深度递归的层次化机器翻译模型

刘宇鹏[1),2)] 马春光[1)] 张亚楠[2)]

[1)](哈尔滨工程大学计算机科学与技术学院 哈尔滨 150001)

[2)](哈尔滨理工大学软件学院 哈尔滨 150001)

摘 要 深度学习在自然语言处理中有很多的应用.深度网络的主要作用是捕获隐藏在语言结构中更深的语义信息.该文出发点为根据原有句子中的对齐作为深度网络生成结构的指导,并融合原有深度翻译模型的优点,提出了深度递归的层次化机器翻译模型.相对于已有的神经翻译模型来说,更好地结合了层次化的翻译过程,同时这种方法结合循环神经网络和递归神经网络的优点.层次化规则的归纳包含两个部分:短语的归纳和形式化规则的归纳,而在该文的建模过程中模拟了这两个部分且符合归纳过程.该文在训练中采用单词级语义错误、单语短语/规则语义错误和双语短语/规则语义错误构造目标函数,训练中能够更好平衡语义中3个部分的影响,同时考虑到对齐信息以指导层次化深度神经网络的训练.在解码过程中通过生成部分翻译结果的语义向量,最终得到句子间的语义关系,这样可以在语法结构中加入语义信息,克服了原有层次化模型语义信息缺乏的问题.该模型的实验结果说明了深度递归的层次化机器翻译模型的有效性,相对于经典的基线系统提高了1.49~1.84 BLEU分数.

关键词 循环神经网络;递归神经网络;词/短语/规则嵌入;层次化递归神经网络;自然语言处理

中图法分类号 TP18 **DOI号** 10.11897/SP.J.1016.2017.00861

Hierarchical Machine Translation Model Based on Deep Recursive Neural Network

LIU Yu-Peng[1),2)] MA Chun-Guang[1)] ZHANG Ya-Nan[2)]

[1)](*College of Computer Science and Technology, Harbin Engineering University, Harbin* 150001)

[2)](*College of Software, Harbin University of Science and Technology, Harbin* 150001)

Abstract Deep Learning has many applications in natural language processing. The main role is to capture the deeper semantic information hidden in the language structure through the deep network. The motivation of this paper is that we use the word alignment of the bilingual sentence as the guide to generate the structure of deep network, and combine these advantages of the original deep translation model. The paper proposes Hierarchical Recursive Neural Network (HRNN) for hierarchical machine translation model. Compared with the existing neural translation model, the model is a better combination of phrase-based hierarchical translation model and deep neural network. It has two advantages of Recurrent Neural Network (RTNN) and Recursive Neural Network (RENN). The procedure of phrase and formal rule induction can be simulated in HRNN, and the model meets induction procedure. In training procedure, the objection function of this paper include the monolingual word-level semantic errors, the monolingual phrase/rule semantic errors and the bilingual phrase/rule semantic error, and the semantic effect of three parts are balanced in statistical machine translation (SMT). In decoding procedure, the semantic relation among sentences are obtained by the semantic vector of partial translation result, and this

收稿日期:2015-09-13;在线出版日期:2016-04-05.本课题得到国家自然科学基金(61300115)、中国博士后科学基金(2014M561331)、黑龙江省教育厅科技研究项目(12521073)资助.**刘宇鹏**,男,1978年生,博士,副教授,主要研究方向为自然语言处理、机器翻译.E-mail:flyeagle99@gmail.com.**马春光**,男,1974年生,博士,教授,主要研究领域为自然语言处理、机器翻译.**张亚楠**,男,1980年生,博士,讲师,主要研究方向为推荐系统.

method, which the semantic information is added to the syntax structure, overcomes the lack of semantic information in the original model. The experimental results show that HRNN significantly improves the performance of a state-of-the-art SMT baseline system, leading to a gain of 1.49～1.84 BLEU points.

Keywords recurrent neural network; recursive neural network; word/phrase/rule embedding; hierarchical recursive neural network; natural language processing

1 引 言

深度神经网络(Deep Neural Network,DNN)作为原型学习[1]的一种重要方法,符合人类认识过程.从机器学习角度来说,深度神经网络是一种极其强大的方法,近年发展较快.它本质上是多层神经网络,在自然语言很多领域都有应用,如问答系统、句法分析、依存分析、SMT 等.DNN 也被引入到 SMT 组件的学习中,包括词对齐、语言模型、翻译模型和扭曲模型等.把深度神经网络应用到自然语言处理中,实际上是生成词嵌入或表示(Word Embedding or Representation).词嵌入是低维、密集和实数值的向量,这些词向量反映了一定的语义.DNN 的优点在于把手工提取的特征变成自动的提取过程,克服了原有人工特征的不完备性,且可以在提取后通过套用一个分类器或回归器来完成分类或回归任务;同时这些特征位于同一连续空间中,便于特征之间关系的度量.

然而传统 DNN 在建模的过程中并没有考虑语言的结构特性.为了解决这样的问题,出现了 RTNN[2]和 RENN,RTNN 网络符合语言生成的顺序结构性,RENN[3]符合语言生成的层次结构性.这两种模型在机器翻译中有很多应用[4-8].虽然在这两个方面的 SMT 模型有一些进展,但没有真正考虑短语/规则归纳过程,导致这些 DNN 并不真正符合翻译的过程.

层次化的机器翻译模型[9]从提出到现在一直是 SMT 任务的主流模型,首先从双语语料中抽取出初始短语对,接着根据抽取出来的初始短语对抽取出层次化规则,这些规则带有一定的特征.虽然语法结构中含有一定的语义信息,但还是缺乏刻画整个句子翻译过程的语义信息,而本文既有语义获取又有翻译过程结构预测,先进行 RTNN 训练语言的顺序性以获得单语语义,再使用 RENN 训练建模语言的层次性即是翻译过程结构的预测,使用相似度函数获取双语语义.本文的出发点是从原有的层次化建模中相对少的语义表示出发,使用深度网络获得语义特征(Semantic Feature)表示,并使用深度网络的递归建模翻译模型中的层次化翻译过程.主要从 3 个方面提出深度递归的层次化机器翻译模型:

(1)使用自编码器(AutoEncoder)对于源/目标语言重构和源到目标语言的翻译过程进行建模,无监督(Unsupervised)的生成语义相关的短语/规则语义向量表示,有监督(Supervised)把源语言和目标语言短语/规则向量之间相似度作为训练目标.采用分层的自编码器更好地建模初始短语对和基于初始短语生成的规则语义特征.本模型进行了 3 个阶段的联合训练,更好地平衡每个阶段的重要性.

