文章编号：1003-0077(2021)11-0080-11

# 基于 BERT 与柱搜索的中文释义生成

范齐楠[1]，孔存良[1]，杨麟儿[1,2]，杨尔弘[2]

(1. 北京语言大学 信息科学学院，北京 100083；
2. 北京语言大学 语言资源高精尖创新中心，北京 100083)

**摘 要**：释义生成任务是指为一个目标词生成相应的释义。该文在中文释义生成任务中使用了目标词的上下文信息，并提出了一个基于 BERT 与柱搜索的释义生成模型。该文构建了包含上下文的 CWN 中文数据集，同时也在 Oxford 英文数据集上开展了实验。实验结果显示，该文模型在中英文数据集上性能均有显著提升，其中 CWN 数据集实验结果相比基线模型 BLEU 指标提升了 10.47，语义相似度指标提升了 0.105。语义相似度指标与人工评价结果相关性更高。最后，该文分析了中文释义生成任务仍存在的四个问题。

**关键词**：中文释义生成；BERT；柱搜索

**中图分类号**：TP391 **文献标识码**：A

## Chinese Definition Modeling Based on BERT and Beam Search

FAN Qinan[1], KONG Cunliang[1], YANG Liner[1,2], YANG Erhong[2]

(1. School of Information Science, Beijing Language and Culture University, Beijing 100083, China;
2. Advanced Innovaion Center for Language Resources, Beijing
Language and Culture University, Beijing 100083, China)

**Abstract**: Definition modeling task refers to generate a corresponding definition for a target word. This paper introduces the context information of the target word and proposes a definition generation model based on BERT and beam search. A CWN Chinese definition modeling dataset is constructed with context of the target word. Experiments on this Chinese dataset and the English Oxford dataset show that the model achieves significant improvements in both dataset. Especially in CWN dataset, compared with the baseline model, the BLEU score is improved by 10.47, and the semantic similarity is improved by 0.105.

**Keywords**: Chinese definition modeling; BERT; beam search

## 0 引言

释义生成(Definition Modeling)任务又称释义建模，是由 Noraset 等人首次提出，任务目的是为一个给定的目标词生成相应的释义[1]。释义生成任务不论在自然语言处理(Natural Language Processing，NLP)领域还是实际应用场景中，都具有非常重要的研究意义和价值。在 NLP 领域：①预训练的静态词向量经常会被用来初始化词嵌入，其质量好坏会对所做任务产生很大影响。目前常用的预训练词向量的质量评价方法有相似性、类比推理等，相比于这些评价方法，为预训练的词向量生成一句文本释义，能够更直观地反映词向量的质量。②低维密集词向量的可解释性问题一直是深度学习领域关注的焦点。以人类可读的形式为低维词向量生成文本释义，可以对词向量捕获到的语义信息予以解释。③词典释义经常被作为外部语义知识融入其他 NLP 任务中，本任务可以极大地丰富词典释义资源。在实际应用中，释义生成任务也可以为词典编撰者及语言学习者提供很大帮助：①不论是编撰新词典还是修订已有词典，都需要耗费大量的人力和物力，而释义生成系统可以作为词典编

**收稿日期**：2021-02-22 **定稿日期**：2021-03-29

**基金项目**：北京语言大学研究生创新基金(中央高校基本科研业务费专项资金)(20YCX139)；北京语言大学语言资源高精尖创新中心项目(TYZ19005)；国家语委信息化项目(ZDI135-105)

著者强有力的辅助工具，节省编撰成本。②对于语言学习者，当他们需要查询陌生词汇时，受限于词典的收录能力，查询不到词语的情况时有发生。当遇到多义词时，他们也只能根据上下文去推断应取哪个义项，往往不能保证准确性。而释义生成任务不仅可以为新词语生成释义，也可以通过融合上下文的方法生成词语在特定语境下的释义。

Noraset 等人最早在英文上研究释义生成任务，出于评价预训练词向量质量的目的，这项工作使用目标词的预训练词向量作为输入来生成释义，根据生成释义是否准确来验证词向量是否包含正确的语义信息。考虑到预训练词向量会将多义词的多个义项合并的问题，Gadetsky 等人借鉴语义消歧任务，采用非参数贝叶斯的方法实现了动态多义，训练模型生成目标词语在给定上下文中的释义[2]。Ishiwatari 等人后来将目标词预训练词向量和上下文向量直接拼接后用于释义生成，该方法达到了目前英文释义生成任务的最优结果[3]。以往研究证明，上下文信息不仅可以对目标词进行消歧，也可以补充更多的语义信息，在释义生成任务中起到了非常重要的作用。在中文上，Yang 等人首次开展了释义生成任务研究，将 HowNet 中的义原作为外部语义知识融入模型来提升生成效果，但没有考虑目标词的上下文信息[4]。

基于上述问题，本文首次将目标词的上下文引入中文释义生成任务，将任务重新定义为给定一个目标词及其所在上下文，为其生成相应的释义，图 1 中给出了数据示例。

| **被释义词**：意外 | | |
|---|---|---|
| 上下文： | 1. 好在我们都已买了保险，如果发生**意外**，一切都由保险公司理赔。 | 2. 我亲口告诉她实情，令我**意外**的是，她出奇的平静，似乎早知这一刻。 |
| **释义**： | 料想不到的事件，指不幸的灾难变故。 | 形容人感到惊讶。 |

