基于 BERT 的任务导向对话系统自然语言理解的改进模型与调优方法

周奇安[1] 李舟军[1]

（1. 北京航空航天大学 计算机学院，北京市 100191）

摘要：任务导向对话系统的自然语言理解，其目的就是解析用户以自然语言形式输入的语句，并提取出可以被计算机所理解的结构化信息。它包含意图识别和槽填充两个子任务。BERT 是近期提出来的一种自然语言处理预训练模型，已有研究者提出基于 BERT 的任务导向对话系统自然语言理解模型。在此基础上，该文提出了一种改进的自然语言理解模型，其编码器使用 BERT，而解码器基于 LSTM 与注意力机制构建。然后，该文提出了该模型的两种调优方法：锁定模型参数的训练方法、使用区分大小写的预训练模型版本。在基线模型与改进模型上，这些调优方法均能够显著改进模型的性能。实验结果显示，利用改进后的模型与调优方法，可以分别在 ATIS 和 Snips 两个数据集上得到 0.8833 和 0.9251 的句子级准确率。

关键词：任务导向对话系统；自然语言理解；BERT

中图分类号：TP391 **文献标识码：**A

BERT Based Improved Model and Tuning Techniques for Task-oriented Dialog System Natural Language Understanding

Qi'an Zhou[1] Zhoujun Li[1]

(1. School of Computer Science and Engineering, Beihang University, Beijing 100191, China)

Abstract: The purpose of task-oriented dialog system natural language understanding is to parse sentences entered by the user in natural language, extracting structured information which is understandable by the computer. It consists of two sub-tasks: intent detection and slot filling. BERT is a pretrained natural language processing model proposed recently. Other researchers have proposed a task-oriented dialog system natural language understanding model based on BERT. This paper proposed an improved natural language understanding model, using BERT as encoder, while the decoder is built with LSTM and attention mechanism. Furthermore, this paper proposed two tuning techniques on this model: training method with frozen model parameters, using cased-sensitive version of pretrained model. These techniques can acquire performance improvement both on the baseline and the improved model. Experiments have shown that the improved model and tuning techniques can get 0.8833 and 0.9251 sentence level accuracy on ATIS and Snips datasets respectfully.

Key words: task-oriented dialog system; natural language understanding; BERT

0 引言

任务导向对话系统自然语言理解是自然语言处理领域的一个难点问题。其目的是解析用户以自然语言形式输入的语句，并提取出可以被计算机所理解的结构化信息。任务导向对话系统使用自然语言与用户交流，并通过这种方式收集足够的信息，为用户完成其所希望的特定任务。自然语言理解的准确度直接关系到整个对话系统的可用性和用户友好性。

任务导向对话系统自然语言理解的两个最重要的子任务是意图识别和槽填充。意图是指用户对话过程中所希望表达的真实目的，一般以动宾短语来刻画，例如：查询天气，购买机票等。而槽是语句中所包含的关键信息或特定概念，是指将用户意图具体化的一些特定条件或参数，以便明确地刻画用户意图的完整信息。一般而言，每一个语句都有恰好一个意图，因此意图识别可以视为一种分类任务。而槽在句子中出现的位置可以使用 BIO 编码表示，因此槽填充可以视为一种序列标注任务。表 1 给出了一个意图识别和槽填充的样例。

收稿日期：2019-6-18；**定稿日期**：2019-8-15

基金项目：国家自然科学基金（61672081、U1636211、61370126、61602025）；国家重点研发计划（2016QY04W0802）

表 1 意图识别与槽填充样例

用户询问	槽标签	意图
first	B-class_type	airfare
class	I-class_type	
fares	O	
from	O	
boston	B-fromloc	
to	O	
denver	B-toloc	

在深度学习技术普及之前，业界常用的自然语言理解方法都是基于传统模型或人工制定的规则。这些方法从数据中学习知识的能力较弱，传统模型的学习能力不足是导致模型性能低下的主要问题。而随着深度学习技术的发展，越来越多的深度神经网络模型开始被应用到自然语言理解任务中。深度神经网络模型具有非常强大的拟合能力，减轻了传统模型学习能力的不足。但是，深度神经网络的训练需要大量的人工标注数据。获取这样的数据的成本高昂，导致很多领域的有标注语料数量较少。

解决这一问题的方法之一是使用预训练模型。这些模型通常使用无标注的文本数据训练，可以学习到文本数据的向量表示，并且可以在多种自然语言处理任务上进行微调以提高有监督模型在这些任务上的性能。BERT[1]是近期提出的一种预训练模型，它在包括任务导向对话系统自然语言理解在内的多个自然语言处理任务上都取得了优异的成绩。然而，现有的利用 BERT 解决任务导向对话系统自然语言理解任务的方法使用的解码器没有根据问题的特点优化，而在训练过程中也存在丢失预训练模型中的通用信息与丢失大小写信息的问题。本文提出了一种改进的自然语言理解模型，其编码器使用 BERT，而解码器基于 LSTM 与注意力机制构建。接下来，本文还提出了该模型的两种调优方法：锁定模型参数的训练方法、区分大小写的预训练模型版本。这些方法在 ATIS[2]和 Snips[3]两个数据集上取得了更优的成绩。

