基于多层次预训练策略和多任务学习的端到端蒙汉语音翻译

王宁宁[1,2,3]，飞龙[1,2,3*]，张晖[1,2,3]

内蒙古大学计算机学院，呼和浩特 010021[1]
蒙古文智能信息处理技术国家地方联合工程研究中心，呼和浩特 010021[2]
内蒙古自治区蒙古文信息处理技术重点实验室，呼和浩特 010021[3]
810190797@qq.com，csfeilong@imu.edu.cn，cszh@imu.edu.cn

摘要

端到端语音翻译将源语言语音直接翻译为目标语言文本，它需要“源语言语音-目标语言文本”作为训练数据，然而这类数据极其稀缺，本文提出了一种多层次预训练策略和多任务学习相结合的训练方法，首先分别对语音识别和机器翻译模型的各个模块进行多层次预训练，接着将语音识别和机器翻译模型连接起来构成语音翻译模型，然后使用迁移学习对预训练好的模型进行多步骤微调，在此过程中又运用多任务学习的方法，将语音识别作为语音翻译的一个辅助任务来组织训练，充分利用了已经存在的各种不同形式的数据来训练端到端模型，首次将端到端技术应用于资源受限条件下的蒙汉语音翻译，构建了首个翻译质量较高、实际可用的端到端蒙汉语音翻译系统。

关键词： 蒙古语；端到端语音翻译；预训练；多任务学习

End-to-end Mongolian-Chinese Speech Translation Based on Multi-level Pre-training Strategies and Multi-task Learning

Ningning Wang[1,2,3], Feilong Bao[1,2,3*], Hui Zhang[1,2,3]

College of Computer Science Inner Mongolia University, Hohhot 010021, China[1]
National & Local Joint Engineering Research Center of
Intelligent Information Processing Technology for Mongolian, Hohhot 010021, China[2]
Inner Mongolia Key Laboratory of Mongolian Information Processing Technology, Hohhot 010021, China[3]
810190797@qq.com, csfeilong@imu.edu.cn, cszh@imu.edu.cn

Abstract

End-to-end speech translation interprets the speech in source language to text in target language. Training the end-to-end speech translation models needs “source-speech - target-text” data, while this type of data is for limited. This study proposes a training method that combines multi-level pre-training strategy and multi-task learning. First, perform multi-level pre-training on each module of the speech recognition and machine translation models, then connect the speech recognition and machine translation models to form a speech translation model, and then use transfer learning to fine-tune the pre-trained model in multiple steps.In this process, the method of multi-task learning is used, speech recognition is used as an auxiliary task of speech translation to organize training, and the existing various forms of data are fully used to train the end-to-end model. We first apply the end-to-end method in the Mongolian Chinese speech translation task, and build the first practical system with high performance.

Keywords: Mongolian , End-to-end speech translation , Pre-training , Multi-task learning

1 引言

语音翻译即语音到语音的翻译，它把用一种语言说出来的话翻译成另外一种语言说出来。语音翻译是不同语言使用者之间最自然的交流方式。近年来，随着蒙古语智能信息处理技术研究的不断深入：蒙古语语音识别技术(Wang et al., 2017)、蒙汉机器翻译技术(Sun et al., 2020)、语音合成技术(Liu et al., 2020)取得了长足发展，蒙汉语音翻译系统似乎已经触手可得。然而这种采用蒙古语语音识别将源语音转录为文本，再使用蒙汉机器翻译将源语言文本翻译成目标语言，最后利用语音合成输出目标语音的级联方法存在着时间延迟、参数冗余和错误累积等问题。

端到端语音翻译技术可以直接从源语言的语音生成翻译文本，它使用一个模型完成语音识别和机器翻译任务，所有参数会根据最终目标共同优化，从而可以在一定程度上缓解级联方法的问题。然而端到端语音翻译技术也存在自己的问题，其中最主要的问题是数据限制：根据语音翻译的任务需求，其训练输入是源语言的语音，训练目标是目标语言的文本，这样的"源语音，目标文本"对的数据获取成本很高，已有的数据难以形成规模；而端到端语音翻译模型往往参数量巨大，又少不了大规模的训练数据。

如何打破数据限制是端到端方法面临的最大挑战。最近的研究提出了多任务学习（Multi-Task Learning，MTL）、知识蒸馏和预训练等技术来缓解语音翻译中训练数据缺乏问题。例如，(Weiss et al., 2017)使用多任务学习的策略来训练端到端语音翻译模型，将语音识别作为一个辅助任务，将输入语音的转录文本作为语音识别的训练目标，语音识别和语音翻译这两个任务共享一个编码器，分别使用两个不同的解码器，提升了语音翻译的质量。(Liu et al., 2019)将知识蒸馏方法运用于端到端语音翻译，首先在平行文本数据上训练一个机器翻译模型用作教师模型，然后通过转移知识来指导语音翻译模型的训练。(Bansal et al., 2019)在端到端语音翻译中使用了语音识别预训练得到的编码器，该方法可以提升低资源条件下的语音翻译的性能。(Berard et al., 2018)使用了大规模语音识别和机器翻译数据分别预训练了语音翻译和机器翻译模型，这一策略可以带来更快的收敛速度和更好的结果。