图 1　中文释义生成数据示例

由于词典资源的获取难度较高，且词典本身的容量有限，释义生成任务缺乏供模型训练的大量数据，属于低资源的文本生成任务。相较于前人工作中普遍使用的 LSTM 模型，参数更多、性能更好的模型（如 Transformer）难以在释义生成任务上得到充分训练，因此无法取得很好的效果。使用预训练语言模型是解决这一问题的有效方法，可以将预训练语言模型在大规模语料上训练获得的先验知识迁移到释义生成任务中。因此，本文提出了基于预训练语言模型 BERT 与柱搜索的释义生成模型。如图 2 所示，该模型采用编码器-解码器框架，将预训练的 BERT[5] 作为模型编码器，用于对目标词及上下文直接拼接后的序列进行编码，将 Transformer[6] 作为模型解码器，用于生成释义。在测试阶段，为缓解陷入局部最优解的问题，我们将前人使用的贪心搜索（Greedy Search）策略替换为柱搜索（Beam Search）策略来扩大搜索空间，以兼顾模型解码的效率和性能，此策略进一步提升了释义生成效果。

为了验证模型的有效性，本文基于中文词汇网络（Chinese WordNet，CWN）构建了新的中文释义生成数据集。与 Yang 等人使用的数据集不同，CWN 数据集中每条数据包含被释义词、上下文及释义三项内容，而 Yang 等人使用的数据集仅包含

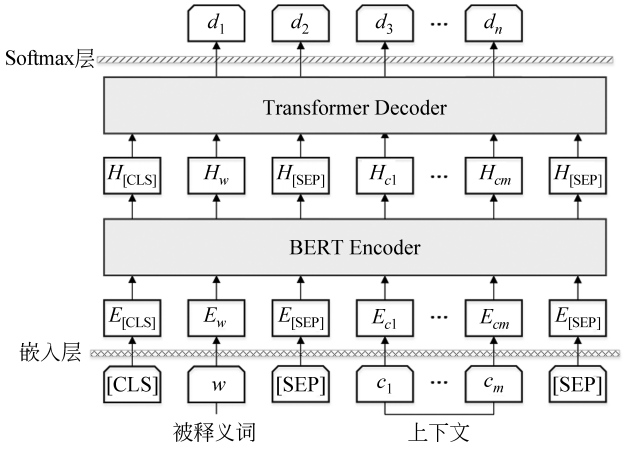



图 2　模型图

被释义词和释义。除了 BLEU 指标之外，本文采用语义相似度作为额外的评价指标，该指标使用余弦相似度计算生成释义和参考答案句向量在语义层面上的相似程度。本文提出的模型在中文 CWN 数据集上的实验结果相比基线模型提升显著，在 Gadetsky 等人构建的 Oxford 英文数据集上实验结果同样明显超出基线模型。另外，我们对本模型及基线模型在 CWN 数据集上的生成结果进行了人工评价，评价结果也与实验结果一致，表明了本文所提出方法的有效性。最后，本文分析了数据分布情况

对释义生成结果的影响，并对模型的生成结果进行了实例分析。

本文的主要贡献有：

(1) 首次在中文释义生成任务中使用了目标词的上下文，更完整地定义了中文释义生成任务。

(2) 提出了基于 BERT 与柱搜索策略的释义生成模型，有效弥补数据量不足的缺陷，获得了较好的效果。

(3) 对本文模型生成结果进行了深入分析，总结了中文释义生成任务仍待解决的四个问题。

## 1 融合上下文的中文释义生成模型

本文提出的中文释义生成任务，指的是生成目标词在特定上下文中的释义。如图 1 给出的数据示例，当给定相同词、不同上下文时，模型生成的释义也不同。形式化地，即给定一个词语 $w$，以及包含该词语的上下文 $C=[c_1,\cdots,c_m]$，为其生成一句相应的释义 $D=[d_1,\cdots,d_n]$。模型的生成过程可以用条件概率表示如式(1)所示。

$$P(D \mid w,C)=\prod_{i=1}^{n} p(d_i \mid d_{<i},w,C) \quad (1)$$

为了弥补缺乏训练数据的问题，本文在 Transformer 模型的基础上，提出了基于预训练语言模型 BERT 和柱搜索策略的模型，整体模型架构如图 2 所示。该模型使用 BERT 初始化编码器参数，使用 Transformer 作为模型解码器，然后在释义生成任务上进行微调，本节将对该模型进行详细介绍。

### 1.1 BERT 编码器

由于 Transformer 模型的参数量庞大，需要借助大规模数据进行参数训练，而中文释义生成属于低资源任务，数据量远远未达到训练要求，因此难以达到理想效果。将预训练语言模型迁移到低资源任务上，是弥补数据量不足的有效方法。BERT 是在大规模无标注语料上预训练的基于 Transformer 的多层双向编码器，近两年被应用于多项 NLP 任务中并刷新了最佳成绩。基于此，本文将 BERT 作为模型编码器，使得模型能够获得 BERT 从大规模语料中学到的先验知识。

本文将目标词 $w$ 和上下文 $C$ 直接拼接后作为输入序列。在嵌入层，本文通过两种方式将目标词和上下文区分开。首先，使用特殊符号“[SEP]”将它们分隔开。其次，为它们分别加上不同的段表征，将目标词的段表征置为 0，上下文的段表征置为 1。如图 3 所示，对于每一个词，其词嵌入由对应的词表征(Token Embedding)、段表征(Segment Embedding)和位置表征(Position Embedding)相加产生。经过 BERT 编码后得到最终的序列表征 $H$，如式(2)所示。

$$H=\mathrm{BERT}([\mathrm{CLS}]\circ w\circ[\mathrm{SEP}]\circ C\circ[\mathrm{SEP}]) \quad (2)$$