(2)在训练过程中,和原有层次化的翻译过程无缝地连接起来,更好地捕捉结构化的翻译过程.在解码的过程中,使用了单双语语义特征,以及单双语敏感性特征来提高翻译性能.

(3)使用对齐信息指导深度网络的训练,在语义向量的获取中可以考虑到这种隐藏变量(Latent Variable)信息,最终根据对齐信息计算语义特征.

本文首先在第 2 节介绍深度神经网络在 SMT 中的相关应用;在第 3 节介绍深度递归的层次化机器翻译模型,包括整个系统的框架、生成语义向量的实现过程和参数估计;在第 4 节介绍进行调参时的特征训练算法,把得到的深度递归的层次化翻译模型整合到原有的(Cocke-Younger-Kasumi algorithm)CKY 解码框架下;在第 5 节设计关于本模型的实验,并与经典的基线系统进行比较;最后一节总结全文并展望下一步研究方向.

2 相关研究

近年来深度神经网络技术在 SMT 领域有很多的应用,按照翻译模型的类型可分为:基于短语(Phrase-based)翻译模型和基于括号转录文法(Bracket Transduction Grammar)翻译模型.在基于短语的翻译模型方面:文献[10]基于双语分别获得语义向量表示,通过相似度函数定义损失函数进行联合优化双语语义

内嵌的参数；文献[11]使用源语言到目标语言和目标语言到源语言的自编码器进行训练，且在训练过程中采用了按照频率进行排序后的课程训练(Curriculum Training)．在括号转录文法翻译模型方面：文献[4]提出了循环和递归的神经机器翻译模型，采用独立的循环神经网络获得初始的双语短语对向量，把RTNN和RENN生成的词向量和翻译过程的特征建模在一起，作为深度神经网络的输入；文献[5]使用自编码器对于源短语和目标短语进行训练，同时使用神经网络构建的分类器进行双语扭曲模型的训练．

按照使用神经网络的不同翻译阶段可以分为：在训练阶段使用深度神经网络，以语义特征的形式指导解码；在解码阶段使用深度神经网络，以置信分数指导解码．在训练阶段：文献[6]使用双语受限自编码器生成的语义向量进行短语剪枝和解码；文献[7]使用源语言的语境信息进行语言模型建模，并把这种语言模型特征加入到解码中．在解码阶段：文献[12]提出了可加的神经网络，使用生成好的源语言和目标语言词向量输入到一层隐层的神经网络中获得翻译置信得分以指导翻译解码过程；文献[8]对于SMT的推导结构进行预测．

在SMT其他方面的应用中，文献[13]采用信息检索模型获得与句子相关的文档集合，使用文档集合对神经网络进行训练以获得该句子的主题信息；文献[14]使用双向的循环神经网络对翻译模型和对齐模型进行一体化建模；文献[15]使用循环神经网络对翻译模型和语言模型进行一体化建模．

本文采用在训练阶段使用神经网络，直接把特征用于解码，这会降低整个解码过程的复杂度，把复杂度分担到训练阶段．不同于以往工作采用的短语/括号转录文法翻译模型，本文是使用更强的翻译模型即层次化的翻译模型进行建模，且在建模的过程中充分地考虑到了源语言和目标语言的语言特性，可以更好地结合原有的翻译过程．同时这种模型有更强的泛化能力，也适合于经典的短语/括号转录文法翻译模型．

3　深度递归的层次化机器翻译模型

3.1　系统框架

本翻译模型采用组件化的思想来进行建模，这样做的好处是可以更好地度量每个组件在翻译过程中所起到的作用，且在训练过程中进行联合训练，以反映每个组件的重要性．由于本文中涉及到很多符号表示：使用加粗正体表示向量和矩阵；使用(F,E)表示训练语料中的句子；(f,e)表示从训练语料中抽取出来的短语/规则；为了区分规则部分的短语和词，短语采用带下标大写字母，词使用带小标的短语表示，如F_i表示第i个位置的短语，f_i表示第i个位置的词．

在图1中数据预处理(Data Preprocessing)包括单语预处理和双语预处理，抽取短语和规则．使用训练好的循环神经网络生成的词向量输入到HRNN中进行训练．神经网络训练部分的层数和句子生成的推导树的高度一样(用省略号代替中间的层数)，训练部分主要包含两个部分：基于短语和规则的自编码器，在3.2节中介绍．

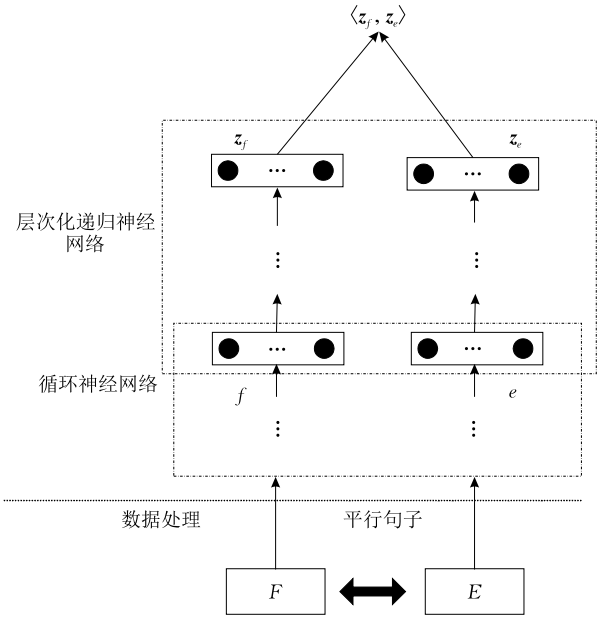

图1　深度神经网络训练总体框架

整个系统训练框架的描述为先使用预处理把每个句对(F_i,E_i)的短语/规则抽取出来，通过RTNN获得初始的单语词向量表示，接着把这些单语词向量表示(即两个神经网络交叉的部分)放入到层次化的递归深度网络中获得最终语义向量表示$\boldsymbol{z}_f,\boldsymbol{z}_e$，并采用内积$\langle \boldsymbol{z}_f,\boldsymbol{z}_e \rangle$对短语和规则的相似度进行度量．

本模型对训练语料中的双语句对进行建模，总体上分为两个部分：使用RTNN获得词向量，获得的词向量被放入到HRNN模型中；在HRNN模型中分为基于短语的自编码器和基于规则的自编码器，在每个自编码器中还包括单语自编码器和双语自编码器．在训练的过程中采用的方法为每个单语自编码器先进行逐层预训练，接着对双语自编码器

进行训练，最后进行联合训练，平衡每个部分的重要性.

