好地捕捉句子中的关联性和重要性。3. 长距离依赖建模：由于 Transformer 模型中的自注意力机制，它能够有效地处理长距离的依赖关系，不受传统循环神经网络中梯度消失或梯度爆炸的问题影响。

尽管 Transformer 模型在句子中三元组提取任务中表现出了很好的性能，但仍然存在一些挑战和改进的空间。一些可能的改进方向包括：a. 数据集规模和质量：更大规模、更丰富多样的数据集可以帮助 Transformer 模型更好地学习句子中的语义和结构信息，提高三元组提取的准确性和泛化能力。b. 预训练模型：通过预训练模型，可以在大规模的语料库上进行预训练，从而提取更丰富的语义信息，并在三元组提取任务中进行微调，进一步提升性能。c. 结合其他技术：结合其他自然语言处理技术，如实体识别、关系抽取等，可以进一步提高三元组提取的效果，使得模型能够更好地理解句子中的实体和关系。

综上所述，Transformer 模型在句子中三元组提取任务中具有很大的潜力，通过不断的改进和优化，可以进一步提高其性能，为自然语言处理任务带来更好的效果。

## 参考文献

- [1] Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. Kernel methods for relation extraction. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Lan-*

- guage Processing (EMNLP 2002)*, pages 71–78. Association for Computational Linguistics, July 2002.
- [2] Qi Li and Heng Ji. Incremental joint extraction of entity mentions and relations. In Kristina Toutanova and Hua Wu, editors, *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 402–412, Baltimore, Maryland, June 2014. Association for Computational Linguistics.
  - [3] Makoto Miwa and Mohit Bansal. End-to-end relation extraction using lstms on sequences and tree structures, 2016.
  - [4] Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. Joint extraction of entities and relations based on a novel tagging scheme. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1227–1236, Vancouver, Canada, July 2017. Association for Computational Linguistics.
  - [5] Daojian Zeng, Ranran Haoran Zhang, and Qianying Liu. Copymtl: Copy mechanism for joint extraction of entities and relations with multi-task learning, 2020.
  - [6] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks, 2014.
  - [7] Yue Yuan, Xiaofei Zhou, Shirui Pan, Qiannan Zhu, Zeliang Song, and Li Guo. A relation-specific attention network for joint entity and relation extraction. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pages 4054–4060, United States of America, 2020. Association for the Advancement of Artificial Intelligence (AAAI). International Joint Conference on Artificial Intelligence-Pacific Rim International Conference on Artificial Intelligence 2020, IJCAI-PRICAI 2020 ; Conference date: 07-01-2021 Through 15-01-2021.
  - [8] Zhiwei Yang, Jing Ma, Hechang Chen, Jiawei Zhang, and Yi Chang. Context-aware attentive multilevel feature fusion for named entity recognition. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–12, 2022.
  - [9] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. Relation classification via convolutional deep neural network. In Junichi Tsujii and Jan Hajic, editors, *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics.
  - [10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.



- [11] Xiang Kong, Zhisong Zhang, and Eduard Hovy. Incorporating a local translation mechanism into non-autoregressive translation. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1067–1073, Online, November 2020. Association for Computational Linguistics.
- [12] Zhuyu Yao, Jiangbo Ai, Boxun Li, and Chi Zhang. Efficient detr: Improving end-to-end object detector with dense prior, 2021.