其中，“∘”表示连接操作，$H$ 由整个序列的上下文相关词向量构成，例如，$H_0$ 是特殊符号“[CLS]”的词向量。$H$ 即为编码器的输出，传给 Transformer 解码器用于解码。

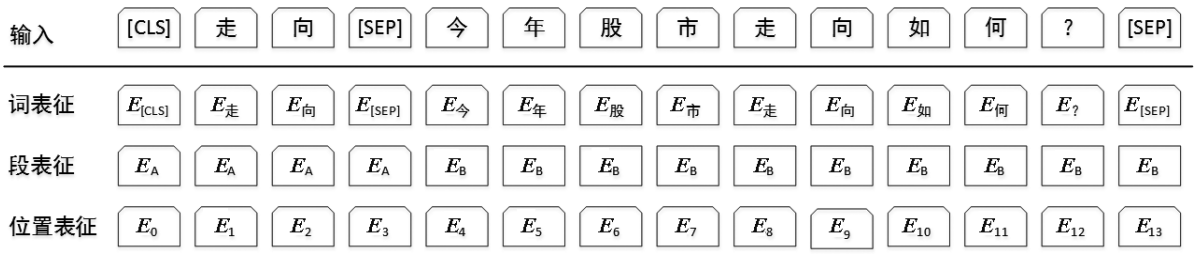



图 3 BERT 嵌入层

### 1.2 Transformer 解码器

Transformer 模型是基于多头注意力机制的序列生成模型，近年来被广泛应用于 NLP 文本生成任务中。该模型的解码器是根据上一时间步的输出预测当前时间步的输出，最后将每个时间步输出的词语拼接起来得到最终的生成序列。

在本任务中，模型首先将之前时间步生成的释义序列通过嵌入层编码后再加上词的位置表征，将得到的词嵌入作为 Transformer 解码器的输入。Transformer 解码器由 $N$ 层相同的模块构成，上层模块输出的隐状态是下层模块的输入。每个模块包含三个子层：一个掩码多头自注意力层、一个编码器-解码器多头注意力层和一个前馈神经网络层。

其中，多头注意力层由多个注意力层得到的向量拼接而成，每个注意力层采用缩放点积运算，如式(3)、式(4)所示。

$$\mathrm{Attention}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})=\mathrm{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{d_k}}\right)V \tag{3}$$

$$\mathrm{MultiHead}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})=\mathrm{Concat}(\mathrm{Attention}_1,\mathrm{Attention}_2,\cdots,\mathrm{Attention}_h) \tag{4}$$

其中，$\boldsymbol{Q}$、$\boldsymbol{K}$ 和 $\boldsymbol{V}$ 分别表示查询矩阵(Query)、键值矩阵(Key)和实值矩阵(Value)，$h$ 表示注意力层的头数。掩码多头自注意力层的 $\boldsymbol{Q}$、$\boldsymbol{K}$ 和 $\boldsymbol{V}$ 相同，都是释义的词嵌入经线性映射后的向量，掩码操作使模型在训练阶段的每个时间步不能看到未来信息。编码器-解码器多头注意力层的 $\boldsymbol{Q}$ 来自于上一层解码器的输出，$\boldsymbol{K}$ 和 $\boldsymbol{V}$ 来自于编码器的输出。另外，这三个子层之后都会接一个归一化层和残差网络，归一化层能够加快模型训练速度，残差网络能够防止神经网络模型退化。

### 1.3　柱搜索策略

在解码阶段，Seq2seq 模型常用的搜索算法有贪心算法和柱搜索算法。在释义生成任务中，前人都选用了贪心算法，但该算法有一些弊端。在每个时间步都选取概率最大的词，很容易陷入局部最优解。另外，当某个时间步概率最大词错误时，该错误也会被继续传播。

柱搜索是一种平衡性能和消耗的搜索算法，目的是解码出相对较优的序列，能够在一定程度上缓解贪心算法的上述问题。因此本文采取了柱搜索策略，与贪心算法在解码的每个时间步都选择概率最大的词不同，柱搜索算法会结合之前时间步已生成的序列，在当前时间步选择使得整体序列概率最大的前 $K$ 个词，最后将 $K$ 个序列中概率最大的词作为最终输出，相比贪心算法能够进一步提升生成效果。

## 2　实验

### 2.1　数据集

不论在英文上还是中文上，词典语料都非常稀缺。目前在中文上，释义生成任务还没有同时包含词语、上下文及释义的数据集。中文词汇网络(CWN)[①]是一个由台湾“中研院”开发的词汇语义关系知识库，该知识库为大部分义项都配备了多条例句，我们选用 CWN 构建了高质量的中文释义生成数据集。本文使用 opencc-python 工具[②]将数据由繁体中文转换为简体，使用 jieba 工具[③]对全部数据进行分词，并对其中的特殊字符等做了预处理。然后按照被释义词数量 8∶1∶1 的比例，将数据集切分为训练集、验证集和测试集，最终每条数据包含一个被释义词、一条上下文和相应的释义。本文在 Oxford 英文数据集上也开展了实验，此数据集由 Gadetsky 等人通过牛津在线词典[④]提供的 API 构建。CWN 及 Oxford 数据集的规模统计如表 1 所示，其中，上下文长度和释义长度是平均长度，中文 CWN 数据集按字统计，英文 Oxford 数据集按词统计。

