基于BERT与柱搜索的中文释义生成

范齐楠 孔存良 杨麟儿 杨尔弘
北京语言大学

摘要

释义生成任务是指为一个目标词生成相应的释义。前人研究中文释义生成任务时未考虑目标词的上下文，本文首次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。本文构建了包含上下文的CWN中文数据集用于开展实验，除了BLEU指标之外，还使用语义相似度作为额外的自动评价指标，实验结果显示本文模型在中文CWN数据集和英文Oxford数据集上均有显著提升，人工评价结果也与自动评价结果一致。最后，本文对生成实例进行了深入分析。

关键词： 中文释义生成 ； BERT ； 柱搜索

Chinese Definition Modeling Based on BERT and Beam Seach

Qinan Fan, Cunliang Kong, Liner Yang, Erhong Yang
Beijing Language and Culture University

Abstract

Definition modeling task refers to generate a corresponding definition for the target word. Previous study on Chinese definition modeling task did not consider the context of the target word. For the first time, this thesis uses the context information of the target word in Chinese definition modeling task and proposes a definition generation model based on BERT and beam search. For experiments, we construct the CWN Chinese definition modeling dataset containing context of the target word. In addition to BLEU score, semantic similarity is used as an additional automatic evaluation metric. The experimental results show that the model has a significant improvement in Chinese CWN dataset and English Oxford dataset, and the results of human evaluation are consistent with the results of automatic evaluation. At last, this thesis makes an in-depth analysis of the generated instances.

Keywords: Chinese Definition Modeling , BERT , Beam Seach

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)

第十九届中国计算语言学大会论文集，第337页-第349页，海南，中国，2020年10月30日至11月1日。

1 引言

释义生成（又称释义建模，Definition Modeling）任务由 Noraset et al. (2017)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。释义生成任务不论在自然语言处理（Natural Language Processing，简称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它NLP任务中，本任务可以极大丰富词典释义资源。**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 et al. (2017)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 et al. (2018)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，训练模型生成词语在给定上下文中的释义。Ishiwatari et al. (2019)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。前人研究证明，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 et al. (2020)首次开展了释义生成任务研究，此项工作将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。

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 。

Figure 1: 中文释义生成示例

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获得充分训练，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，简称CWN）构建了新的中文释义生成数据集，与Yang et al. (2020)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。除了BLEU指标之外，本文采用语义相似度作为额外的

第十九届中国计算语言学大会论文集，第337页-第349页，海南，中国，2020年10月30日至11月1日。

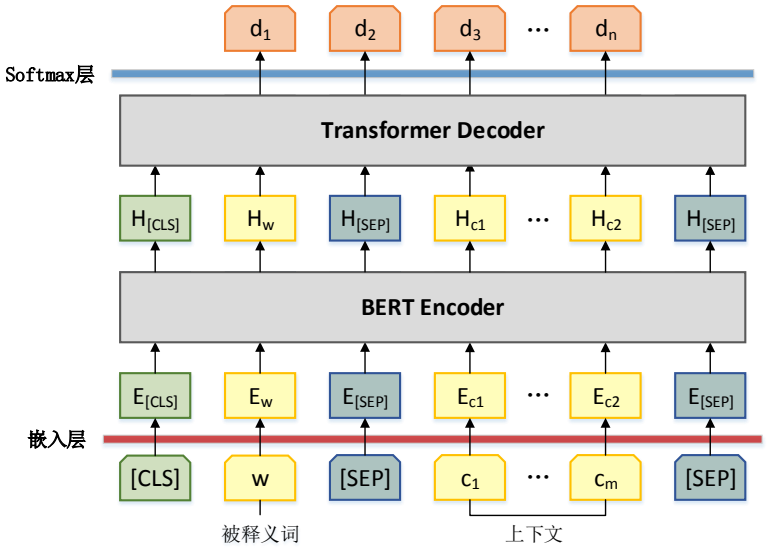


Figure 2: 模型图

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 et al. (2018)构建的英文Oxford数据集上实验结果同样明显超出基线模型。另外，我们对本文模型及基线模型在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：