3.2 语义向量

(1)基于短语的自编码器. 共使用 3 个深度递归的编码器，分别是源语言和目标语言自编码器，源语言作为输入和目标语言作为输出的自编码器.

图 2 下半部分刻画了短语之间的对齐关系，其中不同的投影/转换矩阵(Projection/Transformation Matrix)和语义向量在 3.3 节中说明，源/目标语言不同结构的递归编码器是根据短语中词对齐结果生成的，对于 f_1 的自编码部分恢复出来的语义向量记为 f_1'，其他类似. 对于对空的单词则与邻近的单词一起送入到神经网络中进行训练. 本来在双语推导树中并没有 y_2，y_4，z_2 和 z_4 节点，为了获得层次化的信息，将这些节点加入到推导树中，不会影响推导结果，只是增加了源/目标语言侧的推导树的层数. 不同的向量形式采用不同的灰度表示，在无监督训练部分不加入加灰度的层，而在有监督部分需要加上，因为这些层是用来建模翻译之间的层次化结构.

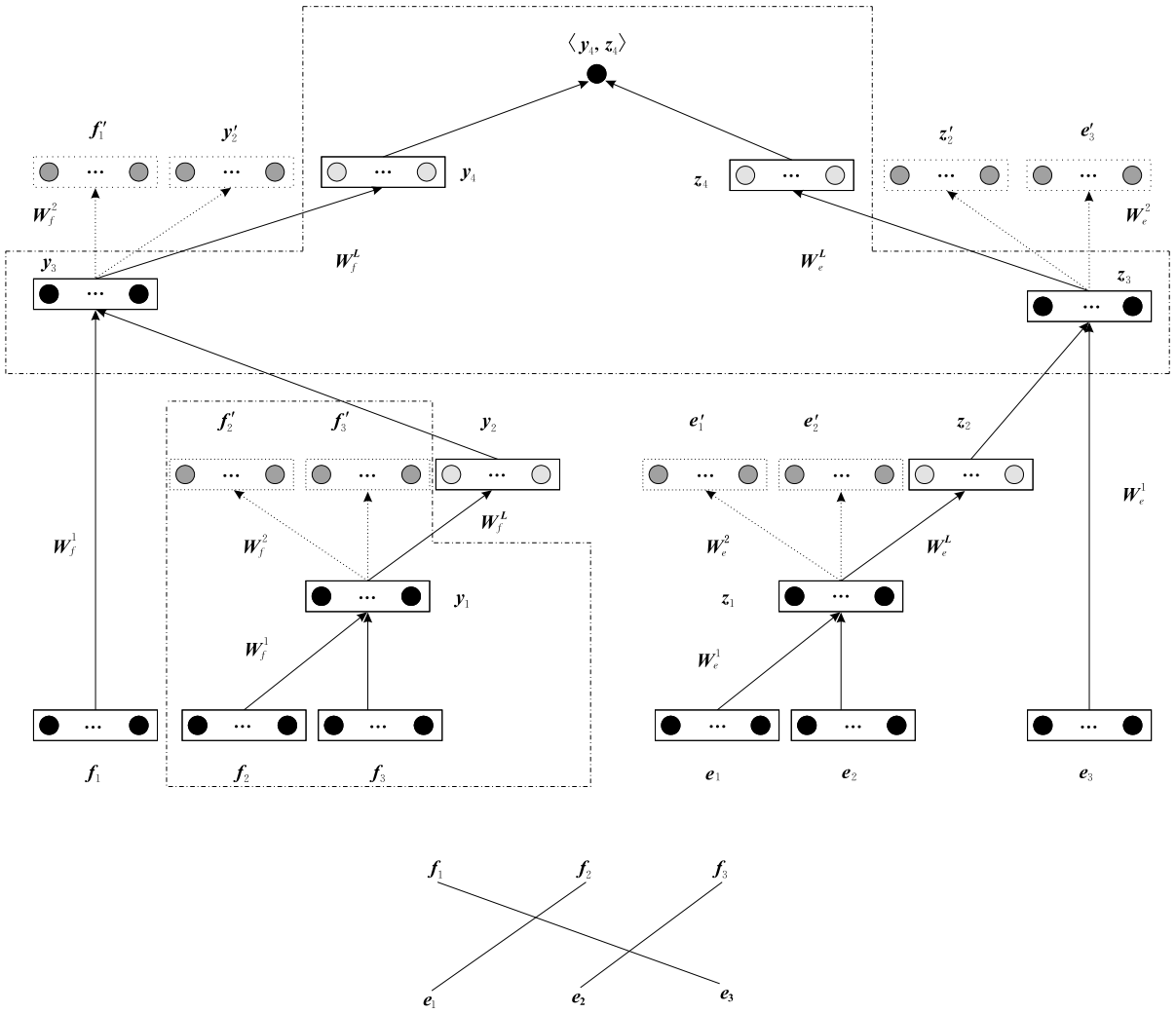

图 2 短语级自编码器和短语对齐关系

(2)基于规则的自编码器. 规则对是在双语句对中抽取短语对的基础上获得的，含有对齐信息的规则对送入到基于规则的自编码器计算其相似度，而相似度作为训练目标函数的一部分进行训练，从而对规则对中的参数矩阵进行了调节. 使用 RTNN 训练得到的词嵌入和基于短语的自编码器生成的短语嵌入作为输入，输出规则嵌入. 为了更好理解整个句子生成的自编码过程，使用图 3 和图 4 进行说明.

(1) $X_{5,7:0,2} \rightarrow F_3/E_1$

(2) $X_{7,9:2,5} \rightarrow F_4/E_2$

(3) $X_{3,9:0,8} \rightarrow f_2 X_{5,7:0,2} X_{7,9:2,5} / X_{5,7:0,2} X_{7,9:2,5} e_3$

(4) $X_{0,3:8,10} \rightarrow F_1/E_4$

(5) $X_{0,9:0,10} \rightarrow X_{0,3:8,10} X_{3,9:0,8} / X_{3,9:0,8} X_{0,3:8,10}$