**表 1　数据集规模统计**

| 数据集 | | 被释义词数量 | 释义数量 | 上下文数量 | 平均上下文长度 | 平均释义长度 |
|---|---|---|---|---|---|---|
| CWN | 训练集 | 6 574 | 21 736 | 67 861 | 21.60 | 9.10 |
| | 验证集 | 823 | 2 606 | 8 082 | 21.77 | 9.03 |
| | 测试集 | 824 | 2 774 | 8 599 | 21.33 | 9.08 |
| Oxford | 训练集 | 33 128 | 97 780 | 97 855 | 17.74 | 11.02 |
| | 验证集 | 8 867 | 12 230 | 12 232 | 17.80 | 10.99 |
| | 测试集 | 8 850 | 12 230 | 12 232 | 17.56 | 10.95 |

① https://lope.linguistics.ntu.edu.tw/cwn2/
② https://github.com/yichen0831/opencc-python
③ https://github.com/fxsjy/jieba
④ https://en.oxforddictionaries.com/

另外，由于 CWN 数据集具有多上下文的特点，本文对 CWN 切分后的数据集每条释义对应的上下文数量做了统计。如表 2 所示，在三个数据集中，上下文数量分布情况非常类似，超过 90%的释义都有两条以上的上下文，有 3 条上下文的释义最多，达到 60%以上。Oxford 数据集中几乎全部的释义都只有 1 条对应上下文，相比之下，CWN 数据集的上下文资源更加丰富。

**表 2 CWN 数据集释义包含的上下文数量统计**

| 数据集 | | 上下文数量 | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7+ |
| 训练集 | 释义数量 | 794 | 3 342 | 13 671 | 1896.00 | 768.00 | 1 063 | 202 |
| | 占比/% | 3.65 | 15.38 | 62.90 | 8.72 | 3.53 | 4.89 | 0.93 |
| 验证集 | 释义数量 | 78 | 424 | 1 671 | 202.00 | 88.00 | 122 | 21 |
| | 占比/% | 2.99 | 16.27 | 64.12 | 7.75 | 3.38 | 4.68 | 0.81 |
| 测试集 | 释义数量 | 111 | 408 | 1 777 | 229.00 | 96.00 | 134 | 19 |
| | 占比/% | 4.00 | 14.71 | 64.06 | 8.26 | 3.46 | 4.83 | 0.68 |

## 2.2 基线模型

本文将 Transformer 模型和 LOG-CaD 模型[3]作为基线模型。Transformer 模型是基于多头自注意力机制的模型，近年来在文本生成任务中被广泛应用，本文不再做详细介绍。LOG-CaD 模型是针对英文释义生成任务提出的模型，该模型在四个英文数据集上都取得了不错的结果。LOG-CaD 模型基于编码器-解码器框架，其中编码器共包含三个部分。

**局部上下文编码器**：局部上下文是指给定的一句包含目标词的上下文。该模型采用双向 LSTM 模型对局部上下文进行编码。在解码的每个时间步，都通过注意力机制计算当前隐状态和局部上下文每个时间步隐状态的注意力系数，加权后得到最终的局部上下文向量表示。

**全局上下文编码器**：全局上下文是指从大规模语料中获得的全局语义信息。CBOW 是使用 Google 新闻语料预训练的静态词向量，该模型从 CBOW 中提取出目标词的预训练词向量作为目标词的全局上下文表示。

**目标词字符级特征提取器**：由于英文单词中的词缀可以体现出重要的词义信息，例如，以“-ist”结尾的通常是名词，表示专家或从事某活动的人。因此，该模型采用 CNN 模型提取目标词的字符级特征表示，用于获取词缀中包含的语义信息。

模型将上述三个编码器的输出拼接后作为解码器的输入。该模型的解码器采用了单向 LSTM 模型，并在每个时间步增加了门控机制，对当前时间步输出的隐状态和编码器输出的拼接向量进行过滤，以更好地控制多种输入信息之间的交互。

## 2.3 实验设置

本文的 Transformer 模型基于 FAIR 开源代码库①实现，使用预训练的中文词向量[7]和 fastText 词向量[8]分别对中文和英文数据集的词嵌入进行初始化，词表维数为 300，解码器的输入和输出词嵌入矩阵共享权重。模型的编码器和解码器均设置为 6 层，其中多头注意力层有 5 个注意力头，前馈层维度为 2 048。训练过程使用 Adam 优化器[9]更新模型参数，初始学习率为 1e-7，增长到 5e-4 后逐步下降，dropout 设置为 0.3。

本文基于 BERT 的模型采用的是 base 版本的 BERT 预训练模型，在 Transformers 开源代码库[10]基础上实现。本文的模型训练分为两个阶段：第一阶段固定编码器参数，仅训练解码器，学习率设置为 5e-4，warm-up 设置为 4 000；第二阶段同时微调编码器和解码器，学习率设置为 2e-5，warm-up 设置为 2 000。两阶段的 dropout 均设置为 0.2。中文和英文释义的词嵌入用与上述相同的预训练词向量进行了初始化，Transformer 解码器的超参数设置也与上述一致，优化器同样使用 Adam。另外，在选择最优模型时采取了 early-stop 策略，每轮模型都会在验证集上计算 PPL 和 BLEU 值(考虑到效率

① https://github.com/pytorch/fairseq

问题，这里使用 NLTK translate 包[①]计算 sentence BLEU，与测试时的 BLEU 指标不同但高度相关），当验证集上 PPL 超过 10 轮不再增长时，取这 10 轮中 BLEU 值最高的模型保存下来用于测试。