1 相关工作

在 BERT 被提出之前，学术界就已经对意图识别和槽填充任务进行了大量的研究。Mesnil 等人[4]对各种使用 RNN 完成槽填充任务的方法进行了一次调查，Yao 等人[5]对一种使用 Elman 型 RNN 构建语言模型的方法进行了修改，以完成槽填充任务。Kim 等人[6]利用同义词、反义词和相关词信息对 GloVe 预训练词嵌入进行了微调，并使用双向 LSTM 网络完成意图识别和槽填充任务。Yao 等人[7]使用多层 LSTM 来完成槽填充任务。Mensio 等人[8]使用了词语和句子两级 RNN 结构完成槽填充任务，并使用 LSTM 作为句子级 RNN 细胞。Deoras 和 Sarikaya[9]使用 DBN 完成槽填充任务。Deng 等人[10]使用 K-DCN 完成意图识别任务，并使用 K-DCN 作为特征提取手段完成槽填充任务。Xu 和 Sarikaya[11]提出了一种基于卷积神经网络和三角 CRF（TriCRF）[12]的意图识别与槽填充联合模型。Zhang 和 Wang[13]提出了一种基于双向 GRU 的意图识别与槽填充联合模型。Liu 和 Lane[14]提出了两种基于注意力机制的意图识别与槽填充联合模型。Schumann 和 Angkititrakul[15]提出了一种考虑语音识别错误的意图识别与槽填充联合模型。Zhang 等人[16]提出了一种联合槽填充、意图识别、专用领域语言模型和通用领域语言模型的多任务模型。Goo 等人[17]提出了一种利用意图识别的结果对槽填充过程进行限制的改进方法。

除去 BERT 外，自然语言处理领域比较著名的预训练模型还有艾伦人工智能研究所提出的 ELMo[18]，以及 OpenAI 组织提出的 GPT[19]和 GPT-2[20]。BERT 已经在其它多个自然语言处理任务中得到了应用。除 BERT 原论文中介绍的分类、句子相似度、自然语言推断与问答任务外，Lee 等人[21]使用生物医学领域语料重新训练了 BERT，Adhikari 等人[22]使用微调的方法将 BERT 应用于文本分类，Nogueira 和 Cho[23]使用微调的方法将 BERT 应用于文章重排序。

Chen 等人[24]将 BERT 用作编码器，并且使用全连接层和条件随机场（CRF）作为解码器，利用微调的方法解决意图识别和槽填充任务。本文内容是对这种方法的改进。

2 基线模型介绍

本文使用 Chen 等人[24]所提出的模型作为基

线。基线模型的编码器部分基于 BERT 实现，而解码器部分则使用了简单的全连接层和条件随机场（CRF）。其完整结构如图 1 所示。

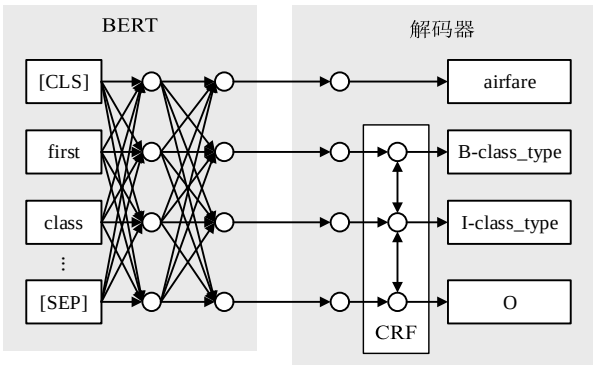

图 1 基线模型结构

2.1 基于 BERT 的编码器

BERT 是一种多层双向 Transformer 编码器。其结构在 BERT[1]和 Transformer[25]的论文中有详细的说明，本论文不再赘述。BERT 的输入$\boldsymbol{x}=(x_1,\ldots,x_T)$是一个标记（token）序列，其中开头的$x_1$为分类特殊标记“[CLS]”，结尾的$x_T$为句子分隔特殊标记 “[SEP]” ，其余部分为对输入的句子执行 WordPiece 切分后的标记序列。将$\boldsymbol{x}$输入 BERT 模型后，得到一个编码向量序列$H_{\text{BERT}}=(\boldsymbol{h}_{\text{BERT},1};\ldots;\boldsymbol{h}_{\text{BERT},T})$。

本文中所使用的是预训练 BERT 模型的微调方法。在开始训练前，BERT 的模型参数使用 GitHub 页面[1]里提供的预训练模型参数初始化，而不是执行随机初始化。在训练过程中，BERT 的模型参数和其它参数一样，仍受到优化算法的优化。

2.2 解码器结构

意图识别与槽填充联合模型的解码器具有两种功能，分别是生成意图的预测和生成槽标签的预测。意图的预测使用的是一个全连接层：

$$p(y_{\text{intent}})=\text{softmax}(\boldsymbol{h}_{\text{BERT},1}W_{\text{intent}}+\boldsymbol{b}_{\text{intent}})\#(1)$$

其中，$\boldsymbol{h}_{\text{BERT},1}$是分类特殊标记“[CLS]”对应的编码向量，$W_{\text{intent}}$和$\boldsymbol{b}_{\text{intent}}$都是可训练的模型参数，softmax是柔性最大值函数，其表达式为：

$$\text{softmax}(z)_i=\frac{\exp(z_i)}{\sum_{j=1}^{K}\exp(z_j)}\#(2)$$

输入序列每个位置的槽标签概率分布也可以用类似的方法得出。然而，BIO 编码的槽标签之间具有强制的依赖关系，如果直接最大化每个位置的槽标签概率有可能会生成非法的槽标签序列。使用条件随机场可以解决这样的问题。条件随机场除了考虑每一个位置的槽标签概率分布外，还会考虑相邻位置的标签的依赖关系。在省略归一化项的情况下，其表达式如下：

$$p(y_{\text{slot},1},\ldots,y_{\text{slot},T})\propto p(y_{\text{slot},i}|i=1)\prod_{i=2}^{T}p(y_{\text{slot},i}|i)p(y_{\text{slot},i}|y_{\text{slot},i-1})\#(3)$$