(3′) $X_{5,9:0,5} \rightarrow X_{5,7:0,2} X_{7,9:2,5} / X_{5,7:0,2} X_{7,9:2,5}$

(3″) $X_{3,9:0,8} \rightarrow f_2 X_{5,9:0,5} / X_{5,9:0,5} e_3$

图 3 抽取出来的原规则和二叉化后的规则

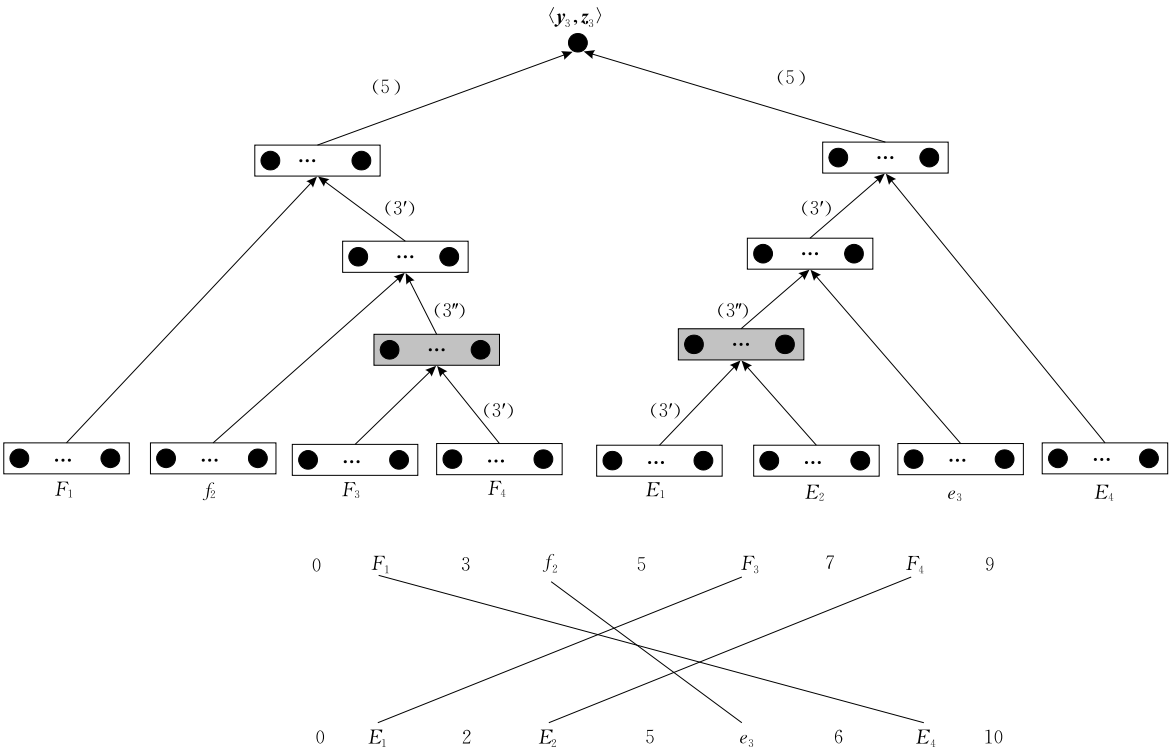

图 4　规则级自编码器和对齐关系

图 3 中共有 5 条层次化的规则和两条二叉化后的规则. $X_{3,9;0,8}$ 中 3,9 和 0,8 分别表示源语言的跨度(Span)是从 3 到 9 和目标语言的跨度是从 0 到 8. 为了得到统一的结构以便对短语和规则的自编码器进行训练,同时也提高解码精度,需对规则和短语进行同步二叉化[16](Synchronous Binarization). 这里只有规则(3)的源端和目标端有 3 个组成部分,需要进行二叉化,得到的两条规则(3′)和(3″),其中 $X_{5,9;0,5}$ 是生成的虚节点(Virtual Node).

根据规则(5)所得到的自编码器的结构如图 4,虚节点语义向量背景使用斜杠表示,把所使用层次化的规则分别标在层次化图的中间位置,最终会生成源和目标语言侧的规则向量 $\boldsymbol{X}_f^{0,9;0,10}$ 和 $\boldsymbol{X}_e^{0,9;0,10}$. 在图 4 下半部分中每个翻译单元左右标的是词/短语的跨度,下标是词/短语在短语中的位置,图 3 也是根据这个对齐关系抽取出来的规则. 把训练好的短语级语义向量(使用大写字母表示)输入到规则中,和左/右临接的短语/词向量一起输入到深度神经网络中进行训练以获得规则级语义向量. 由于空间问题没有把有/无监督自编码部分画上,同短语级自编码形式一样,为了获得规则级别的自编码和层次性只是采用的投影矩阵不一样.

3.3　训练算法

由于本模型分为基于短语和基于规则的自编码器,而且每个部分的层数也较多,为了减少时间复杂度采用分开训练,这样可以更快地达到训练目标. 先对基于词/短语/规则的自编码器进行分开训练,然后再进行联合训练. 基于短语和基于规则的训练公式基本差不多,只是在短语层和规则层选用不同投影矩阵进行线性变换,下面以基于短语为例进行说明.

(1)无监督自编码. 对于分层的无监督预训练阶段,首先需要获得词向量,词向量存在于一个由词向量所组成的矩阵 $\boldsymbol{L}\in\mathbb{R}^{n\times|V|}$ 中,矩阵 $\boldsymbol{L}$ 的形式如下:

$$\boldsymbol{L}=[\boldsymbol{f}_1,\cdots,\boldsymbol{f}_{|V|}] \tag{1}$$

其中:n 是词向量的维数,$|V|$ 表示字典的维数,$\boldsymbol{L}$ 是通过 RTNN 已经获得的词向量矩阵,$\boldsymbol{L}$ 中每个列向量的下标为单词在字典 V 中的编号. 通过下面公式取出第 i 个单词的词嵌入:

$$\boldsymbol{f}_i=\boldsymbol{L}\boldsymbol{e}_i\in\mathbb{R}^n \tag{2}$$

其中:$\boldsymbol{e}_i$ 是单位列向量,除了第 i 个位置为 1 外其他位置都为 0.

以图 2 左下面虚框部分为例建立模型. 除了根据对齐生成的结构不一致以外,源语言侧和目标语言侧自编码器基本一致,所以本部分只以源语言侧的自编码器为例. 编码层(Encoder Layer)和解码层(Decoder Layer)采用不同的矩阵进行线性变换,神经网络的激活函数采用“sigmoid”函数,编码层的函

数采用的公式如下：

$$\boldsymbol{y}_1=\text{sigmoid}(\boldsymbol{W}_f^1[\boldsymbol{f}_2;\boldsymbol{f}_3]) \tag{3}$$

解码层采用的公式如下：

$$[\boldsymbol{f}_2';\boldsymbol{f}_3']=\text{sigmoid}(\boldsymbol{W}_f^2\boldsymbol{y}_1) \tag{4}$$

其中：$\boldsymbol{f}_2$ 和 $\boldsymbol{f}_3$ 为词向量，$\boldsymbol{y}_1$ 是生成部分短语向量，$[\boldsymbol{f}_2;\boldsymbol{f}_3]\in\mathbb{R}^{2n\times 1}$ 表示把 n 维向量 $\boldsymbol{f}_2$ 和 $\boldsymbol{f}_3$ 连接起来形成一个 $2n$ 维向量，$\boldsymbol{W}_f^1\in\mathbb{R}^{n\times 2n}$ 和 $\boldsymbol{W}_f^2\in\mathbb{R}^{2n\times n}$ 是源语言侧投影矩阵. 虽然图 2 中解码后的向量用不同符号进行表示，但应该和原来输入的向量很接近. 为了便于比较向量的距离，每个向量需要进行归一化.