### 2.4　实验结果

本文分别在 CWN 中文数据集和 Oxford 英文数据集上评测了模型效果。由于前人使用的 BLEU[11] 评价指标只能衡量生成释义与参考答案在字面上的相似性，因此本文将语义相似度作为额外的评价指标，从语义层面衡量生成释义和参考答案的相似性。该指标的计算方法是，首先使用 sentence-transformers 工具[②]分别对生成释义和参考答案句子进行编码[12-13]，然后使用 scipy 包[③]计算两个句向量的余弦相似度。表 3 列出了 BLEU 和语义相似度两个指标的实验结果。其中，Transformer 和 LOG-CaD 为本文的基线模型，ESD-sem 为 Li 等人提出的基于显式语义分解的模型[14]。BERT-fix-encoder 表示训练的第一阶段固定编码器参数仅训练解码器，BERT-fine-tune 表示第二阶段同时微调编码器和解码器，这两个模型解码时均使用贪心算法。

**表 3　实验结果**

| 模型 | BLEU 指标 | | | | 语义相似度指标 | | | |
|---|---|---|---|---|---|---|---|---|
| | CWN | | Oxford | | CWN | | Oxford | |
| | 验证集 | 测试集 | 验证集 | 测试集 | 验证集 | 测试集 | 验证集 | 测试集 |
| Transformer | 21.16 | 20.77 | 17.03 | 17.02 | 0.273 | 0.269 | 0.369 | 0.368 |
| LOG-CaD | 30.76 | 29.58 | 19.13 | 18.95 | 0.362 | 0.415 | 0.269 | 0.306 |
| ESD-sem | — | — | — | 20.86 | — | — | — | — |
| BERT-fix-encoder (Greedy) | 38.96 | 37.25 | 19.87 | 20.14 | 0.508 | 0.486 | 0.443 | 0.443 |
| BERT-fine-tune (Greedy) | **43.25** | **40.05** | **21.95** | **22.01** | **0.538** | **0.520** | **0.473** | **0.459** |

可以看到，Transformer 模型在 CWN 中文数据集上表现欠佳，BLEU 和语义相似度两个指标均与 LOG-CaD 模型有较大差距。在 Oxford 英文数据集上，Transformer 模型的 BLEU 值与 LOG-CaD 模型差距不大，语义相似度甚至超过了 LOG-CaD 模型。有了 BERT 的加持后，本文提出的 BERT-fix-encoder(Greedy)模型在两个数据集上的指标值都得到了显著提升，经过第二阶段微调后的模型比起第一阶段也均有一定提升，验证了本文模型和两阶段训练策略的有效性。

本文在 BERT-fine-tune (Greedy) 模型基础上，将贪心算法改进为柱搜索算法，对柱取 2～12 不同大小时的 BERT-fine-tune 模型结果进行了对比实验，结果如图 4 所示，在 CWN 中文数据集上，当柱取值较小时，两个评价指标都得到了提升，但继续增加柱的大小甚至会导致指标值低于贪心算法。在 Oxford 英文数据集上，柱搜索策略带来的性能提升更明显，但随着柱的增大也会出现指标值下降的情况。针对这一现象，Cohen 和 Beck 指出，柱搜索算法的柱取值越大，在解码过程较靠前的时间步会越倾向于选择低概率的词语，对生成效果产生影响，因此一味增加柱的大小并不能带来持续的性能提升[15]。

## 3　质量分析

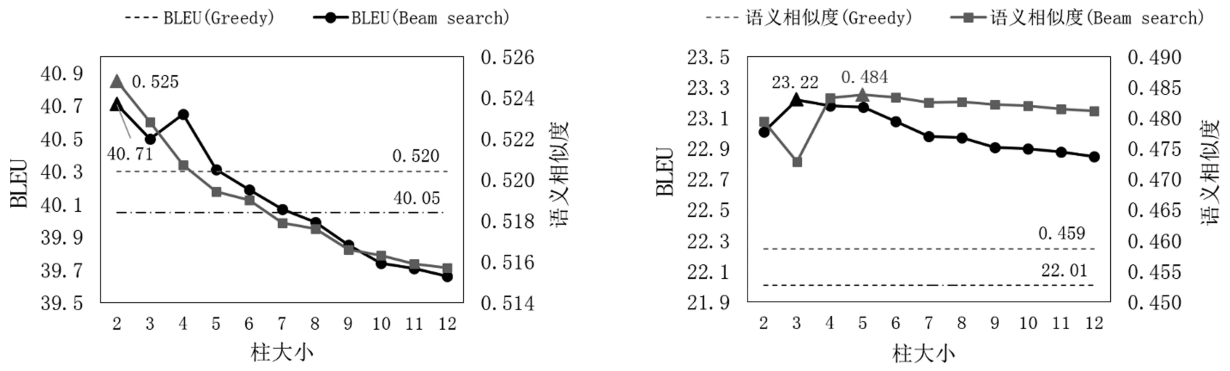
### 3.1　人工评价

为了更准确地评价生成释义的质量，本文从 CWN 测试集中随机采样了 200 条数据，其中被释义词没有重复，然后采用人工评价的方式对基线模型和本文模型的生成释义进行了质量评估。我们邀请了四名语言学专业学生作为标注员，使用 Likert 量表[16]，按照 1～5 五个等级让标注员分别从语法和语义两个角度对模型的生成释义进行独立评分。其中语法角度仅衡量生成释义是否符合语法规则，完全符合为 5 分，完全不符合为 1 分；语义角度衡量生成释义与参考答案表示的语义是否一致，完全一致为 5 分，完全不一致为 1 分。表 4 展示了四名标注员的人工评价结果。