其中，$p(y_{\text{slot},i}|i)$为每个位置独立的槽标签概率分布，该分布由 BERT 在这个位置输出的编码向量得出：

$$p(y_{\text{slot},i}|i)\propto\exp(\boldsymbol{h}_{\text{BERT},i}W_{\text{slot}}+\boldsymbol{b}_{\text{slot}})\#(4)$$

而$p(y_{\text{slot},i}|y_{\text{slot},i-1})$为每一对相邻的位置的槽标签依赖关系，由一个转移矩阵得出：

$$p(y_{\text{slot},i}|y_{\text{slot},i-1})\propto\exp(W_{\text{trans},y_{\text{slot},i-1},y_{\text{slot},i}})\#(5)$$

其中W_{slot}、$\boldsymbol{b}_{\text{slot}}$、$W_{\text{trans}}$为可训练的模型参数。在训练过程中，为了得到正确的槽标签的对数似然函数，需对式(3)执行归一化。

3 改进的模型

先前已有的对 BERT 模型的微调工作大多和基线模型类似，解码器使用简单的全连接层，没有与特定任务相关的改进。在 BERT 问世以前，很多不使用预训练模型的方法也在意图识别与槽填充两个任务中取得了较好的成绩。基于长短期记忆网络（LSTM）和注意力机制的模型就是其中之一。基于以上工作，本文在第 2.2 节所述的解码器基础上引入基于 LSTM 与注意力机制的任务特定结构，改进后的模型结构如图 2 所示。

[1] https://github.com/google-research/bert

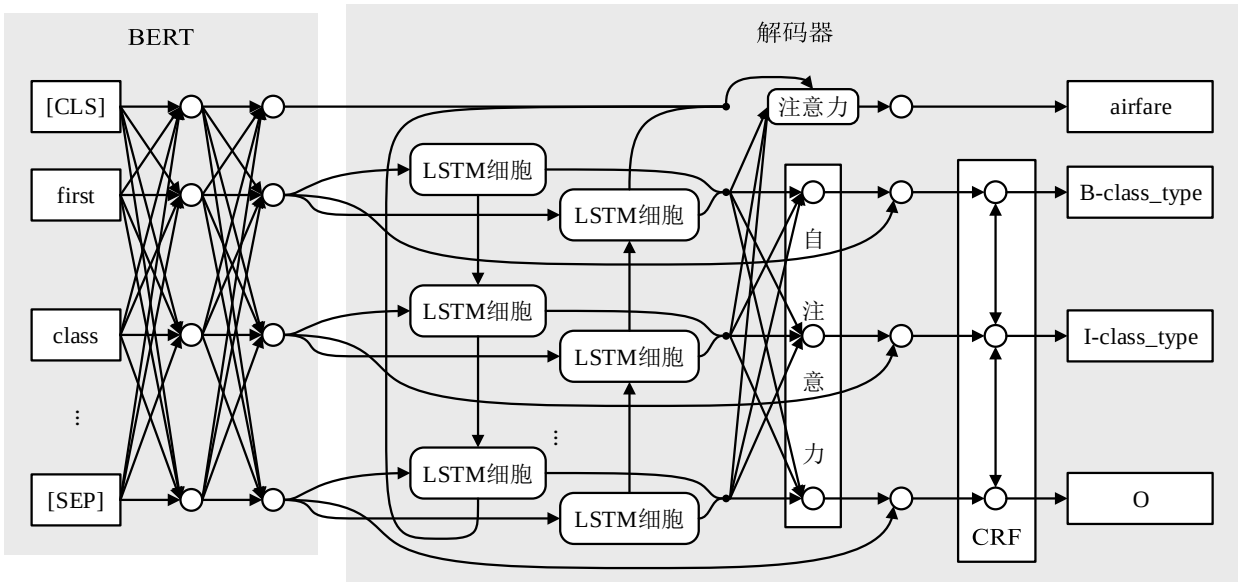

图 2 改进的模型结构

LSTM[26–28]是循环神经网络（RNN）的一种变体，它的细胞结构如图 3 所式。一个 LSTM 细胞中包含遗忘门F，输入门I与输出门O，其形式如下：

$$F(\boldsymbol{x},\boldsymbol{h}_{\text{prev}})=\text{sigmoid}\left([\boldsymbol{h}_{\text{prev}},\boldsymbol{x}]W_{\text{f}}+\boldsymbol{b}_{\text{f}}\right)\#(6)$$

$$I(\boldsymbol{x},\boldsymbol{h}_{\text{prev}})=\text{sigmoid}\left([\boldsymbol{h}_{\text{prev}},\boldsymbol{x}]W_{\text{i}}+\boldsymbol{b}_{\text{i}}\right)\#(7)$$

$$O(\boldsymbol{x},\boldsymbol{h}_{\text{prev}})=\text{sigmoid}\left([\boldsymbol{h}_{\text{prev}},\boldsymbol{x}]W_{\text{o}}+\boldsymbol{b}_{\text{o}}\right)\#(8)$$