相比于英语、汉语等主流语种，蒙古语语料规模尚小，满足端到端语音翻译模型训练的"蒙古语语音-汉语文本"数据仅有300小时左右，仅使用这些数据得到的端到端语音翻译模型还比不上简单的级联方法。为了提升端到端蒙汉语音翻译的性能，就需要充分挖掘现有存量数据，将尽可能多的有效知识转移到端到端语音翻译模型当中。迁移学习作为一种很实用的机器学习方法，可以实现将有效知识从某一个模型转移到另一个模型的目的，通常迁移学习是以预训练模型作为载体来实现的，可以让预训练模型的训练成果得到很好的拓展使用或者再利用，所以结合现有数据储备和前人提出的有效解决方案，本文提出了一种多层次、多步骤的迁移学习和预训练策略，从而打破了端到端语音翻译的数据限制，同时也得到了一个翻译质量较高、实际可用的端到端蒙汉语音翻译系统。

本文的主要贡献是：（1）提出了一套多层次、多步骤的迁移学习和预训练策略，充分挖掘了各种形式的存量数据的价值，实验表明这是一套有效的技术方案。（2）本文首次将端到端技术应用于蒙汉语音翻译，构建了首个翻译质量较高、实际可用的端到端蒙汉语音翻译系统。

2 端到端语音翻译模型

端到端语音翻译将源语言中的语音信号直接转换为目标语言中的文本。本文将源语言的语音信号转换为声学特征序列，并将其定义为$\mathbf{x}^s = (x_1^s, ..., x_T^s)$；对应的源语言文本和相应的目标语言文本分别定义为$\mathbf{y}^s = (y_1^s, ..., y_{T_s}^s)$和$\mathbf{y}^t = (y_1^t, ..., y_{T_t}^t)$。在本文的工作中，假设训练时存在一个小规模的数据集$(\mathbf{x}^s, \mathbf{y}^s, \mathbf{y}^t)$，其中的$\mathbf{y}^s$用作语音识别辅助任务的训练目标。预测时输入一个新的源语言的语音特征序列$\mathbf{x}^s$，得到一个目标语言的文本$\mathbf{y}^t$的预测$\hat{\mathbf{y}}^t$。上标s标记源语言（source），对应的上标t标记目标语言（target）。

本文使用(Wang et al., 2020)提出的串联连接编码网络（Tandem Connectionist Encoding Network，TCEN）作为端到端蒙汉语音翻译模型的骨架，并对其进行适应性修改使其成为最终使用的模型。

TCEN由一个语音识别模型（ASR）和一个机器翻译模型（MT）串联而成。语音识别模型和机器翻译模型都由多层双向LSTM构成，其中机器翻译模型的编码器和解码器之间还包含注意力连接。在TCEN中，语音识别模型仅使用CTC（Connectionist Temporal Classification）(Graves et al., 2006)训练，在本文的研究中，我们将基于CTC的语音识别模型替换为更

优的基于RNN-T（Recurrent Neural Network Transducer）(Graves, 2012)的语音识别模型，我们将其命名为TCEN-RNNT。