- 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。
- 提出了基于BERT与柱搜索策略的释义生成模型，有效弥补了数据量不足的缺陷，获得了很好的效果。
- 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大问题。

2 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，以及包含该词语的一句上下文$C = [c_1, \ldots, c_m]$，为其生成一句相应的释义$D = [d_1, \ldots, d_n]$。模型的生成过程可以用条件概率表示为：

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

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。

2.1 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 (Devlin et al., 2018)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，让模型能够获得BERT从大规模语料中学到的先验知识。

本文将目标词w和上下文序列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

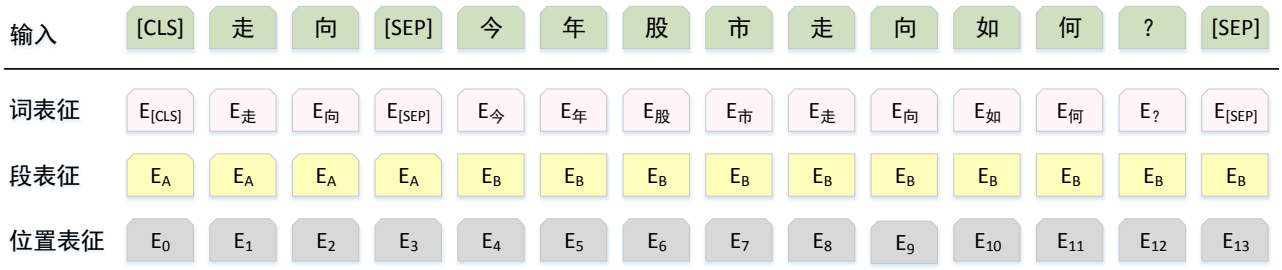


Figure 3: BERT嵌入层

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：

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

其中$\circ$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。

2.2 Transformer解码器

Transformer(Vaswani et al., 2017)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连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，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：

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{3}$$

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{Attention}_1, \text{Attention}_2, ..., \text{Attention}_\text{h}) \tag{4}$$

其中Q、K和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Q、K和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Q来自于上一层解码器的输出，K和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。

2.3 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，而该算法具有一些弊端。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，能够一定程度上缓解上述贪心算法的问题。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作为最终输出，相比贪心算法能够进一步提升生成效果。

3 实验

3.1 数据集

不论在英文还是中文上，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）[0]是一个由台湾中研院开发的词汇语义关系知识库，其中的大部分义项都具有多条例句，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[1]将数据由繁体中文转换为简体，使用jieba工具[2]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。本文在英文Oxford数据集上也开展了实验，此数据集由Gadetsky et al. (2018)通过牛津在线词典[3]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。

数据集		被释义词数量	释义数量	数据条数	上下文长度	释义长度
CWN	训练集	6,574	21,736	67,861	34.49	14.76
	验证集	823	2,606	8,082	34.73	14.60
	测试集	824	2,774	8,599	34.06	14.72
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

Table 1: 数据集规模统计

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2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。

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	1,896	768	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	1671	202	88	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	1777	229	96	134	19
	占比	4.00%	14.71%	64.06%	8.26%	3.46%	4.83%	0.68%

Table 2: CWN数据集释义包含的上下文数量统计

3.2 基线模型

本文将Transformer模型(Vaswani et al., 2017)和LOG-CaD模型(Ishiwatari et al., 2019)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很好的结果。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。

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

- **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模型提取了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。

3.3 实验设置

本文的Transformer模型基于FAIR开源代码库[4]实现，使用预训练的中文词向量(Li et al., 2018)和fastText词向量(Bojanowski et al., 2017)分别对中文和英文数据的词嵌入进行初始化，词表维数为300维，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2048。训练过程使用Adam优化器(Kingma and Ba, 2015)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(Wolf et al., 2019)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4000；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2000。两阶段的dropout均设置为0.2。中文和英文释义的词嵌入使用了和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（考虑到效率问题，这里使用NLTK translate包[5]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。