① https://www.nltk.org/_modules/nltk/translate/bleu_score.html

② https://github.com/UKPLab/sentence-transformers

③ https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.cdist.html




(a) CWN数据集 (b) Oxford数据集

图 4 柱取不同大小的结果对比

**表 4 CWN 数据集人工评价结果**

| | 模型 | 标注员 | | | | 平均分 |
|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | |
| 语法 | Transformer | 4.985 | 4.890 | 3.905 | 4.760 | 4.635 |
| | LOG-CaD | 4.890 | 4.390 | 3.785 | 4.450 | 4.379 |
| | BERT-fix-encoder (Greedy) | 5.000 | 4.830 | 4.320 | 4.840 | 4.748 |
| | BERT-fine-tune (Greedy) | 5.000 | 4.920 | 4.525 | 4.905 | 4.838 |
| | BERT-fine-tune (Beam=2) | 5.000 | 4.930 | 4.615 | 4.915 | 4.865 |
| 语义 | Transformer | 1.575 | 1.605 | 1.815 | 1.435 | 1.608 |
| | LOG-CaD | 2.425 | 2.220 | 2.545 | 2.000 | 2.298 |
| | BERT-fix-encoder (Greedy) | 2.945 | 2.740 | 3.210 | 2.755 | 2.913 |
| | BERT-fine-tune (Greedy) | 3.315 | 2.955 | 3.615 | 3.165 | 3.263 |
| | BERT-fine-tune (Beam=2) | 3.340 | 3.060 | 3.735 | 3.165 | 3.325 |

可以看到,四名标注员对模型生成释义语法的评分都普遍较高,本文模型语法的平均分接近满分,说明模型具备了出色的生成流畅句子的能力。而五个模型在语义上的评分都相对较低,但本文模型的评分还是显著优于基线模型,这与上节中的自动评价结果也保持了一致。

为了衡量 BLEU 和语义相似度两个自动评价指标与人工评价指标的相关程度,本文计算了自动评价指标与人工评价指标的 Pearson 相关系数,结果如表 5 所示。可以看到,相比前人使用的 BLEU 指标,本文额外使用的语义相似度指标与人工评价指标具有更强的相关性。这说明语义相似度指标的结果更接近人类评价结果,更具有参考价值。

**表 5 CWN 数据集人工评价结果**

| 自动评价 | 人工评价 | |
|---|---|---|
| | 语法 | 语义 |
| BLEU | 0.245($p<0.000\ 1$) | 0.482($p<0.000\ 1$) |
| 语义相似度 | 0.298($p<0.000\ 1$) | 0.639($p<0.000\ 1$) |

### 3.2 数据分布情况对结果的影响分析

对于人类来说,义项越多的词语推断其含义的难度越大,而上下文可以帮助我们对多义词进行消歧,上下文中被释义词的搭配也能够为我们提供更多语义信息,那么上下文在模型中同样可以得到有效利用吗?本节在 CWN 数据集上,从释义、上下文两项数据内容的不同分布情况出发,对基线模型及本文模型的生成结果进行了对比分析。由于计算时 BLEU 指标会将多义词的全部释义都作为参考答案,而这会对我们的分析结果产生影响,因此本节选用了语义相似度指标进行分析。如图 5 所示,两张子图中分别展示了不同释义数量以及上下文长度对模型语义相似度结果的影响。

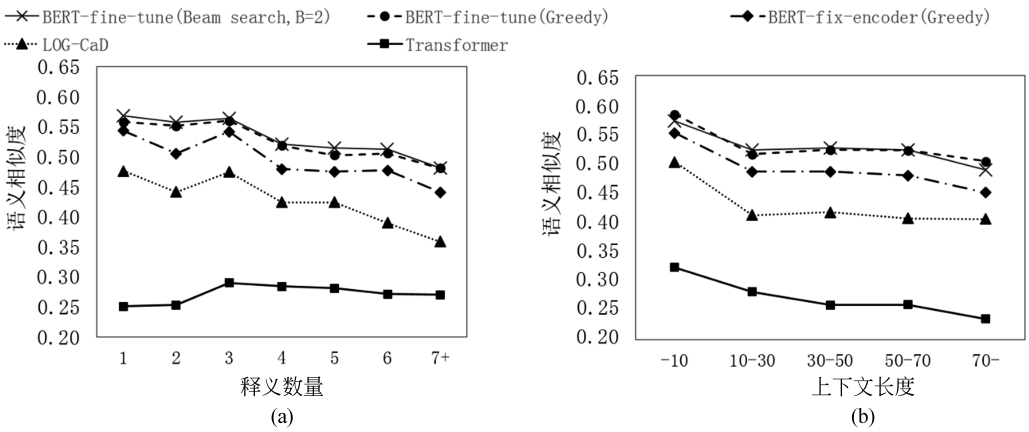



图 5　数据分布情况对语义相似度结果的影响

可以看到，前两个子图中 Transformer 模型的折线趋势与其他四个模型有明显差异，本文认为这是由于 Transformer 模型在此任务上本身生成的结果较差，由此带来的影响比本节分析的数据因素要大得多。因此，本节主要对另外四个模型进行对比分析。

图 5(a)中释义数量是指被释义词拥有的释义数量。可以看到随着释义数量增加，本文模型和 LOG-CaD 基线模型的语义相似度指标都呈现下降趋势。但当释义数量超过 6 条时，对 LOG-CaD 模型的结果影响显然更大，这可能是由于本文模型和 LOG-CaD 模型不同的编码方式造成的。本文是将被释义词和上下文同时编码，BERT 编码器能够将输入序列编码为一组上下文相关的词向量，更好地捕获上下文信息。而 LOG-CaD 模型是将上下文和被释义词各自编码后再拼接，当义项数量过多时，此方法可能难以获得满意的消歧效果。