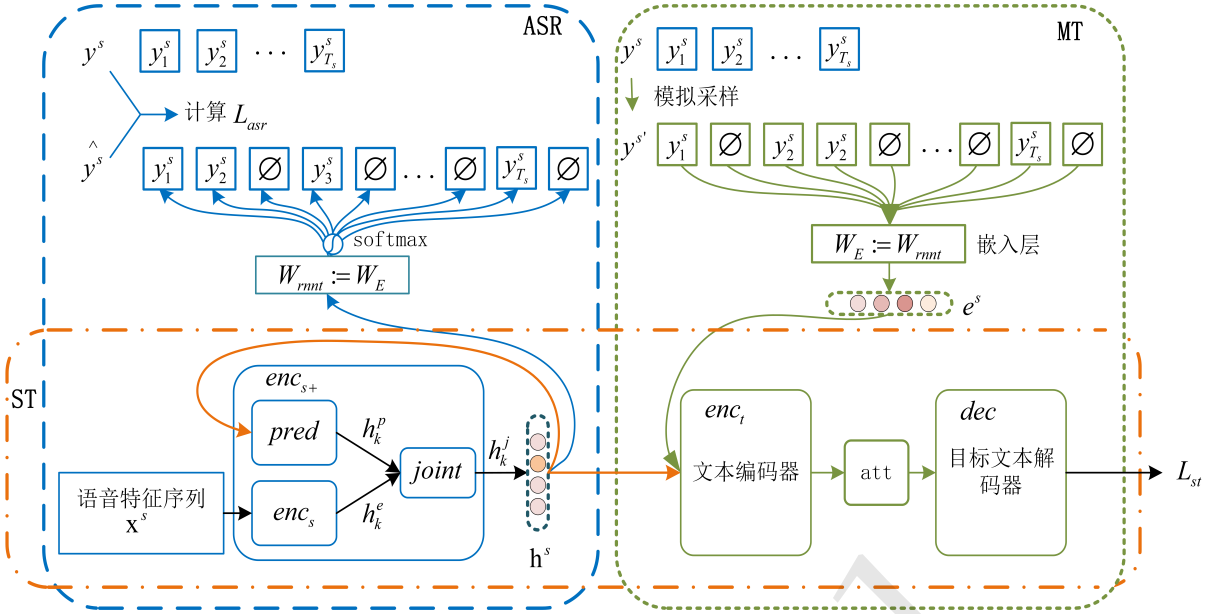

图 1.TCEN-RNNT模型的整体结构

图1展示了TCEN-RNNT的整体结构，左侧是语音识别ASR模块，源语言的语音特征$\mathbf{x}^s$输入ASR模块，经过增强的语音编码器enc_{s+}编码转换为一系列源语言词编码形式$\mathbf{h}^s$，即词向量序列。

$$\mathbf{h}^s = enc_{s+}(\mathbf{x}^s) \tag{1}$$

之后$\mathbf{h}^s$经过一个线性变换层W_{rnnt}和softmax变换为词概率，经解码后即可得到识别后的源语言文本。在本文提出的模型中，我们是使用了RNN-T作为识别模型，因此增强的语音编码器enc_{s+}本身包含三个部分：一个由多层双向LSTM构成的语音编码器enc_s，一个由LSTM构成的预测器$pred$和一个简单线性层构成的连接器$joint$。其中语音编码器enc_s，接受语音特征输入，预测词向量输出；预测器接受前一次的输出作为输入产生本次预测，其作用类似于一个语言模型；连接器将来源于语音编码器和预测器的输出组合起来。设模型的输入输出为一个时序向量，如$\mathbf{x}^s = (x_0^s, \ldots, x_k^s, \ldots)$，$x_k^s$表示第$k$帧输入语音特征，$h_k^e$，$h_k^p$，$h_k^j$分别是第$k$时刻编码器、预测器和连接器的输出，整个增强的语音编码器enc_{s+}的输出由h_k^j构成，即$\mathbf{h}^s = (h_0^j, \ldots, h_k^j, \ldots)$。

$$h_k^e = enc_s(x_k^s) \tag{2}$$

$$h_k^p = pred(h_{k-1}^j) \tag{3}$$

$$h_k^j = joint(h_k^e, h_k^p) \tag{4}$$

$$P(\hat{y_k^s}|\hat{y_{<k}^s}, \mathbf{x}^s) = softmax(W_{rnnt} \cdot h_k^j) \tag{5}$$

在原始的RNN-T中，预测器的输入是上一步的预测结果，即经过线性变换层W_{rnnt}和softmax的输出。而在本文的工作中，由于语音识别模块与机器翻译模块连接时会舍弃W_{rnnt}和softmax层，如果再将预测输出作为预测器的输入，那么在连接后的模型中，预测器将会失去输入，因此，我们将预测器的输入调整为连接层$joint$的输出，即W_{rnnt}层的输入，跳过了W_{rnnt}和softmax层直接输入到预测器中（图中橙色箭头）。我们认为原始版本

的RNN-T中的预测器是由前一个词预测下一个词的语言模型，而在新模型中的预测器是一个由前一个词的词向量预测下一个词的词向量的语言模型，其本质作用并没有发生改变。

图1右侧是机器翻译MT模块，语音识别模块输出的词编码向量$\mathbf{h}^s$输入到翻译模型中，经过文本编码器enc_t编码后由目标语言解码器dec产生目标语言的文本输出。在编码器与解码器之间由一个注意力连接att解决对齐问题。

$$\mathbf{h}^t = enc_t(\mathbf{h}^s) \tag{6}$$

$$c_k = att(z_{k-1}, \mathbf{h}^t) \tag{7}$$

$$z_k = dec(z_{k-1}, y^t_{k-1}, c_k) \tag{8}$$

$$P(\hat{y^t_k}|\hat{y^t_{<k}}, \mathbf{x}^s) = softmax(W \cdot z_k) \tag{9}$$

其中$P(\hat{y^t_k}|\hat{y^t_{<k}}, \mathbf{x}^s)$是机器翻译模型的输出，解码后即为目标语言文本，$z_k$是解码器在时刻$k$的状态，$c_k$是动态上下文向量。若将机器翻译模型单独分离出来，文本编码器enc_t的输入变为源语言的文本词向量序列$\mathbf{e}^s$，词向量由一个embeding层提供，设其参数为W_E，即源语言文本的词向量编码矩阵。