LSTM 细胞的形式如下：

$$C(\boldsymbol{x},\boldsymbol{h}_{\text{prev}},\boldsymbol{c}_{\text{prev}})=F(\boldsymbol{x},\boldsymbol{h}_{\text{prev}})\circ\boldsymbol{c}_{\text{prev}}+I(\boldsymbol{x},\boldsymbol{h}_{\text{prev}})\circ\tanh\left([\boldsymbol{h}_{\text{prev}},\boldsymbol{x}]W_{\text{c}}+\boldsymbol{b}_{\text{c}}\right)\#(9)$$

$$H(\boldsymbol{x},\boldsymbol{h}_{\text{prev}},\boldsymbol{c}_{\text{prev}})=O(\boldsymbol{x},\boldsymbol{h}_{\text{prev}})\circ C(\boldsymbol{x},\boldsymbol{h}_{\text{prev}},\boldsymbol{c}_{\text{prev}})\#(10)$$

$$\text{LSTM}(\boldsymbol{x},\boldsymbol{h}_{\text{prev}},\boldsymbol{c}_{\text{prev}})=\left(H(\boldsymbol{x},\boldsymbol{h}_{\text{prev}},\boldsymbol{c}_{\text{prev}}),C(\boldsymbol{x},\boldsymbol{h}_{\text{prev}},\boldsymbol{c}_{\text{prev}})\right)\#(11)$$

LSTM 细胞可以在 LSTM 的输入序列上展开，得到 LSTM 的输出序列。将 LSTM 细胞在 BERT 输出的编码序列上沿正反两个方向展开，就会得到：

$$(\overrightarrow{\boldsymbol{h}}_{\text{LSTM},i},\overrightarrow{\boldsymbol{c}}_{i})=\begin{cases}(\boldsymbol{0},\boldsymbol{0}) & i=0\\ \text{LSTM}(\overrightarrow{\boldsymbol{h}}_{\text{BERT},i},\overrightarrow{\boldsymbol{h}}_{\text{LSTM},i-1},\overrightarrow{\boldsymbol{c}}_{i-1}) & i>0\end{cases}\#(12)$$

$$(\overleftarrow{\boldsymbol{h}}_{\text{LSTM},i},\overleftarrow{\boldsymbol{c}}_{i})=\begin{cases}(\boldsymbol{0},\boldsymbol{0}) & i=T+1\\ \text{LSTM}(\overleftarrow{\boldsymbol{h}}_{\text{BERT},i},\overleftarrow{\boldsymbol{h}}_{\text{LSTM},i+1},\overleftarrow{\boldsymbol{c}}_{i+1}) & i\le T\end{cases}\#(13)$$

最后将两个方向的输出序列合并，就可以得到：

$$H_{\text{LSTM}}=(\boldsymbol{h}_{\text{LSTM},1};\ldots;\boldsymbol{h}_{\text{LSTM},T})\#(14)$$

$$\boldsymbol{h}_{\text{LSTM},i}=[\overrightarrow{\boldsymbol{h}}_{\text{LSTM},i},\overleftarrow{\boldsymbol{h}}_{\text{LSTM},i}]\#(15)$$

同时合并两个方向的最终隐藏状态，得到$\overleftrightarrow{\boldsymbol{h}}_{\text{LSTM}}=[\overrightarrow{\boldsymbol{h}}_{\text{LSTM},T},\overleftarrow{\boldsymbol{h}}_{\text{LSTM},1}]$。

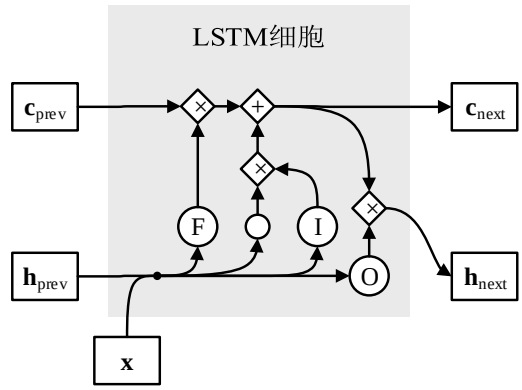

图 3 LSTM 细胞

注意力机制是一种记忆的检索机制，它可以从一系列以键值对形式保存的记忆中找到与某个主题最相关的部分，并以加权的形式返回检索到的内容。本文中使用的是 Bahdanau 等人[29]提出的注意力机制，它使用一个可训练的前馈神经网络来衡量键与主题之间的相关性：

$$\text{Bahdanau}(\boldsymbol{q},K,V)=\text{softmax}\left(a(\boldsymbol{q},K)\right)V\#(16)$$

$$a(\boldsymbol{q},K)_i=\tanh(\boldsymbol{q}W_a+\boldsymbol{k}_iU_a)\cdot\boldsymbol{v}_a\#(17)$$

其中$\boldsymbol{q}$是注意力机制需关注的主题，$K=(\boldsymbol{k}_1;\ldots;\boldsymbol{k}_T)$与$V=(\boldsymbol{v}_1;\ldots;\boldsymbol{v}_T)$分别为记忆的键和值，$W_a$、$U_a$和$\boldsymbol{v}_a$都是可训练的模型参数。对于意图识别任务，将$\boldsymbol{q}_{\text{intent}}=[\boldsymbol{h}_{\text{BERT},1},\overleftrightarrow{\boldsymbol{h}}_{\text{LSTM}}]$作为主题向量应用注意力机制后，得到：

$$\boldsymbol{c}_{\text{intent}}=\text{Bahdanau}(\boldsymbol{q}_{\text{intent}},H_{\text{LSTM}},H_{\text{LSTM}})\#(18)$$

而对于槽填充任务，则对 LSTM 的输出序列自身

应用注意力机制，得到：

$$\boldsymbol{c}_{\text{slot},i} = \text{Bahdanau}\left(\boldsymbol{h}_{\text{LSTM},i}, H_{\text{LSTM}}, H_{\text{LSTM}}\right)\#(19)$$