本文使用向量的 2-范数对自编码的重构错误(Reconstruction Error)进行建模. 区分以下两种情况，对于词的重构错误采用式(5)，对于部分短语/规则的重构错误采用式(6). 区分的原因由于长度不同的部分短语和词对于短语重构影响不一样，所以式(6)在每个重构部分需加上根据长度得到的权重，公式如下：

$$E_{\text{rec}}([\boldsymbol{f}_2;\boldsymbol{f}_3]|\boldsymbol{W}_f^1,\boldsymbol{W}_f^2)=\frac{1}{2}\|[\boldsymbol{f}_2';\boldsymbol{f}_3']-[\boldsymbol{f}_2;\boldsymbol{f}_3]\|_2^2 \tag{5}$$

$$E_{\text{rec}}([\boldsymbol{f}_1;\boldsymbol{y}_2]|\boldsymbol{W}_1,\boldsymbol{W}_2)=\frac{|f_1|}{|f_1|+|y_2|}\|[\boldsymbol{f}_1'-\boldsymbol{f}_1\|_2^2+\frac{|y_2|}{|f_1|+|y_2|}\|\boldsymbol{y}_2'-\boldsymbol{y}_2\|_2^2 \tag{6}$$

式(5)中为了求导方便在范数前面加了一个系数，式(6)中 $|f_1|$，$|y_2|$ 分别表示词 $\boldsymbol{f}_1$ 和部分短语 y_2 的长度.

根据源短语 p_f 的重构错误 $E_{\text{rec}}(p_f|\boldsymbol{W}_f^1,\boldsymbol{W}_f^2)$，可以得到所有训练源短语 C_f 的重构错误 $E_{\text{rec}}(C_f|\boldsymbol{W}_f^1,\boldsymbol{W}_f^2)$，见下面公式：

$$E_{\text{rec}}(p_f|\boldsymbol{W}_f^1,\boldsymbol{W}_f^2)=\sum_{n\in T(p_f)}E_{\text{rec}}([\boldsymbol{n.c}_1,\boldsymbol{n.c}_2]|\boldsymbol{W}_f^1,\boldsymbol{W}_f^2) \tag{7}$$

$$E_{\text{rec}}(C_f|\boldsymbol{W}_f^1,\boldsymbol{W}_f^2)=\frac{1}{|C|}\sum_{p_f\in C}E_{\text{rec}}(p_f|\boldsymbol{W}_f^1,\boldsymbol{W}_f^2)+\frac{\lambda_{\boldsymbol{W}_f^1}}{2}\|\boldsymbol{W}_f^1\|_2^2+\frac{\lambda_{\boldsymbol{W}_f^2}}{2}\|\boldsymbol{W}_f^2\|_2^2 \tag{8}$$

其中：p_f 表示源短语，$T(p_f)$ 表示源短语对应的生成树的中间节点集合，$\boldsymbol{n.c}_1$ 和 $\boldsymbol{n.c}_2$ 表示中间节点的左孩子和右孩子的语义向量，$\lambda_{\boldsymbol{W}_f^1}$ 和 $\lambda_{\boldsymbol{W}_f^2}$ 权重是通过手动调节设置的.

根据源短语的重构错误 $E_{\text{rec}}(C_f|\boldsymbol{W}_f^1,\boldsymbol{W}_f^2)$，目标短语的重构错误 $E_{\text{rec}}(C_e|\boldsymbol{W}_e^1,\boldsymbol{W}_e^2)$，源规则的重构错误 $E_{\text{rec}}(R_f|\boldsymbol{W}_f^{1'},\boldsymbol{W}_f^{2'})$ 和目标规则的重构错误 $E_{\text{rec}}(R_e|\boldsymbol{W}_e^{1'},\boldsymbol{W}_e^{2'})$，可以定义总的重构错误 $J_{\text{rec}}(\Theta_{\text{rec}})$ 见下面公式：

$$J_{\text{rec}}(\Theta_{\text{rec}})=E_{\text{rec}}(C_f|\boldsymbol{W}_f^1,\boldsymbol{W}_f^2)+E_{\text{rec}}(C_e|\boldsymbol{W}_e^1,\boldsymbol{W}_e^2)+E_{\text{rec}}(R_f|\boldsymbol{W}_f^{1'},\boldsymbol{W}_f^{2'})+E_{\text{rec}}(R_e|\boldsymbol{W}_e^{1'},\boldsymbol{W}_e^{2'}) \tag{9}$$

这里为了区分短语和规则以及源侧和目标侧均采用不同的编码和解码投影矩阵，Θ_{rec} 包含了源/目标语言短语/规则的投影矩阵 $\boldsymbol{W}_f^1,\boldsymbol{W}_f^2,\boldsymbol{W}_e^1,\boldsymbol{W}_e^2,\boldsymbol{W}_f^{1'},\boldsymbol{W}_f^{2'},\boldsymbol{W}_e^{1'},\boldsymbol{W}_e^{2'}$.