上述的端到端语音翻译模型TCEN-RNNT，可以直接训练，需要的提供的训练数据为源语言语音特征$\mathbf{x}^s$和对应的目标语言文本$\mathbf{y}^t$。如前所述，获取“源语言语音-目标语言文本”，$(\mathbf{x}^s, \mathbf{y}^t)$，较为困难，因此采用将语音识别模块和机器翻译模块分别训练，之后再将两个模块连接在一起，使用$(\mathbf{x}^s, \mathbf{y}^t)$形式的数据进行微调即可。

语音识别模块和机器翻译模块分别训练之后再进行连接时，想要不丢失预训练的成果，需要保证机器翻译模型MT模块在单独训练时和连接成为整体后的输入是匹配的，想要保证这种匹配，需要更进一步要求单独训练的语音识别模型ASR模块的输出与机器翻译模块的输入是匹配的。

原始的语音识别模块和机器翻译模块的训练可能会出现两种情况的不匹配，首先，语音识别输出的$\mathbf{h}^s$是源语言文本的某种词向量表示，但是这种词向量表示必须与训练机器翻译模块所使用的词向量表示是一致的，否则语音识别模块的输出相对于机器翻译模块而言还是“外语”，连接起来之后，机器翻译模型还需要学习如何迁就语音识别模块所使用的词向量表示，这会破坏机器翻译模块的预训练成果。这时可以通过将语音识别模块的线性变换层W_{rnnt}与机器翻译的源语言文本的词向量编码矩阵W_E绑定，这时语音识别模块要想产生某个词作为输出，就必须将h^j_k输出为这个词的词向量。

其次，语音识别模块输出的$\mathbf{h}^s$是一个CTC解码输入，其中会包含重复的词向量和空白输出标记。而在原始的机器翻译模型的训练中输入的是源语言文本的词向量序列，其中不包含重复和空白。为弥合这种差别，我们在单独训练机器翻译模块时，对源语言文本$\mathbf{y}^s$做模拟采样，以一定的概率产生重复并插入空白标记，并将经过模拟采样后的源语言文本标记为$\mathbf{y}^{s'}$。

类似地，本文也将语音编码器中的预测器$pred$部分当作一个语言模型单独预训练，输入前一个词的词向量预测后一个词的词向量，这时，我们会使用与机器翻译模块相同的源语言文本的词向量编码矩阵W_E，并且对训练数据做模拟采样，以一定的概率产生重复并插入空白标记，模拟语音识别的输出。

以上描述的在级联连接的端到端语音翻译模型中保留两个模块预训练成果的策略，来源于(Wang et al., 2020)，请读者参看原始论文，以获取更多信息。

除使用预训练策略外，本研究还使用多任务学习策略(Luong et al., 2016)提升语音翻译的性能。本文将语音识别作为一个辅助任务，在训练集$(\mathbf{x}^s, \mathbf{y}^s, \mathbf{y}^t)$上，输入源语言特征$\mathbf{x}^s$，经语音识别模块得到源语言文本词向量序列$\mathbf{h}^s$，经线性变换$W_{rnnt}$和softmax后，得到语音识别结果$\hat{\mathbf{y}^s}$，与源语言文本$\mathbf{y}^s$计算交叉熵损失作为语音识别部分的代价。将$\mathbf{h}^s$输入到机器翻译模块，获取输出的翻译结果$\hat{\mathbf{y}^t}$，与目标语言文本$\mathbf{y}^t$计算交叉熵损失作为翻译部分的代价。最后将两个部分的代价求和作为整个语音翻译任务的代价:

$$L_{total} = \alpha_{st} \cdot L_{st} + \alpha_{asr} \cdot L_{asr} \tag{10}$$

3 迁移学习和预训练策略

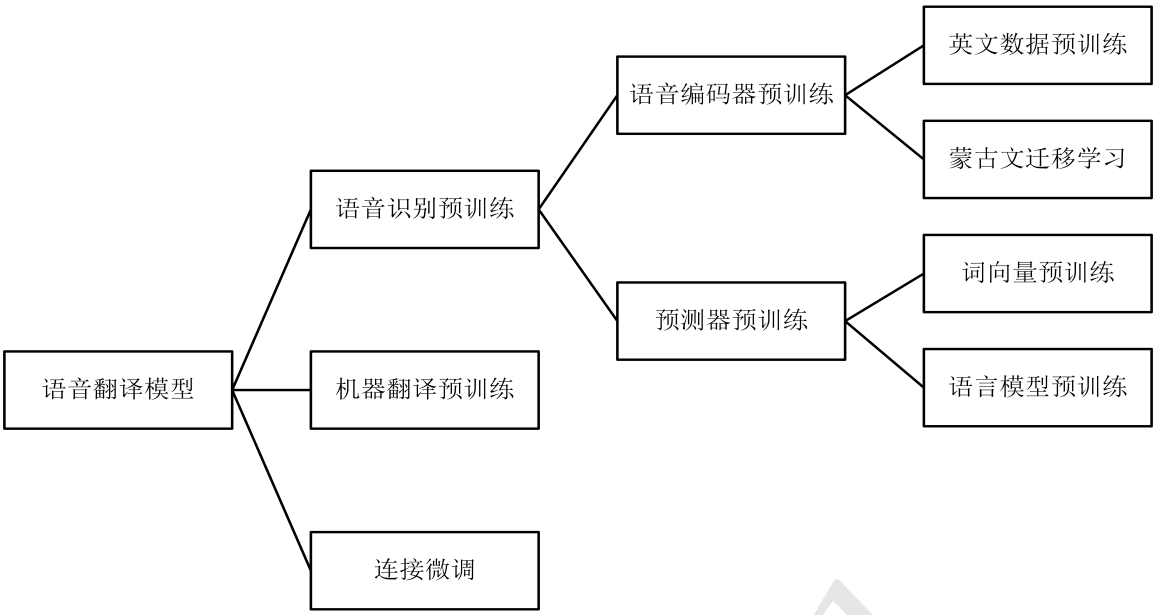

图 2.多层次、多步骤的迁移学习和预训练策略

图2展示了提出的多层次、多步骤的迁移学习和预训练策略的框架。总的来看，整个端到端蒙汉语音翻译模型的训练过程分作三步：首先单独训练一个语音识别模块，之后单独训练一个机器翻译模块，最后将两个模块连接起来，做整体的微调。

增强的语音编码器enc_{s+}需要对其语音编码器和预测器分别做预训练，其中语音编码器部分，本文进一步采用迁移学习策略，先使用英文数据训练语音编码器，之后将输出层重设为蒙古文的输出层，使用蒙古语语音数据以较小的学习率更新语音识别编码器参数，得到性能更好的蒙古语语音识别编码器。预测器部分，使用前一个词的词向量预测后一个词的词向量，首先需要获取词向量表示，本文在大规模蒙古语文本数据上，训练得到一个词向量编码表，将所有训练数据编码转换为词向量后，对得到的词向量序列进行模拟采样，以一定的概率插入重复的词向量和代表空白标记的词向量，使用这一数据完成语音识别预测器的训练。之后将语音编码器和预测器通过连接变换层$joint$连接起来，重新使用“蒙古语语音-文本”数据对语音识别模块进行微调。

机器翻译模型预训练时，使用前面训练得到的词向量编码表，并对源语言文本进行模拟采样，以一定的概率插入重复的词向量和代表空白标记的词向量，使用这一数据作为输入，目标语言文本作为训练目标，完成机器翻译模块训练。

最后移除机器翻译模块的输入部分，将语音识别模块的输出连接到机器翻译模块的输入上，使用“蒙古语语音、蒙古文文本、汉语文本”作为训练数据，微调整个模型。

相比于原来的端到端训练方法，提出的多层次、多步骤的迁移学习和预训练策略引入了更多的复杂性，但是这种方法打破了端到端语音翻译对训练数据的限制，原来我们只能使用“源语言语音-目标语言文本”这样的训练数据，而在提出的训练方案中，我们可以使用其他语言（如英语）的“语音-文本”数据、源语言的“语音-文本”数据、单纯的源语言文本数据、“源语言文本-目标语言文本”数据，这些数据中的有效知识在训练过程中被转移到模型中。如果不使用这样的策略，想要达到同样的效果，所有需要的知识必须都包含在“源语言语音-目标语言文本”这样的数据中，这需要极大规模的数据量，获取这样的数据既有较高的时间和经济成本，也浪费了现有的存量数据。

下一节，本文将通过实验证明提出的方案的有效性。

4 实验

4.1 数据集

本文描述的工作，涉及多种不同形式的数据集，这些数据都被用于提升整个语音翻译系统

的性能，使用的数据集总体情况列于表1中。其中第1、2、3、4项数据是端到端训练时无法使用的，在本文的工作中，我们使用这些数据做预训练和迁移学习。第5、6、7项数据是蒙古语语音-蒙古语文本-汉语文本数据集，这是端到端模型的训练数据，总计约300小时22万句对。我们将其划分为训练集、测试集和开发集，它们之间没有重叠。

序号	数据形式	规模	应用
1	蒙古语文本	370万句	源语言词向量、语音识别模块预测器训练
2	英语语音-文本	1000小时	语音识别模块编码器训练
3	蒙古语语音-文本	700小时	语音识别模块编码器、语音识别模块训练
4	蒙古语-汉语文本	266万句对	机器翻译模块训练
5	蒙古语语音-汉语文本	280小时，20万句对	语音翻译训练集
6	蒙古语语音-汉语文本	13小时，1万句对	语音翻译测试集
7	蒙古语语音-汉语文本	7小时，5千句对	语音翻译开发集

表 1. 数据集

4.2 训练过程

本文提出的多层次、多步骤的迁移学习和预训练策略分以下步骤施行：

第一步，蒙古文的词向量由内蒙古大学蒙古文信息处理技术重点实验室收集的总计370万句的蒙古语文本语料，使用python版本的Word2vec工具(Mikolov et al., 2013)训练得到。通过上述Word2vec工具训练得到的蒙古文的词向量将用于语音识别模块的预测器训练和语音识别模块整体的训练，以及机器翻译模块的训练。为降低蒙古文的词汇量，我们使用Byte Pair Encoding（BPE）(Sennrich et al., 2016)技术进行了切词处理；为降低机器翻译模型困惑度，文本数据中不包含长度大于50个词的句子。