3.4 实验结果

本文分别在中文CWN数据集和英文Oxford数据集上评测了模型效果。由于前人使用的BLEU(Papineni et al., 2001)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[6]分别对生成释义和参考答案句子进行编码(Reimers and Gurevych, 2019; Reimers and Gurevych, 2020)，然后使用scipy包[7]计算两个句向量的余弦相似度。表3和表4中分别给出了BLEU和语义相似度两个指标的实验结果。其中Transformer (Vaswani et al., 2017)和LOG-CaD (Ishiwatari et al., 2019)为本文的基线模型，ESD-sem为Li et al. (2020)提出的基于显式语义分解的模型。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。

模型	CWN		Oxford	
	验证集	测试集	验证集	测试集
Transformer (Vaswani et al., 2017)	21.16	20.77	17.03	17.02
LOG-CaD (Ishiwatari et al., 2019)	30.76	29.58	19.13	18.95
ESD-sem (Li et al., 2020)	–	–	–	20.86
BERT-fix-encoder (Greedy)	38.96	37.25	19.87	20.14
BERT-fine-tune (Greedy)	**43.25**	**40.05**	**21.95**	**22.01**

Table 3: 实验结果（BLEU）

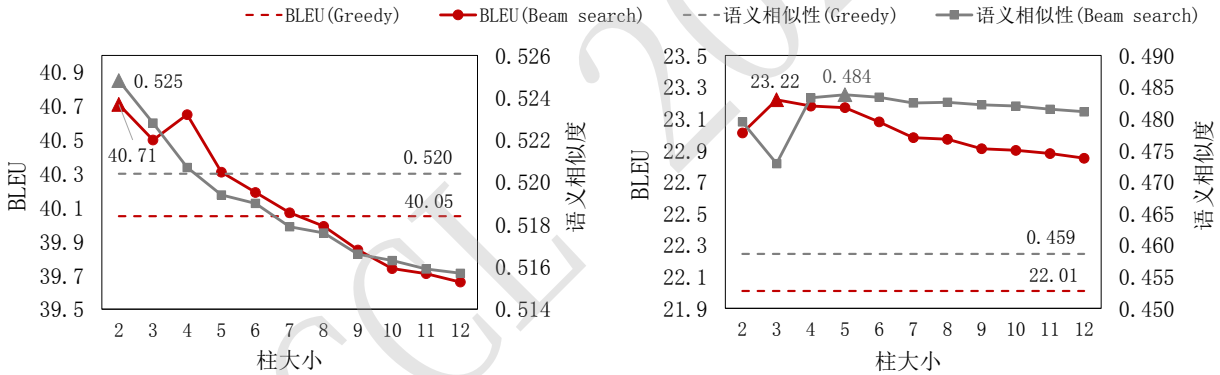
[4]https://github.com/pytorch/fairseq
[5]https://www.nltk.org/_modules/nltk/translate/bleu_score.html
[6]https://github.com/UKPLab/sentence-transformers
[7]https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.cdist.html

模型	CWN		Oxford	
	验证集	测试集	验证集	测试集
Transformer (Vaswani et al., 2017)	0.273	0.269	0.369	0.368
LOG-CaD (Ishiwatari et al., 2019)	0.362	0.415	0.269	0.306
BERT-fix-encoder (Greedy)	0.508	0.486	0.443	0.443
BERT-fine-tune (Greedy)	**0.538**	**0.520**	**0.473**	**0.459**

Table 4: 实验结果（语义相似度）

可以看到，Transformer模型在中文CWN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。在英文Oxford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结果都得到了非常显著的提升，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。如图4所示，在中文CWN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结果低于贪心算法。在Oxford数据集上，柱搜索策略带来的提升更明显，但随着柱的增大也会出现指标下降的情况。针对这一现象，Cohen and Beck (2019)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提升。



(a) CWN数据集　　(b) Oxford数据集

Figure 4: 柱取不同大小的结果对比

4 质量分析

4.1 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。附录A.1中展示了部分用于人工评价的生成释义实例。我们请了四名语言学专业学生作为标注员，使用Likert量表(Likert, 1932)，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5中展示了四名标注员的人工评价结果。

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。

	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**

Table 5: CWN数据集人工评价结果

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，如表6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。