(2)有监督自编码. 除了需要考虑到原有无监督自编码器得到的短语/规则语义向量外，还需考虑源语言和目标语言之间的自编码器对于短语/规则语义向量的修正. 和无监督训练中采用的转换矩阵不一样，源/目标语言端短语采用的转换矩阵分别为 $\boldsymbol{W}_f^L$ 和 $\boldsymbol{W}_e^L$，它们是用来获得源语言/目标语言短语的层次化结构，而源/目标语言规则采用的转换矩阵分别为 $\boldsymbol{W}_F^L$ 和 $\boldsymbol{W}_E^L$. 这里也是以双语短语的自编码器为例，规则部分的自编码器类似，如图 2 中上面虚线的部分，转换矩阵的转换函数公式如下：

$$\boldsymbol{y}_4=\text{sigmoid}(\boldsymbol{W}_f^L\boldsymbol{y}_3) \tag{10}$$

$$\boldsymbol{z}_4=\text{sigmoid}(\boldsymbol{W}_e^L\boldsymbol{z}_3) \tag{11}$$

需先定义含有对齐的双语短语/规则 (f,e,a) 相似度(也是语义特征)，公式如下：

$$h_{\text{bi-phr}}(f,e)=sim(f,e)=\langle\boldsymbol{y}_4,\boldsymbol{z}_4\rangle \tag{12}$$

其中：双语短语 (f,e) 对应的源语言侧和目标语言侧的语义向量分别为 $\boldsymbol{y}_4$ 和 $\boldsymbol{z}_4$. 由于是有监督的训练，需要定义损失函数，常用的损失函数为：有/无结构间隔的排序损失函数[2,4,6,8,13](Ranking Loss with/without Structure Margin)和交叉熵[5,17](Cross Entropy). 本文所采用的由最小错误率训练[18](Minimum Error Rate Training)思想得到的负的基于 N-best 列表的期望 $BLEU$(N-best List based Expected BLEU)作为训练目标函数，公式如下：

$$J_{\text{exp}}(\Theta_{\text{exp}})=-\sum_{E\in Gen(F_i)}P(E|F_i)sBLEU(E_i,E)=-\frac{\sum_{E\in Gen(F_i)}\exp(\boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{h}(F_i,E))sBLEU(E_i,E)}{\sum_{E\in Gen(F_i)}\exp(\boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{h}(F_i,E))} \tag{13}$$

其中：$sBLEU(E_i,E)$ 是生成候选翻译句子 E 和参考翻译句子 E_i 之间的句子级别的 $BLEU$，$Gen(F_i)$ 表示源句子 F_i 产生的所有可能的候选翻译结果，其中使用 $softmax$ 计算翻译概率 $P(E|F_i)$. 本部分需要对训练语料中每个句对 (F_i,E_i) 进行解码生成 N-best 翻译结果，因为无法衡量双语相似度特征对解码器性能的影响，这里采用先使用解码器进行解码生成翻译结果，训练的目标函数使得句子级的 $sBLEU$(度量翻译结果句子 E 与参考翻译句子 E_i 的

相似程度)期望最大，使得生成的翻译结果句子 E 与参考翻译句子 E_i 接近.

由于无监督目标函数的训练只是反应了单语自编码器的性能，需要和双语的有监督训练一起进行，还得加上训练词向量的 RTNN 部分，所以总错误 J_{total} 的公式如下：

$$J_{\text{total}}=\alpha J_{\text{rec}}(\Theta_{\text{rec}})+\beta J_{\text{rtnn}}(\Theta_{\text{rtnn}})+(1-\alpha-\beta)J_{\text{exp}}(\Theta_{\text{exp}}) \tag{14}$$

其中：α 和 $\beta(0\leq\alpha,\beta\leq1)$ 是权重，刻画每个子模型的重要程度，3 个错误中包含深度网络的训练参数. 通过 RTNN 中目标函数 $J_{\text{rtnn}}(\Theta_{\text{rtnn}})$ 可以获取压缩后的词向量，其中句子 f_i 和 e_i 每个词采用一处激活(one-shot)向量表示，Θ_{rtnn} 中包含 3 个投影矩阵，具体矩阵见文献[3]；通过 $J_{\text{rec}}(\Theta_{\text{rec}})$ 可以生成源语言/目标语言短语/规则的语义向量，在 Θ_{rec} 中包含 8 个投影矩阵，分别为源语言侧短语投影矩阵 $\boldsymbol{W}_f^1$ 和 $\boldsymbol{W}_f^2$，目标语言侧短语投影矩阵 $\boldsymbol{W}_e^1$ 和 $\boldsymbol{W}_e^2$，源语言侧规则投影矩阵 $\boldsymbol{W}_f^{1'}$ 和 $\boldsymbol{W}_f^{2'}$，目标语言侧规则投影矩阵 $\boldsymbol{W}_e^{1'}$ 和 $\boldsymbol{W}_e^{2'}$；通过 $J_{\text{exp}}(\Theta_{\text{exp}})$ 可以矫正单语短语/规则语义向量成为双语短语/规则语义向量表示，Θ_{exp} 中包含 4 个投影矩阵，分别是源语言侧短语投影矩阵 $\boldsymbol{W}_f^{\boldsymbol{L}}$，源语言侧规则投影矩阵 $\boldsymbol{W}_F^{\boldsymbol{L}}$，目标语言侧短语投影矩阵 $\boldsymbol{W}_e^{\boldsymbol{L}}$，目标语言侧规则投影矩阵 $\boldsymbol{W}_E^{\boldsymbol{L}}$.

(3)参数估计. 为了降低训练的时间复杂度并克服梯度弥散现象，采用 3 个训练阶段：无监督的词向量训练阶段、无监督的单语短语/规则的训练阶段和有监督的双语短语/规则的训练阶段，前一个阶段作为后一个阶段的输入. 在监督的词向量训练阶段中，对于目标函数 $J_{\text{rtnn}}(\Theta_{\text{rtnn}})$ 中的参数 Θ_{rtnn} 的估计采用文献[3]中的方法，Θ_{rtnn} 中特征为全局参数. 在无监督的单语短语/规则的训练中先采用无监督分层(Layer-wise)预训练(Pre-training)，再采用有监督的精调(Fine-tuning)训练. 训练方法采用一种按误差逆传播(Back Propagation)算法训练的多层前馈网络，需计算对于不同参数的梯度.

对于目标函数 $J_{\text{rec}}(\Theta_{\text{rec}})$ 中的参数 Θ_{rec} 的估计. $J_{\text{rec}}(\Theta_{\text{rec}})$ 包含 4 个重构错误部分，分别对每个部分的参数求导. 由于每个部分和参数的求导公式基本相同，所以下面只给出源语言侧短语重构错误对于参数 $\boldsymbol{W}_f^1$ 的偏导数，公式如下：

$$\frac{\partial E_{\text{rec}}(C_f|\boldsymbol{W}_f^1,\boldsymbol{W}_f^2)}{\partial \boldsymbol{W}_f^1}=\frac{1}{|C|}\sum_{p_f\in C}\frac{\partial E_{\text{rec}}(p_f|\boldsymbol{W}_f^1,\boldsymbol{W}_f^2)}{\partial \boldsymbol{W}_f^1}+\lambda_{\boldsymbol{W}_f^1}\boldsymbol{W}_f^1 \tag{15}$$