第二步，上面用到的这370万句蒙古语文本语料由上一步训练得到的词向量编码表转换为词向量表示，之后利用模拟采样方法插入重复和空白，训练语音识别模块的预测器。

第三步，使用包含大约1000小时的英语语音-英语文本的LibriSpeech语料库(Panayotov et al., 2015)训练一个英语语音识别器。在本文的工作中使用梅尔频率倒谱系数（Mel-Frequency Cepstral Coefficients, MFCC）(Muda et al., 2010)作为声学特征，特征提取时帧长设为25 ms，帧移10 ms，MFCC保留40维梅尔倒谱系数，并通过减均值除标准差做归一化处理。

第四步，使用由内蒙古大学蒙古文信息处理技术重点实验室提供的总时长约700小时的蒙古语语音-蒙古语文本数据集，在上一步得到的英语语音识别器的基础上训练得到蒙古语语音识别器。

第五步，将第二步训练得到的语音识别模块的预测器和第四步训练得到的语音识别模块的语音编码器，通过连接层*joint*连接起来，使用前一步使用的700小时蒙古语语音-文本数据训练语音识别模块，在最初的几次训练中，仅有连接层的权重被调整。

第六步，使用由内蒙古大学蒙古文信息处理技术重点实验室提供的240万句和CWMT 2017提供的26万句蒙古语文本-汉语文本数据集，共计266万句蒙古语-汉语文本数据集训练机器翻译模型，其中输入做模拟采样插入重复和空白标记。

第七步，移除上一步训练得到的蒙汉机器翻译模型的输入，将第五步训练得到的蒙古语语音识别模块的输出连接到机器翻译模块，使用由内蒙古大学蒙古文信息处理技术重点实验室提供的约280小时，20万句对的蒙古语语音-蒙古语文本-汉语文本数据集训练语音翻译模型，其中蒙古语语音识别将作为辅助任务。

4.3 模型实现

基于RNN-T语音识别模型架构的语音编码器网络由8个双向LSTM层组成，其中每层有2048个隐藏单元（即，在正向和反向各有1024个隐藏单元）和一个640维的投影层。预测

器网络是2个双向LSTM层，其中每层有2048个隐藏单元和一个640维的投影层。编码器和预测器的输出被馈送到具有640个隐藏单元的连接器网络。连接器网络由两个线性层组成，其中每层具有640个隐藏单元，并且没有任何激活函数。

机器模型架构的编码器网络由5个双向LSTM层组成，其中每层有2048个隐藏单元（即，在正向和反向各有1024个隐藏单元）。解码器网络由两个单向LSTM层和一个注意力机制组成，其中每个单向LSTM层有1024个隐藏单元，注意力层的维度为1024。

模型使用ESPnet工具包(Watanabe et al., 2018)实现。采用dropout方法来降低过拟合风险，dropout值为0.2。使用Adadelta作为优化器，初始学习率为1.0。所有模型都在4个Tesla P40 GPU上训练50个epochs。

通过在开发集上的实验，本文确定多任务训练时总的代价函数中的翻译部分的权重$\alpha_{st} = 0.6$，语音识别部分的权重$\alpha_{asr} = 0.2$。

在本文中，通过报告BLEU(Papineni et al., 2002)作为语音翻译的性能指标，BLEU值越高表示性能越好。

4.4 对比方法

本文的方法将与以下的方法进行比较：

（1）端到端方法：参照(Bérard et al., 2016)实现，由基于注意力的编码器-解码器结构组成，而且使用280小时的蒙古语语音-汉语文本数据集从头开始训练。

（2）级联方法：由1000小时的蒙古语语音-蒙古语文本数据集独立训练的语音识别模型和由266万句蒙古语文本-汉语文本数据集独立训练的机器翻译模型级联而成，注意该方法使用的蒙古语语音-文本数据要多于提出的方法，它是目前最优的蒙古语语音识别器，词错误率（WER）为8.29。

（3）TCEN方法：参照(Wang et al., 2020)实现，通过串联一个基于CTC的语音识别模型编码器和一个机器翻译模型编码器，后面再接一个基于注意力的机器翻译模型解码器的方式构成模型的整体框架。TCEN使用与本文TCEN-RNNT相同的数据集进行训练（不包含预测器所涉及的数据集）。