在得到注意力机制的输出后，将基线模型中的式(1)和式(4)修改为如下所述：

$$p(y_{\text{intent}}) = \text{softmax}(\boldsymbol{c}_{\text{intent}} W_{\text{intent}} + \boldsymbol{b}_{\text{intent}})\#(20)$$

$$p\left(y_{\text{slot},i} | i\right) \propto \exp\left(\left[\boldsymbol{h}_{\text{BERT},i}, \boldsymbol{c}_{\text{slot},i}\right] W_{\text{slot}} + \boldsymbol{b}_{\text{slot}}\right)\#(21)$$

得到改进后的模型。

4 调优方法

4.1 锁定模型参数的训练方法

在 BERT 这样的多层模型中，不同的层往往可以提取不同抽象层次的特征。一般而言，靠近输入的层提取的是抽象层次较低的特征，这些特征往往包含自然语言本身的特性，而与具体的任务的关联不大，通用性较强。而靠近输出的层提取的是抽象层次较高的特征，这些特征通常因任务的不同而具有较大的独特性。在对预训练模型执行微调的过程中，如果无差别训练所有的模型参数，有可能导致预训练模型丢失先前学习到的通用知识，造成过拟合的问题。

BERT 的结构按照从输入到输出的顺序依次为 1 个词嵌入层、12 个 Transformer 层和 1 个池化层。词嵌入负责将输入的标记序列转换为向量序列，Transformer 层负责不同层次的特征提取，池化层则对分类特殊标记“[CLS]”对应的输出 $\boldsymbol{h}_{\text{BERT},1}$应用了一次线性变换。根据上文所述，词嵌入层和靠近输入的一部分 Transformer 层可以提取较低层次的特征。

为了保护这些通用性较强的层，缓解过拟合的现象，本文尝试在微调过程中锁定这些层的参数。如图 4 所示，BERT 模型的正向传播过程和往常一样，没有任何变化，而反向传播过程略过了上文所述的词嵌入层和部分 Transformer 层。反向传播过程不会计算这些层中可训练参数的梯度，也不会更新可训练参数的值。除此之外，BERT 中还利用了 dropout[30]技术进行正则化。由于被锁定的层不存在训练过程，无需正则化，这些层中的 dropout 也被去除。

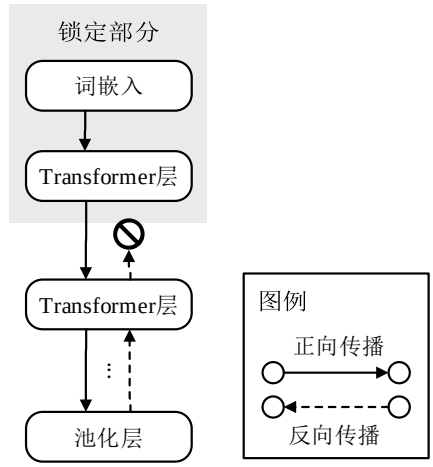

图 4 锁定参数

4.2 使用区分大小写的预训练模型版本

在传统的英文自然语言处理方法中，经常需要忽略输入文本的大小写信息。这是由于英文存在句首大写的规则，如果不忽略大小写信息，同一个单词会在词汇表中出现大写和小写两种形式，这会导致词汇表的大小发生膨胀。然而对于任务导向对话系统自然语言理解任务而言，大小写信息又具有非常重要的作用。以表 2 中的语句为例，用户询问中“The Fall-Down Artist”这个词组开头大写表明了它是一个专有名词，不论这个名字的内容有何意义，它都很可能形成一个槽。如果模型无法获得这个词组的大小写信息，就必须试图理解这个词组以及它的上下文的意义，这很可能对模型的准确率带来负面的影响。

表 2 区分大小写的样例

用户询问	槽标签	意图
Rate	O	RateBook
The	B-object_name	
Fall-Down	I-object_name	
Artist	I-object_name	
5	B-rating_value	
stars	B-rating_unit	

在基线模型中，预训练模型使用的是不区分大小写的版本。而 BERT 的预训练模型是具有区分大小写的版本的。不仅如此，BERT 预训练过程中使用的巨大的数据量可以弥补词汇表膨胀带来的问题。因此，本文尝试使用区分大小写的预训练模型版本替换原先的版本，进行微调工作。

5 实验

5.1 数据集

本文的实验使用的是ATIS和Snips两个数据集。ATIS数据集[2]是由美国国防高级研究计划局发布的，它包含关于航班信息的口语询问。ATIS数据集的划分方式与Tur等人[2]一致，包含4978个训练样本、893个测试样本、26种意图和82种槽。Snips数据集[3]是Snips公司利用众包的方式获得的一个语音助手数据集。本文使用的是Snips数据集的2017年自定义意图引擎部分，其划分方式和Snips公司在GitHub上开源[2]的方式一致，包含13084个训练样本、700个测试样本、7种意图和39种槽。

5.2 训练过程

本文中的BERT预训练模型使用的是BERT-Base版本。LSTM的隐层大小为128，LSTM层的输入和输出向量各应用了0.5概率的dropout，LSTM细胞间的循环连接无dropout。

训练过程中使用的批次大小为16，优化算法为带L2梯度衰减的Adam算法[31]，学习速率为5×10^{-5}，优化算法的其它超参数与Devlin等人使用的超参数一致。学习速率在前10%的训练步骤线性增加，而在其余的步骤线性衰减。训练的轮次（epoch）数和BERT锁定参数的层数因数据集和模型版本而异，它们的最优值由训练集上的5折交叉验证确定。