对于目标函数 $J_{\text{exp}}(\Theta_{\text{exp}})$ 中的参数 Θ_{exp} 的估计. 主要是在句对 (F_i,E) 中短语/规则对 (f,e) 的语义相似度特征 $h_{\text{bi-phr}}(f,e)$ 求导，以图 2 上面虚线的部分为例，对于 $\boldsymbol{W}_f^{\boldsymbol{L}}$ 求导公式如下，对于 $\boldsymbol{W}_e^{\boldsymbol{L}}$ 求导公式基本与之相同：

$$\begin{aligned}\frac{\partial h_{\text{bi-phr}}(f,e)}{\partial \boldsymbol{W}_f^{\boldsymbol{L}}}&=\frac{(\partial \boldsymbol{y}_4)^{\mathrm{T}}}{\partial \boldsymbol{W}_f^{\boldsymbol{L}}}\boldsymbol{z}_4+(\boldsymbol{y}_4)^{\mathrm{T}}\frac{\partial \boldsymbol{z}_4}{\partial \boldsymbol{W}_f^{\boldsymbol{L}}}\\&=\boldsymbol{y}_3(\boldsymbol{z}_4\circ\text{sigmoid}'(\boldsymbol{y}_3'))^{\mathrm{T}}+\boldsymbol{z}_3(\boldsymbol{y}_4\circ\text{sigmoid}'(\boldsymbol{z}_3'))^{\mathrm{T}}\end{aligned} \tag{16}$$

其中："$\circ$"是向量间元素相乘的 *hamard* 乘积，$\text{sigmoid}'(\boldsymbol{y}_3')$ 是对向量 $\boldsymbol{y}_3'=\boldsymbol{W}_f^{\boldsymbol{L}}\boldsymbol{y}_3$ 中每个元素求导，$\text{sigmoid}'(\boldsymbol{z}_3')$ 是对向量 $\boldsymbol{z}_3'=\boldsymbol{W}_f^{\boldsymbol{L}}\boldsymbol{z}_3$ 中每个元素求导.

4　解码整合和特征权重的训练

解码整合是把语义特征整合到解码器中. 由于获得的语义向量是按照不同对齐生成的，在解码中使用的短语/规则特征数值是经过统计并计算得到的数值，而对于训练获得的这些短语/规则向量 $\boldsymbol{z}_f$ 和 $\boldsymbol{z}_e$ 也需要进行统计. 本文采用文献[9]的方法，设置每个短语对的共现次数 c 为 1，每个规则对的共现次数 c 为分数，公式如下：

$$\boldsymbol{z}_f=\frac{\sum_{(e,f)\in(f,e,a)}c\times\boldsymbol{z}_f}{\sum_{(e,f)\in(f,e,a)}c} \tag{17}$$

其中：(f,e,a) 表示含有对齐结果的短语/规则对. 对含有对齐关系的短语对 (f,e,a) 中的每个短语 (f,e) 的源语言语义向量 $\boldsymbol{z}_f$ 进行求和，获得新的不带对齐关系的源语言语义向量 $\boldsymbol{z}_f$，$\boldsymbol{z}_e$ 计算公式相似. 把源/目标短语/规则语义向量之间的相似度作为双语短语/规则 (f,e) 的语义特征 $h_{\text{bi-phr}}(f,e)$，这里采用内积进行计算，公式如下：

$$h_{\text{bi-phr}}(f,e)=\langle\boldsymbol{z}_f,\boldsymbol{z}_e\rangle \tag{18}$$

除了双语语义特征外，还有源语言和目标语言的两个单语语义特征 $h_{f\text{-phr}}(f',f)$ 和 $h_{e\text{-phr}}(e',e)$，其形式相同. 由于单语重构后得到的复现结果和原来的单语语义向量有一定的差异，单语语义相似度特征反映了单语自编码器的性能.

为了区分语义向量的每一维以反应不同语义信息[13]，本文使用标准熵定义源语言侧短语/规则语义敏感度特征 $h_{f\text{-sen}}(f,e)$，公式如下：

$$h_{f\text{-sen}}(f,e)=-\sum_{i=1}^{|\boldsymbol{z}_f|}z_{f_i}\times\log z_{f_i} \tag{19}$$

目标语言侧短语/规则语义敏感度特征 $h_{e\text{-sen}}(f,e)$ 与之形式相同. 标准熵越大，反应了语义向量每一维

有趋向于相同的语义信息(即语义向量的分布趋向于平坦分布),其敏感度越低.对于同样的语义相似度,解码器更趋向于选择不敏感的短语/规则.双语语义敏感度特征 $h_{\text{bi-sen}}(f,e)$ 通过连接源语言语义向量 $\boldsymbol{z}_f$ 和目标语言语义向量 $\boldsymbol{z}_e$ 形成新的既包含源语言又包含目标语言的语义向量,再通过与式(19)相似的形式进行计算;反应了在新的向量空间中源语言和目标语言语义向量之间的重要性.

特征向量 $\boldsymbol{h}(f,e)$ 中包含三大类特征,具体如下:

(1)传统的标准特征.两个方向的翻译概率,两个方向的词汇权重,5元语言模型,词数,短语数,层次化短语的个数,空惩罚;

(2)语义特征.双语语义相似度(Bilingual Semantic Similarity),两个单语语义相似度(Monolingual Semantic Similarity),双语语义敏感度(Bilingual Semantic Sensitivity),两个单语语义敏感度(Monolingual Semantic Sensitivity);

(3)稀疏特征.高频(频率出现大于300 k)特征编号.

解码方法采用经典的对数线性模型(Log-Linear Model),其解码公式如下:

$$\begin{aligned} N(F_i) &= \underset{N(F_i)\subseteq GEN(F_i)}{\arg\max_N} \boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{h}(F_i,E) \\ &= \underset{N(F_i)\subseteq GEN(F_i)}{\arg\max_N} \underbrace{\sum_j \lambda_j h_j(F_i,E)}_{standard} + \\ &\quad \underbrace{\sum_k \lambda_k h_k(F_i,E)}_{semantic} + \underbrace{\sum_l \lambda_l h_l(F_i,E)}_{sparse} \end{aligned} \tag{20}$$

在公式中 $N(F_i)$ 代表每个源语言句子 F_i 生成的 N-best 结果,$\arg\max_N$ 是用来求解码空间中 N-best 的候选翻译结果,$\boldsymbol{\lambda}$ 是权重向量,$\boldsymbol{h}(F_i,E)$ 是源句子 F_i 和生成的候选翻译句子 E 之间的特征向量,句子级语义特征是通过规则/短语级的语义特征(见式(18))计算获得的.精确的解码是无法实现的,本文采用保留 N-best 的 CYK 进行近似解码,且在解码过程后采用森林技术[19]进行重排序(Re-ranking),这样可以在更大的解码空间下进行解码以生成更加精确的结果.