（4）机器翻译方法：用266万句蒙古语文本-汉语文本数据集训练机器翻译模型。我们直接使用蒙古文文本作为输入，理论上它的性能是语音翻译的性能上限。

4.5 实验结果

方法	BLEU
端到端	8.4
级联	16.32
TCEN	18.27
TCEN-RNNT	**19.64**
机器翻译	35.86

表 2. TCEN-RNNT与各对比方法的性能比较

实验结果列于表2中，从表中数据可以看出随着数据使用量的增加，语音翻译的性能也在逐步提高。直接训练端到端语音翻译模型可用的训练数据最少，实验结果表明仅用少量的蒙古语语音-汉语文本数据集难以直接训练端到端语音翻译模型，甚至无法达到直接级联的效果。级联方法使用了蒙古语语音-文本数据训练语音识别器，蒙古语文本-汉语文本训练机器翻译模型，整体连接取得了较好的性能，但是语音识别器的错误会传递到机器翻译模型，造成了整体性能的下降。如果能获得完美的语音识别结果，在机器翻译方法中可以将BLEU值提升19.54，这19.54就是后续工作的最大提升空间。TCEN除利用了蒙古语语音-文本、蒙古语-汉语文本外，还利用了蒙古语语音-汉语文本数据进行微调，消除了两个模型的连接冲突，相较于级联方法提升了1.95。

本文提出的方法（TCEN-RNNT）在TCEN的基础上又提升了1.37，相比于级联方法和直接端到端训练分别提升了3.32和11.24。我们认为这种提升来源于两个方面，一方面是我们使用了RNN-T替换CTC，并使用蒙古文文本数据预训练其中的预测器，提升了语音识别模块的性能，另一方面是我们将语音识别作为辅助任务，多任务学习策略提升了最终的性能。以下通过两个实验证明这一观点。

首先，证明使用蒙古文文本数据预训练语音识别器的预测器，可以提升语音识别模块的性能。在语音识别模块中，设计两步预训练，即语音编码器的预训练和预测器预训练，我们分别去除这两个预训练步骤，语音识别器的性能列于表3，我们报告词错误率（WER）作为性能指标，词错误率越低越好。

模型	**WER(%)**
级联ASR	8.29
TCEN-ASR	11.58
TCEN-RNNT-ASR	**9.13**
–编码器预训练	12.82
–预测器预训练	11.94

表 3. 不同预训练策略对基于RNN-T语音识别模型影响的对比实验结果

表3中的减号（–）表示“没有”，表中显示没有编码器预训练、仅做预测器预训练，WER为12.82；没有预测器预训练、仅做编码器预训练，WER为11.94。说明无论是编码器预训练还是预测器预训练都可以带来语音识别模块的性能提升，通过编码器和预测器的预训练，整个语音识别模块的性能得到了提升（9.13），已经接近于目前最优的语音识别器的性能（8.29），并优于TCEN的语音识别模块的性能（11.58）。

相比之下，去除语音编码器预训练带来的性能下降更为明显，我们认为这来源于两个方面，首先在语音识别中，声学模型的重要性要比语音模型更大，比如语音识别器从传统的高斯混合模型（GMM）转变为深度的神经网络（DNN）就取得了巨大的提升，我们使用的基于CTC训练的语音识别器本身就可以达到较高的性能，对语言模型依赖较小。其次，我们使用了复杂的语音编码器预训练技术，除了使用蒙古语语音-文本数据做预训练之外，还使用了英语语音-文本做迁移学习，这也提升了语音识别器的性能。

与TCEN的语音编码器相比，若仅添加一个预测器将CTC改为RNN-T而不对新增的预测器做预训练并不能带来性能的提升。说明扩展模型的规模，如果没有相应的训练数据和训练策略的话，大的模型是一个负担，而不是优势。我们提出的模型TCEN-RNNT并不是因为参数规模扩大了，而是由于提出的多层次、多步骤的迁移学习和预训练策略是有效的。

我们进一步证明多任务学习的有效性。表4给出了在提出的方法中是否包含特定的训练步骤对最终结果的影响。

方法	**BLEU**
TCEN-RNNT + ASR预训练	16.35
TCEN-RNNT + MT预训练	16.98
TCEN-RNNT + ASR预训练+ MT预训练	18.77
TCEN-RNNT + 多任务	17.06
TCEN-RNNT + ASR预训练+ MT预训练+ 多任务	**19.64**

表 4. 针对TCEN-RNNT模型使用不同策略的对比实验结果

从表4中可以看出，多任务学习在提出的方法中起到了积极作用，使用了多任务学习的模型（19.64）比没有使用多任务学习的模型（18.77）的BLEU值提升了0.87。

表4中还分解出了语音识别模块预训练、机器翻译模块预训练和多任务学习各自起到的贡献。其中多任务学习起到的作用最为明显，强调了像提出的多层的神经网络模型设定中间目标的重要性，增加中间目标会明显降低学习难度。表4中没有列出既没有多任务学习也没有模块预训练的模型结果，是因为失去有效约束的TCEN-RNNT模型在现有规模的数据集下根本无法完成训练。无论是多任务学习还是模块预训练都可以改善这一情况。机器翻译模块预训练起到的作用稍逊于多任务学习，主要是因为这是一个语音翻译任务，最终的评判标准是翻译结果，机器翻译模块的预训练对约束翻译结果起到的贡献更大。作用最小的是语音识别模块的预训练，但也起到了很重要的作用。