图 5(b)中的上下文长度是按字数统计的，整体上折线都呈下降趋势。这说明当上下文长度过长时，会对模型在上下文中定位重要信息产生干扰，因此释义生成任务中使用的上下文句子不宜过长。

### 3.3　生成释义的问题分析

本文从最优模型 BERT-fine-tune(Beam＝2)在 CWN 数据集上的生成结果中发现了一些典型问题，并将问题及相应实例分类整理在表 6 中。

**表 6　生成释义的问题及相应实例**

| | |
|---|---|
| 问题一：生成相反释义 | |
| 近 | 参考答案：形容时间的距离短。 |
| | 生成结果：形容时间的距离长。 |
| 问题二：缺乏特定知识 | |
| 河南 | 参考答案：位于黄河南岸的一省。介于湖北省与陕西省之间。 |
| | 生成结果：中国省名，位于湖北、西藏之间的区域。 |
| 问题三：生成释义中包含被释义词 | |
| 解释 | 参考答案：说明特定事件的原因、理由使听话者明白。 |
| | 生成结果：解释使听话者明白。 |
| 箱 | 参考答案：计算箱装物品的单位。 |
| | 生成结果：计算箱子的单位。 |
| 问题四：生成与训练集中的近义词相同的释义 | |
| 聚集(近义词“聚”) | 参考答案：多数的前述对象同一时间在同一地点出现。 |
| | 生成结果：多数的前述对象同一时间在同一地点出现。 |

续表

| | |
|---|---|
| 施暴(近义词"施虐") | 参考答案：以暴力对待。 |
| | 生成结果：以不合人道，受事者无法忍受的方式对待。 |

第一类问题是模型生成的释义与参考答案的语义刚好相反，这一问题在英文释义生成任务中也会出现，是由于反义词的上下文语境通常极为相似，导致它们的词向量也会非常接近，这是通过大规模语料训练词向量方法的固有问题，这一问题也被转移到了释义生成任务上。

第二类问题是由于模型缺乏特定领域知识而导致生成错误释义，这一问题可以通过融入外部知识的方法得以缓解。

第三类问题是生成的释义中包含被释义词，本文认为这一问题是否归于错误不应一概而论。例如，表6中针对该问题给出的第一个实例的情况是错误的，但对于第二个实例，释义中出现被释义词应该是被允许的。

第四类问题是如果被释义词的近义词在训练集中出现过，模型会倾向于生成与该近义词完全相同的释义。这种做法有时可以帮助模型生成完全正确的释义，例如，表6中针对此问题给出的第一个实例；但有时由于近义词的语义有细微差别，也会导致生成释义不准确，例如，表6中针对此问题给出的第二个实例。

## 4 相关工作

### 4.1 释义生成任务

释义生成是近年来提出的一项文本生成任务，最初用于验证预训练静态词向量能否捕捉到正确且充分的语法、语义信息，或用于对低维密集词向量包含的语义信息予以解释，后来此任务的研究目的逐渐落地到辅助语言学习者学习新词汇的实际应用场景。目前对该任务的研究基本都在英文上开展，对于中文释义生成的研究仅有一篇文章公开发表。

Noraset等人首次提出了释义生成任务，用于直接评估预训练词向量的质量。文中将任务定义为给定目标词，为其生成相应的一句释义。方法上，除了目标词预训练词向量以外，还使用了CNN模型来提取目标词的字符级语义特征，解码器采用LSTM模型，并通过门控机制在解码的每一个时间步对输入向量进行信息过滤，但这项工作忽略了预训练词向量存在将多义词意义合并的缺陷。此后，在英文上的工作基本都使用了上下文信息，让模型生成目标词在特定上下文中的释义[1]。Gadetsky等人提出了基于AdaGram对词向量进行消歧的方法[2]。Mickus等人提出了Select和Add两种编码机制对目标词及上下文进行编码，突出目标词在上下文序列中的重要性[17]。Ishiwatari等人直接将目标词预训练词向量、字符级特征向量和上下文向量拼接起来，作为解码器输入，进一步提升了生成效果[3]。Li等人提出将词的含义明确分解为若干个语义成分，并使用离散的潜在变量对语义成分建模后用于释义生成，该模型在英文数据集上取得了当前最优BLEU结果[4]。

还有研究者从低维密集词向量的可解释性出发研究释义建模任务。Chang等人将给定的目标词及其上下文嵌入高维稀疏空间，然后从中选择最能解释目标词语义的特定维，使用RNN模型生成目标词的文本释义，该方法能够对目标词嵌入包含的语义信息进行直接解释[18]。Chang等人随后又将释义建模任务重新定义为分类任务，即根据目标词及其上下文选择最合理的释义，来研究BERT、ELMO等预训练语言模型的上下文相关词向量捕获了什么语义信息[19]。

释义生成任务在中文上的研究还很少。Yang等人首次在中文上开展释义生成研究工作，使用基于Transformer的模型，并将HowNet中的义原序列融入模型，为模型提供更多外部语义知识信息，但这项工作没有考虑上下文信息[4]。基于此，本文首次将上下文信息引入中文释义生成任务。