除去有特殊说明的情况外，最终的性能评测对每一种模型形式都在全体训练集上执行5次独立的训练，并计算5次训练在测试集上得到的评测指标的平均值。

5.3 评测指标

本文的实验中意图识别的评测指标为准确率，槽填充的评测指标为F1值，双任务联合的评测指标为句子级准确率。设测试集中共有n个样本，$y_{\text{intent},i}$为第i个样本的意图预测值，$y^*_{\text{intent},i}$为第i个样本的意图真实值，则意图准确率定义为：

$$\text{acc}_{\text{intent}}=\frac{\sum_{i=1}^{n}\begin{cases}1 & y_{\text{intent},i}=y^*_{\text{intent},i}\\ 0 & y_{\text{intent},i}\neq y^*_{\text{intent},i}\end{cases}}{n}\#(22)$$

将槽标签$y_{\text{slot},i}$进行解码，得到槽的起止位置集合：

$$S=\left\{(l,r)\middle|r=\max\left(r\middle|\forall i\in(l,r]\left(y_{\text{slot},i}=\text{"I-}s\text{"}\right)\right),\ y_{\text{slot},l}=\text{"B-}s\text{"}\right\}\#(23)$$

设S_i为第i个样本的槽预测值，S_i^*为第i个样本的槽真实值，则槽F1定义为：

$$\text{F1}_{\text{slot}}=\frac{2\sum_{i=1}^{n}|S_i\cap S_i^*|}{\sum_{i=1}^{n}|S_i|+\sum_{i=1}^{n}|S_i^*|}\#(24)$$

句子级准确率定义为：

$$\text{acc}_{\text{sentence}}=\frac{\sum_{i=1}^{n}\begin{cases}1 & y_{\text{intent},i}=y^*_{\text{intent},i}\wedge S_i=S_i^*\\ 0 & \text{其它}\end{cases}}{n}\#(25)$$

5.4 结果与分析

[2] https://github.com/snipsco/nlu-benchmark

表 3 对比实验结果

数据集	模型	意图准确率	槽 F1	句子级准确率
ATIS	Hakkani-Tür 等[32]	0.926	0.943	0.807
ATIS	Liu 和 Lane[14]	0.911	0.942	0.789
ATIS	Goo 等[17]	0.941	0.952	0.826
ATIS	BERT+CRF（Chen 等[24]，基线模型）	0.9758	0.9583	0.8812[3]
ATIS	BERT+LSTM+注意力+CRF（本文）	0.9738	**0.96**	**0.8833**[3]
Snips	Hakkani-Tür 等[32]	0.969	0.873	0.732
Snips	Liu 和 Lane[14]	0.967	0.878	0.741
Snips	Goo 等[17]	0.97	0.888	0.755
Snips	BERT+CRF（Chen 等[24]，基线模型）	0.9877	0.9618	0.9089
Snips	BERT+LSTM+注意力+CRF（本文）	**0.9906**	**0.9678**	**0.9251**

[3] 由于 5 次独立训练无法在 ATIS 数据集上得到句子级准确率提升显著的结论，本文在 ATIS 数据集上进行了更多的试验次数，对基线模型进行了 14 次独立训练，而对改进的模型上进行了 13 次独立训练。

本文对基线模型与改进的模型的实验结果进行了对比。为了探究预训练模型的有无对意图识别与槽填充任务的重要性，本文还从 Goo 等人[17]的文献中摘录了几种未使用预训练模型的方法的结果与上述两种模型进行对比。它们分别为 Hakkani-Tür 等人[32]基于 LSTM 的方法，Liu 和 Lane[14]基于 LSTM 与注意力机制的方法，以及 Goo 等人[17]利用意图识别的结果对槽填充过程进行限制的方法。对比实验的结果如表 3 所示。通过对比本文所述的改进模型与基线模型之间的性能，可以发现在 ATIS 数据集上本文所述方法在槽填充性能和双任务联合性能上较基线模型具有一定的优势。改进模型在 ATIS 数据集上的意图准确率略低于基线模型，这可能是由于双任务联合模型对槽识别任务更加敏感。而在 Snips 数据集上，三个评测指标均较基线模型具有显著优势。而通过对比基于 BERT 的方法与未使用预训练模型的方法，可以发现 BERT 对意图识别与槽填充两个任务的提升尤为明显。

为了验证本文改进的模型相对于基线模型在句子级准确率上提升的显著性，本文对多次独立训练得到的结果应用了单侧 Welch 检验。其结果如表 4 所示。可以发现，在句子级准确率方面，改进的模型相对于基线模型的提升具有显著性。

表 4 显著性检验结果

数据集	模型	样本数量	样本均值	样本方差	显著性水平
ATIS	BERT+CRF（Chen 等[24]，基线模型）	14	0.8812	9.6215×10^{-6}	0.0409
ATIS	BERT+LSTM+注意力+CRF（本文）	13	0.8833	6.5843×10^{-6}	
Snips	BERT+CRF（Chen 等[24]，基线模型）	5	0.9089	5.8366×10^{-5}	0.0045
Snips	BERT+LSTM+注意力+CRF（本文）	5	0.9251	1.9930×10^{-5}	