同时做语音识别模块和机器翻译模块的预训练后，再应用多任务学习，并没有取得预期的特别大幅度的提升，这是因为，语音识别模块和机器翻译模块的预训练实际上也起到了对中间语音识别结果的约束作用，但这种约束可能会在后续微调中被逐渐放松，而增加多任务学习可以保证这种约束能一直维持下去。

总的来说，无论是多层次、多步骤的迁移学习和预训练策略，还是多任务学习策略，只要能解决数据限制问题，通过充分挖掘现有存量数据，将尽可能多的有效知识转移到端到端语音翻译模型当中，就能够让我们的端到端蒙汉语音翻译模型的性能得以提升。

5 结论

本文研究了端到端蒙汉语音翻译的方法。基于串联连接编码网络（TCEN）做出两点改进，一是使用RNN-T替换CTC改善了CTC语音识别模型主要侧重于声学特征提取，而不能捕获上下文语言知识或理解语义的问题；二是利用了多任务学习，设置语音识别为中间辅助目标，增加了模型约束，提出了TCEN-RNNT模型。为了能够对整个模型进行有效的训练，我们提出了多层次、多步骤的迁移学习和预训练策略，充分挖掘了现有存量数据的价值，打破了端到端语音翻译对训练数据的限制，把尽可能多的有效知识从各式各样的训练数据转移到端到端语音翻译模型中。并且在蒙汉语音翻译中实现了由传统级联方法到端到端语音翻译的跨越，把端到端方法引入了对蒙汉语音翻译的研究中。

致谢

本文的研究获得了国家重点研发计划项目（2018YFE0122900），国家自然基金项目（61773224，62066033），内蒙古自然科学基金项目（2018MS06006），内蒙古自治区成果转化项目（CGZH2018125），内蒙古自治区应用技术研究与开发资金项目（2019GG372，2020GG0046）的支持。

参考文献

Sameer Bansal, Herman Kamper, Karen Livescu, Adam Lopez, and Sharon Goldwater. 2019. Pre-training on high-resource speech recognition improves low-resource speech-to-text translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 58–68.

Alexandre Berard, L. Besacier, A. Kocabiyikoglu, and Olivier Pietquin. 2018. End-to-end automatic speech translation of audiobooks. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6224–6228.

Alexandre Bérard, Olivier Pietquin, Laurent Besacier, and Christophe Servan. 2016. Listen and translate: A proof of concept for end-to-end speech-to-text translation. In *NIPS Workshop on end-to-end learning for speech and audio processing*.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.

A. Graves. 2012. Sequence transduction with recurrent neural networks. *Computer Science*, 58(3):235–242.

Yuchen Liu, Hao Xiong, Jiajun Zhang, Zhongjun He, Hua Wu, Haifeng Wang, and Chengqing Zong. 2019. End-to-end speech translation with knowledge distillation. In *Interspeech 2019*, pages 1128–1132.

Rui Liu, Berrak Sisman, Feilong Bao, Guanglai Gao, and Haizhou Li. 2020. Modeling prosodic phrasing with multi-task learning in tacotron-based tts. *IEEE Signal Processing Letters*, 27:1470–1474.

Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. Multi-task sequence to sequence learning. In *ICLR 2016 : International Conference on Learning Representations 2016*.

Tomas Mikolov, Kai Chen, Greg S. Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *ICLR (Workshop Poster)*.

L. Muda, M. Begam, and I. Elamvazuthi. 2010. Voice recognition algorithms using mel frequency cepstral coefficient (mfcc) and dynamic time warping (dtw) techniques. *Ttps*, 2(3):138–143.

Vassil Panayotov, Guoguo Chen, D. Povey, and S. Khudanpur. 2015. Librispeech: An asr corpus based on public domain audio books. *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1715–1725.

Shuo Sun, Hongxu Hou, Nier Wu, and Ziyue Guo. 2020. Neural machine translation based on prioritized experience replay. In *International Conference on Artificial Neural Networks*, pages 358–368.

Yonghe Wang, Feilong Bao, Hongwei Zhang, and Guanglai Gao. 2017. Research on mongolian speech recognition based on fsmn. *National CCF Conference on Natural Language Processing and Chinese Computing*, pages 243–254.

Chengyi Wang, Yu Wu, Shujie Liu, Zhenglu Yang, and Ming Zhou. 2020. Bridging the gap between pretraining and fine-tuning for end-to-end speech translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9161–9168.

Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, Adithya Renduchintala, and Tsubasa Ochiai. 2018. Espnet: End-to-end speech processing toolkit. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pages 2207–2211.

Ron J. Weiss, Jan Chorowski, Navdeep Jaitly, Yonghui Wu, and Zhifeng Chen. 2017. Sequence-to-sequence models can directly translate foreign speech. In *Interspeech 2017*, pages 2625–2629.