人工评价 / 自动评价	语法	语义
BLEU	0.245 ($p < 0.0001$)	0.482 ($p < 0.0001$)
语义相似度	0.298 ($p < 0.0001$)	0.639 ($p < 0.0001$)

Table 6: 自动评价与人工评价指标Pearson相关系数

4.2 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意思的难度越大，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BLEU指标计算时，会将多义词的全部释义都作为参考答案，这会对我们的分析结果产生影响，因此本节选用语义相似度作为衡量指标，对BLEU指标的影响分析见附录A.2。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。

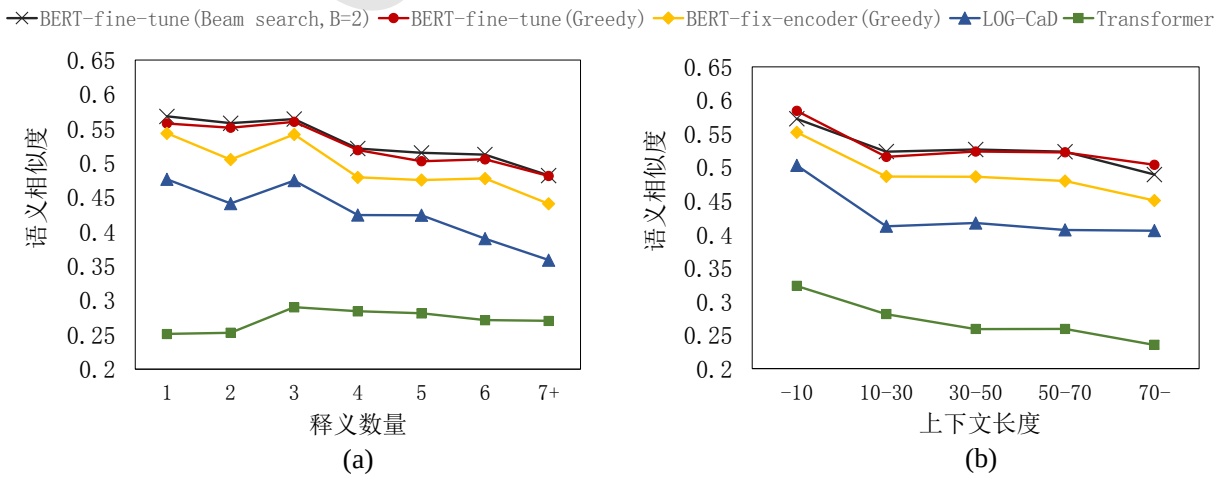


Figure 5: 数据分布情况对语义相似度结果的影响

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，此方法可能难以起到很好的消歧效果。

图5(b)中的上下文长度是按字统计的，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。

4.3 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7中。

- 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(Noraset et al., 2017)，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7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7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7中针对此问题给出的第二个实例。

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。
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。

Table 7: 生成释义的问题及相应实例

5 相关工作

5.1 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 et al. (2017)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。Gadetsky et al. (2018)提出了基于AdaGram对词向量进行消歧的方法。Mickus et al. (2019)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。Ishiwatari et al. (2019)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。Li et al. (2020)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。

还有研究者从低维密集词向量的可解释性问题出发研究释义建模任务。Chang et al. (2018)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，能够对目标词嵌入包含的语义信息进行直接解释。Chang and Chen (2019)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。

释义生成任务在中文上的研究还很少。Yang et al. (2020)等人首次在中文上开展释义生成任务，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。基于此，本文首次将上下文信息引入中文释义生成任务。

5.2 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(Mikolov et al., 2013a; Mikolov et al., 2013b)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。后来提出的ELMo(Peters et al., 2018)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。随后，GPT(Radford, 2018)和BERT(Devlin et al., 2018)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任务中都展示出强大的性能(Adhikari et al., 2019; Kaneko and Komachi, 2019)。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。

6 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中文CWN数据集以及前人构建的英文Oxford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，又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。在未来的工作中，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。此项工作使用的完整数据和代码公开于https://github.com/blcuicall/AutoDict/tree/ccl2020。

参考文献

Ashutosh Adhikari, Achyudh Ram, Raphael Tang, and Jimmy Lin. 2019. Docbert: Bert for document classification. *ArXiv*, abs/1904.08398.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors

with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Ting-Yun Chang and Yun-Nung Chen. 2019. What does this word mean? explaining contextualized embeddings with natural language definition. In *EMNLP/IJCNLP*.