为了验证本文改进的模型以及两种调优方法各自对意图识别与槽填充任务性能的影响，本文还对解码器选择，是否执行参数锁定与是否区分大小写三个参数的所有组合进行了实验，其结果如表 5 所示。可以发现，使用本文所述的改进模型在两个数据集上较基线模型均可以获得一定程度的提升。对于包含大小写信息的 Snips 数据集，使用区分大小写的 BERT 预训练模型版本可以获得显著的性能提升。而锁定参数可以在两个数据集的意图识别任务中获得一定程度的提升。

表 5 组合实验结果

数据集	解码器	锁定参数	大小写	意图准确率	槽 F1	句子级准确率	最优轮次数
ATIS	CRF	无	不区分	0.9758	0.9583	0.8812[3]	6
ATIS	CRF	词嵌入+1 层	不区分	**0.9776**	0.9591	**0.8829**	8
ATIS	LSTM+注意力+CRF	无	不区分	0.9738	**0.96**	**0.8833**[3]	30
ATIS	LSTM+注意力+CRF	词嵌入	不区分	0.9742	0.9594	0.8809	40
Snips	CRF	无	不区分	0.9877	0.9618	0.9089	13
Snips	CRF	无	区分	0.9886	0.9639	0.9183	13
Snips	CRF	词嵌入	不区分	0.9883	0.96	0.9043	13
Snips	CRF	词嵌入	区分	0.9883	0.9634	0.9186	13
Snips	LSTM+注意力+CRF	无	不区分	0.9883	0.9624	0.9077	25
Snips	LSTM+注意力+CRF	无	区分	0.9906	**0.9678**	**0.9251**	25
Snips	LSTM+注意力+CRF	词嵌入	不区分	0.9889	0.961	0.9063	25
Snips	LSTM+注意力+CRF	词嵌入	区分	**0.9909**	0.9643	0.9197	25

一个例外是 Snips 数据集上基线模型区分大小写的测试，这个测试中锁定参数未能带来意图准确率的提升，这可能是由于模型发生了欠拟合。基线模型的参数数量较小，锁定参数进一步减少了可训练参数的数量。而 Snips 数据集的样本量较大，区分大小写同时增加了样本的复杂程度。在这种情况下，模型发生欠拟合的风险会有所提高。

图 5 给出了 ATIS 数据集上基线模型交叉验证中句子级准确率与锁定参数层数的关系，可以发现锁定参数的层数有一个最优值，不论过少还是过多均会导致模型性能的下降。我们推测这是由于锁定层数过少时模型发生了过拟合而丢失了一些 BERT 中与任务无关的通用信息，而锁定层

数过多时 BERT 携带了过多与预训练任务高度相关的无用信息。

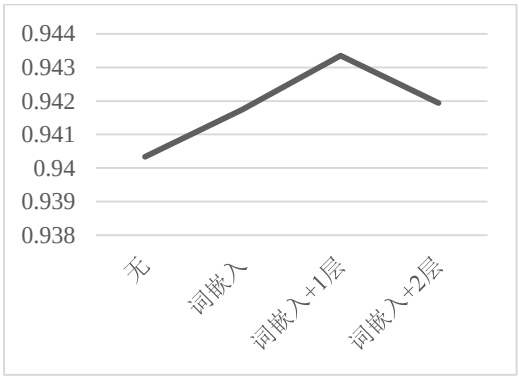

图 5 句子级准确率与锁定参数层数的关系

表 6 给出了一个不区分大小写的模型在测试集上的一个槽填充错误样本。在这个样本中，《A Very Cellular Song》是一个歌名，这一事实是非常容易根据它的大小写发现的。不区分大小写的模型无法捕捉这一信息，因此进行了错误的猜测。而区分大小写的模型则轻易给出了正确的结果。

表 6 不区分大小写模型错误样本

用户询问	预测槽标签	正确槽标签
add	O	O
A	**O**	**B-entity_name**
Very	**B-artist**	**I-entity_name**
Cellular	**I-artist**	**I-entity_name**
Song	**B-music_item**	**I-entity_name**
to	O	O
masters	B-playlist	B-playlist
of	I-playlist	I-playlist
metal	I-playlist	I-playlist
playlist	O	O

6 结论

本文提出了一种改进的自然语言理解模型，其编码器使用 BERT，而解码器基于 LSTM 与注意力机制构建。接下来，本文还提出了该模型的两种调优方法：锁定模型参数的训练方法、区分大小写的预训练模型版本。实验发现，基于 LSTM 与注意力机制的解码器可以提高任务导向对话系统自然语言理解模型的性能。在此基础上，锁定模型参数的训练方法可以提高意图识别任务的性能，而区分大小写的预训练模型版本可以提高两个任务的性能。