### 4.2 预训练语言模型BERT

近年来，面向NLP的预训练技术研究取得了长足进展。早期使用的Word2Vec预训练静态词向量能够为NLP任务带来的提升十分有限，且无法解决一词多义的问题[20-21]。后来提出的ELMo是一种上下文相关的文本表示方法，可有效处理多义词问题[22]。随后，GPT[23]和BERT等预训练语言模型被相继提出。其中，BERT是迄今为止应用范围最广、效果最佳的预训练语言模型，在文本分类、语法改错等多项NLP任务中都展示出了强大的性

能[24-25]。BERT 是基于 Transformer 的双向编码表示模型，该模型的预训练使用了掩码语言模型和后句预测两个子任务，模型的优化目标函数是两个子任务目标函数的结合。将预训练后的 BERT 迁移到文本生成任务中，只需在 BERT 后增加一个解码器，即可进行微调训练。

本文将预训练语言模型 BERT 迁移到释义生成任务中，使用 BERT 初始化编码器的模型参数，使用 Transformer 作为模型解码器，此方法有效缓解了缺乏训练数据的问题。

## 5　总结

本文首次将上下文信息应用于中文释义生成任务，为了弥补缺乏训练数据的问题，提出了基于 BERT 与柱搜索策略的模型。为了验证模型的性能，本文分别在新构建的 CWN 中文数据集以及前人构建的 Oxford 英文数据集上开展了实验，结果表明，本文模型相比基线模型能显著提升释义生成的效果。本文还分析了数据分布情况对生成结果的影响，且通过实例分析总结了目前中文释义生成仍存在的四类重要问题。未来的工作中，我们计划研究是否可以提出一种新的编码机制，更充分地利用多条上下文信息。

## 参考文献


[1] Noraset T, Liang C, Birnbaum L, et al. Definition modeling: Learning to define word embeddings in natural language[C]//Proceedings of the AAAI Conference on Artificial Intelligence, 2017.

[2] Gadetsky A, Yakubovskiy I, Vetrov D. Conditional generators of words definitions[C]//Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, 2018: 266-271.

[3] Ishiwatari S, Hayashi H, Yoshinaga N, et al. Learning to describe unknown phrases with local and global contexts[C]//Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2019: 3467-3476.

[4] Yang L, Kong C, ChenY, et al. Incorporating sememes into Chinese definition modeling[J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2020, 28: 1669-1677.

[5] Devlin J, Chang M W, Lee K, et al. BERT: Pre-training of deep bidirectional transformers for language understanding[C]//Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2019: 4171- 4186.

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

[7] Li S, Zhao Z, Hu R, et al. Analogical reasoning on Chinese morphological and semantic relations[C]//Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, 2018: 138-143.

[8] Bojanowski P, Grave E, Joulin A, et al. Enriching word vectors with subword information[C]//Proceedings of Transactions of the Association for Computational Linguistics, 2017: 135-146.

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

[10] Wolf T, Debut L, Sanh V, et al. HuggingFace's Transformers: state-of-the-art natural language processing[J]. arXiv preprint arXiv: 1910.03771, 2019.

[11] Papineni K, Roukos S, Ward T, et al. BLEU: A method for automatic evaluation of machine translation[C]//Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, 2002: 311-318.

[12] Reimers N, Gurevych I. Sentence-BERT: Sentence embeddings using Siamese BERT- Networks[C]//Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2019.

[13] Reimers N, Gurevych I. Making monolingual sentence embeddings multilingual using knowledge distillation[J]. arXiv preprint arXiv: 2004.09813, 2020.

[14] Li J, Bao Y, Huang S, et al. Explicit semantic decomposition for definition generation[C]//Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020: 708-717.

[15] Cohen E, Beck C. Empirical analysis of beam search performance degradation in neural sequence models[C]//Proceedings of the International Conference on Machine Learning, 2019: 1290-1299.

[16] Likert R. A technique for the measurement of attitudes[M]. New York: The Science Press, 1932.

[17] Mickus T, Paperno D, Constant M. Mark my word: A sequence-to-sequence approach to definition modeling[J]. arXiv preprint arXiv: 1911.05715, 2019.

[18] Chang T Y, Chi T C, Tsai S C, et al. xSense: Learning sense-separated sparse representations and textual definitions for explainable word sense networks[J].

arXiv preprint arXiv: 1809.03348, 2018.

[19] Chang T Y, Chen Y N. What does this word mean? Explaining contextualized embeddings with natural language definition[C]//Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, 2019: 6066-6072.

[20] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[J]. arXiv preprint arXiv: 1301.3781, 2013.

[21] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality[J]. arXiv preprint arXiv: 1310.4546, 2013.

[22] Peters M E, Neumann M, Iyyer M, et al. Deep contextualized word representations[C]//Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2018: 2227-2237.

[23] Radford A, Narasimhan K, Salimans T, et al. Improving Language Understanding by Generative Pre-training[OL]. https://s3-us-west-2.amazonaws.com [2019-08-16].

[24] Adhikari A, Ram A, Tang R, et al. DocBert: Bert for document classification[J]. arXiv preprint arXiv: 1904.08398, 2019.

[25] Kaneko M, Komachi M. Multi-head multi-layer attention to deep language representations for grammatical error detection[J]. arXiv preprint arXiv: 1904.07334, 2019.

范齐楠(1995—),硕士研究生,主要研究领域为自然语言处理和智能语言学习。
E-mail: blcufqn@hotmail.com

杨麟儿(1983—),通信作者,博士,副教授,主要研究领域为自然语言处理和智能语言学习。
E-mail: yangtianlin@blcu.edu.cn

孔存良(1995—),博士研究生,主要研究领域为自然语言处理和智能语言学习。
E-mail: cunliang.kong@outlook.com