Ting-Yun Chang, Ta-Chung Chi, Shang-Chi Tsai, and Yun-Nung Chen. 2018. xsense: Learning sense-separated sparse representations and textual definitions for explainable word sense networks. *ArXiv*, abs/1809.03348.

Eldan Cohen and J. Christopher Beck. 2019. Empirical analysis of beam search performance degradation in neural sequence models. In *ICML*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Artyom Gadetsky, Ilya Yakubovskiy, and Dmitry Vetrov. 2018. Conditional generators of words definitions. In *ACL*.

Shonosuke Ishiwatari, Hiroaki Hayashi, Naoki Yoshinaga, Graham Neubig, Shoetsu Sato, Masashi Toyoda, and Masaru Kitsuregawa. 2019. Learning to describe unknown phrases with local and global contexts. In *NAACL-HLT*.

Masahiro Kaneko and Mamoru Komachi. 2019. Multi-head multi-layer attention to deep language representations for grammatical error detection. *ArXiv*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *ArXiv*, abs/1412.6980.

Shen Li, Zhe Zhao, Renfen Hu, Wensi Li, Tao Liu, and Xiaoyong Du. 2018. Analogical reasoning on chinese morphological and semantic relations. In *ACL*.

Jiahuan Li, Y. Bao, Shujian Huang, X. Dai, and Jiajun Chen. 2020. Explicit semantic decomposition for definition generation. In *ACL*.

Rensis Likert. 1932. A technique for the measurement of attitudes.

Timothee Mickus, Denis Paperno, and Mathieu Constant. 2019. Mark my word: A sequence-to-sequence approach to definition modeling. *ArXiv*, abs/1911.05715.

Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *ArXiv*, abs/1301.3781.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. *ArXiv*, abs/1310.4546.

Thanapon Noraset, Chen Liang, Lawrence Birnbaum, and Doug Downey. 2017. Definition modeling: Learning to define word embeddings in natural language. In *AAAI*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. Bleu: a method for automatic evaluation of machine translation. In *ACL*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *NAACL*.

Alec Radford. 2018. Improving language understanding by generative pre-training.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP*.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. *arXiv*.

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 *NIPS*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Liner Yang, Cunliang Kong, Yun Chen, Yang Liu, Qinan Fan, and Erhong Yang. 2020. Incorporating sememes into chinese definition modeling. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:1669–1677.

A 附录

A.1 模型生成释义实例

被释义词：作用 **上下文**：一旦自动增词及印象增强的功能开始作用，这就不是一个可以重复的实验。	
参考答案	特定对象运作并发挥功能。
Transformer	比喻特定事件进行的快慢的程度。
LOG-CaD	特定对象的作用。
BERT-fix-encoder(Greedy)	特定对象在使用时发挥的效能。
BERT-fine-tune(Greedy)	特定对象所能发生的效能，常用于否定句。
BERT-fine-tune(Beam=2)	特定对象在使用时发挥的效能。
被释义词：开阔 **上下文**：真正的享乐，只在心灵的充实，胸襟的开阔	
参考答案	形容心胸宽大。
Transformer	享受物质上所带来的快乐，常隐含负面的涵义。
LOG-CaD	形容比喻思想或知识深刻且没有阻碍。
BERT-fix-encoder(Greedy)	形容比喻度量大，能容忍。
BERT-fine-tune(Greedy)	形容比喻度量大，能容忍。
BERT-fine-tune(Beam=2)	比喻拓展后述知识领域或经验。
被释义词：大打折扣 **上下文**：森林受到大量砍伐，植物吸收二氧化碳的能力大打折扣。	
参考答案	比喻将前述对象的功能或价值大幅度地降低。
Transformer	物体或有机体把组织外部的物质吸到内部。
LOG-CaD	以unk的方式使特定对象离开说话者的距离。
BERT-fix-encoder(Greedy)	形容特定对象的程度降低。
BERT-fine-tune(Greedy)	比喻前述对象的价值降低到负面评价。
BERT-fine-tune(Beam=2)	比喻前述对象的价值降低到负面评价。
被释义词：扯 **上下文**：阳台的墙缘上停着一只麻雀，仰着头、张嘴合嘴，扯着喉咙叫。	
参考答案	喉咙用力以大声量发声。
Transformer	前述物体移动时的力量带动后述物体。
LOG-CaD	用肢体的动作使声音达到最佳状态。
BERT-fix-encoder(Greedy)	使特定对象的毛发变得滑顺且触感好。
BERT-fine-tune(Greedy)	使特定对象的头发变得紧密。
BERT-fine-tune(Beam=2)	使特定对象离开其附着的表面。
被释义词：想要 **上下文**：现在的小朋友想要什么，就有什么。	
参考答案	提出后述需求或条件。
Transformer	表加强语气，强调后述对象。
LOG-CaD	表达成后述事件的愿望。
BERT-fix-encoder(Greedy)	因喜好而想要得到后述对象。
BERT-fine-tune(Greedy)	因分开而思念后述对象。
BERT-fine-tune(Beam=2)	认为符合自己心意而同意做后述事件。