参考文献

[1] Devlin J, Chang M-W, Lee K, et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[J]. arXiv:1810.04805 [cs], 2018.
[2] Tur G, Hakkani-Tur D, Heck L. What is left to be understood in ATIS?[C]//2010 IEEE Spoken Language Technology Workshop. Berkeley, CA, USA: IEEE, 2010: 19–24.
[3] Coucke A, Saade A, Ball A, et al. Snips Voice Platform: an embedded Spoken Language Understanding system for private-by-design voice interfaces[J]. arXiv:1805.10190 [cs], 2018.
[4] Mesnil G, He X, Deng L, et al. Investigation of Recurrent-Neural-Network Architectures and Learning Methods for Spoken Language Understanding[C]//Interspeech 2013. 2013.
[5] Yao K, Zweig G, Hwang M-Y, et al. Recurrent Neural Networks for Language Understanding[C]//Interspeech, 2013.
[6] Kim J-K, Tur G, Celikyilmaz A, et al. Intent detection using semantically enriched word embeddings[C]//2016 IEEE Spoken Language Technology Workshop (SLT). San Diego, CA: IEEE, 2016: 414–419.
[7] Yao K, Peng B, Zhang Y, et al. Spoken language understanding using long short-term memory neural networks[C]//2014 IEEE Spoken Language Technology Workshop (SLT). South Lake Tahoe, NV, USA: IEEE, 2014: 189–194.
[8] Mensio M, Rizzo G, Morisio M. Multi-turn QA: A RNN Contextual Approach to Intent Classification for Goal-oriented Systems[C]//WWW '18: Companion Proceedings of the The Web Conference 2018. Lyon, France: ACM Press, 2018: 1075–1080.
[9] Deoras A, Sarikaya R. Deep Belief Network based Semantic Taggers for Spoken Language Understanding[C]//ISCA Interspeech. ISCA, 2013.
[10] Deng L, Tur G, He X, et al. Use of kernel deep convex networks and end-to-end learning for spoken language understanding[C]//2012 IEEE Spoken Language Technology Workshop (SLT). 2012: 210–215.
[11] Xu P, Sarikaya R. Convolutional neural network based triangular CRF for joint intent detection and slot filling[C]//2013 IEEE Workshop on Automatic Speech Recognition and Understanding. Olomouc, Czech Republic: IEEE, 2013: 78–83.
[12] Jeong M, Lee G G. Triangular-Chain Conditional Random Fields[J]. IEEE Transactions on Audio, Speech, and Language Processing, 2008, 16(7): 1287–1302.
[13] Zhang X, Wang H. A Joint Model of Intent Determination and Slot Filling for Spoken Language Understanding[C]//Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence. New York, New York, USA: AAAI Press, 2016: 2993–2999.
[14] Liu B, Lane I. Attention-Based Recurrent Neural Network Models for Joint Intent Detection and Slot Filling[C]//2016: 685–689.
[15] Schumann R, Angkititrakul P. Incorporating ASR Errors with Attention-Based, Jointly Trained RNN for

Intent Detection and Slot Filling[C]//2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Calgary, AB: IEEE, 2018: 6059–6063.
[16] Zhang H, Zhu S, Fan S, et al. Joint Spoken Language Understanding and Domain Adaptive Language Modeling[G]//Peng Y, Yu K, Lu J, et al. Intelligence Science and Big Data Engineering. Cham: Springer International Publishing, 2018, 11266: 311–324.
[17] Goo C-W, Gao G, Hsu Y-K, et al. Slot-Gated Modeling for Joint Slot Filling and Intent Prediction[C]//Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). New Orleans, Louisiana: Association for Computational Linguistics, 2018: 753–757.
[18] Peters M, Neumann M, Iyyer M, et al. Deep Contextualized Word Representations[C]//Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). New Orleans, Louisiana: Association for Computational Linguistics, 2018: 2227–2237.
[19] Radford A, Narasimhan K, Salimans T, et al. Improving Language Understanding by Generative Pre-Training[J]. 2018: 12.
[20] Radford A, Wu J, Child R, et al. Language Models are Unsupervised Multitask Learners[J]. 2019: 24.
[21] Lee J, Yoon W, Kim S, et al. BioBERT: a pre-trained biomedical language representation model for biomedical text mining[J]. arXiv:1901.08746 [cs], 2019.
[22] Adhikari A, Ram A, Tang R, et al. DocBERT: BERT for Document Classification[J]. arXiv:1904.08398 [cs], 2019.
[23] Nogueira R, Cho K. Passage Re-ranking with BERT[J]. arXiv:1901.04085 [cs], 2019.
[24] Chen Q, Zhuo Z, Wang W. BERT for Joint Intent Classification and Slot Filling[J]. arXiv:1902.10909 [cs], 2019.
[25] Vaswani A, Shazeer N, Parmar N, et al. Attention is All you Need[G]//Guyon I, Luxburg U V, Bengio S, et al. Advances in Neural Information Processing Systems 30. Curran Associates, Inc., 2017: 5998–6008.
[26] Hochreiter S, Schmidhuber J. Long Short-Term Memory[J]. Neural Computation, 1997, 9(8): 1735–1780.
[27] Gers F A. Learning to forget: continual prediction with LSTM[C]//9th International Conference on Artificial Neural Networks: ICANN '99. Edinburgh, UK: IEE, 1999, 1999: 850–855.
[28] Greff K, Srivastava R K, Koutnik J, et al. LSTM: A Search Space Odyssey[J]. IEEE Transactions on Neural Networks and Learning Systems, 2017, 28(10): 2222–2232.
[29] Bahdanau D, Cho K, Bengio Y. Neural Machine Translation by Jointly Learning to Align and Translate[J]. arXiv:1409.0473 [cs, stat], 2014.
[30] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: A Simple Way to Prevent Neural Networks from Overfitting[J]. Journal of Machine Learning Research, 2014, 15: 1929–1958.
[31] Kingma D P, Ba J. Adam: A Method for Stochastic Optimization[J]. arXiv:1412.6980 [cs], 2014.
[32] Hakkani-Tür D Z, Tür G, Çelikyilmaz A, et al. Multi-Domain Joint Semantic Frame Parsing Using Bi-Directional RNN-LSTM[C]//INTERSPEECH. 2016.

周奇安（1995-），硕士研究生，主要研究领域为自然语言处理。

E-mail：jiefangxuanyan@buaa.edu.cn

李舟军（1963-），通信作者，博士，教授，博士生导师。CCF 高级会员，主要研究领域为数据挖掘与自然语言处理、网络与信息安全。

E-mail：lizj@buaa.edu.cn