Table 8: 模型生成释义实例

A.2 数据分布情况对BLEU结果的影响

如图6所示，两张子图中分别展示了不同释义数量及上下文长度对模型BLEU结果的影响。

与4.2节类似，(a)子图中Transformer模型折线趋势与其他模型差别较大，这主要是受到模型本身生成效果较差的影响。

从图6(a)中可以看到，不论是基线模型还是本文提出的模型，释义数量小于2条的词的BLEU结果都很不好，这与使用的BLEU指标的计算方式有关。该指标在计算时，会将多义词的全部释义都作为参考答案，因此当释义数量过少时，BLEU值会比较低。当释义数量超过5条时，本文模型和LOG-Cab模型的BLEU值都出现了不同程度的下降，但对LOG-Cab模型的结果影响显然更大。

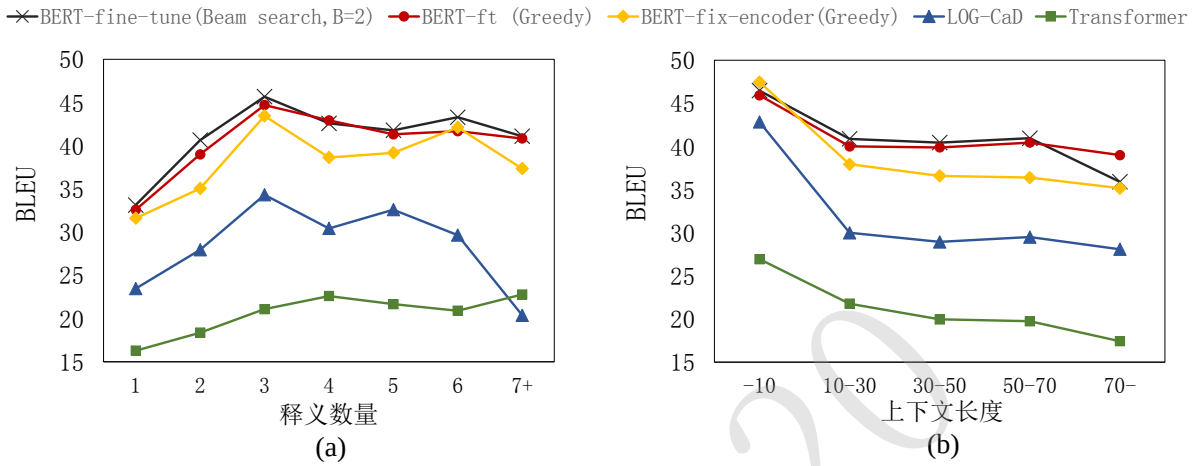


Figure 6: 数据分布情况对BLEU结果的影响

图6(b)中对不同上下文长度的BLEU结果进行了比较。可以看到当上下文长度超过10时，BLEU指标出现非常明显的下降，和对语义相似度指标的影响情况基本一致。