训练采用轮换交替和自举(Bootstrapping)的思想,基本过程如下:

(1)给定预先设定的参数向量 $\boldsymbol{\lambda}$,其中语义特征 $\lambda_{\text{bi-phr}},\lambda_{f\text{-phr}},\lambda_{e\text{-phr}},\lambda_{\text{bi-sen}},\lambda_{e\text{-sen}},\lambda_{f\text{-sen}}$ 设置为0,翻译系统根据当前设置生成 N-best 翻译候选;

(2)固定参数向量 $\boldsymbol{\lambda}$,语义特征 $\lambda_{\text{bi-phr}},\lambda_{f\text{-phr}},\lambda_{e\text{-phr}},\lambda_{\text{bi-sen}},\lambda_{e\text{-sen}},\lambda_{f\text{-sen}}$ 设置为1.采用拟牛顿法(L-BFGS)对深度网络中有/无监督部分中参数 $\Theta_{\text{rec}},\Theta_{\text{exp}}$ 和 Θ_{rtnn} 进行估计.这些固定结构神经网络的推导(Derivation)可以通过反向传播算法实现,其梯度式见(15)和(16);

(3)固定深度神经网络的参数 $\Theta_{\text{rec}},\Theta_{\text{exp}}$ 和 Θ_{rtnn},在开发集上进行特征权重的学习.由于本文中采用了大规模稀疏特征进行训练,所以使用成对排序优化[20](Pairwise Ranking Optimization)进行训练.

5 实 验

5.1 超参设置

层次化神经网络超参的设置包含词向量维数 n,分别设置维数为50,100,200.L-BFGS 中的学习率 η,按照最好的实验结果设置为0.01.对于式(14)中的超参 α 和 β 以步长0.05从范围0.1到0.5进行变化,其中 $\alpha=0.2,\beta=0.15$,说明有监督自编码器起到最重要的作用,其次是无监督自编码器,作用最小的是 RTNN 模型;变化式(8)以及其他重构错误中8个正则化超参(变化的范围从 10^{-1} 到 10^{-5},步长为 10^{-1}),层次化神经网络指导搜索过程,获得这些正则化超参分别为 $\lambda_{\boldsymbol{W}_f^1}=10^{-2},\lambda_{\boldsymbol{W}_f^2}=10^{-2},\lambda_{\boldsymbol{W}_e^1}=10^{-1},\lambda_{\boldsymbol{W}_e^2}=10^{-1},\lambda_{\boldsymbol{W}_f^{1'}}=10^{-2},\lambda_{\boldsymbol{W}_f^{2'}}=10^{-2},\lambda_{\boldsymbol{W}_e^{1'}}=10^{-1},\lambda_{\boldsymbol{W}_e^{2'}}=10^{-1}$,我们发现在源语言侧的正则化超参的数值小于目标侧,说明源语言的正则化项起的作用相对较小,也就是说源语言无监督自编码器中起作用的参数相对较多,即源语言侧的语义向量在相似度特征的贡献上起到了更加重要的作用.

除此以外,初始词向量采用循环神经网络进行训练生成.我们使用开源的工具包 rnnlm[3] 在大规模数据集上进行训练,包括5.2节中的 SMT 双语训练语料,还有第3版 Gigaword 语料的中英文部分(中文侧含有30 M 句子和1 G 的词,英文侧含有7 M 句子和238 M 的词).

5.2 SMT 设置

本实验在 NIST 中进行英文翻译任务.所采用的训练语料为 LDC2000T50,LDC2002L27,LDC2002E18,LDC2003E4,LDC2005T10,LDC2005T83,LDC2006E85,LDC2007T09.双语语料中含有0.99 M 句对,31.3 M 中文词和32.4 M 英文词,使用 Gigaword 语料和双语训练语料的目标语言训练语言模型,使用特征衰减算法(Feature Decay Algorithm)[21]过滤双语语料.开发集采用的是 NIST02,测试集采用的是 NIST05,NIST06,NIST08.对于双语语料的预处理

包括，采用最大熵模型进行中文分词，使用 tokenizer.perl(http://www.statmt.org)进行英文分词. 本文使用 Berkeley LM[22] 工具训练 5 元语言模型，对齐采用 GIZA++进行双语对齐，然后使用 Grow-Diag-Final-And 启发式规则获得多对多的词对齐. 为了评价生成翻译结果的质量，采用大小写不敏感的 5 元 BLEU[23] 进行评价，并采用重采样(Re-sampling)方法[24]对翻译结果进行统计显著性检验(Statistical Significance Test). 短语抽取采用基于 Hadoop 技术的开源系统 Thrax[25]，解码器采用层次化翻译模型开源系统 Joshua 5.0[26] 的解码部分.

为了进一步减少深度神经网络的训练复杂度，本文先使用解码器在训练双语句对上进行解码，只把使用到的短语/规则对放入到深度神经网络中进行训练，剩下短语对 1.37 M，剩下规则对 2.13 M，并使用这些短语/规则对对原有含有对齐信息的短语/规则对进行过滤.

5.3　实验结果与分析

(1) 语义特征的影响. 为了比较语义向量对翻译性能的影响和层次化神经网络的有效性，本部分采用两个基线系统：不加任何语义特征的 baseline1 和不采用层次化建模的 baseline2. baseline2 的实现过程基本与 HRNN 相同，只是在训练过程中不考虑对齐以指导源语言和目标语言侧的短语/规则的层次化，采用源语言和目标语言侧自左向右的方法建立短语/规则语义向量，同时 baseline2 中也有双语语义相似性特征.

为了验证特征的有效性，我们首先加入双语特征：双语语义相似性特征(bssm)，双语语义敏感性特征(bssn)；再加入单语特征：单语语义相似性特征(mssm)，单语语义敏感性特征(mssn). 使用粗体表示在检验指标($p<0.05$)下显著的优于基线系统.

表 1 中给出了相应的实验结果，ALL 表示把所有的测试语料放在一起，我们发现 baseline2 的性能基本和 baseline1 一样，基本没有获得语义信息以增强翻译结果. HRNN 的结果分别在 3 个测试集和 ALL 上比基线 baseline2 分别提高了 1.31，0.9，0.85 和 1.56 BLEU 分数，说明了对齐信息在建模过程中的重要性. 我们也发现主要是双语语义特征和双语敏感度特征在翻译中起到了作用，而双语语义特征相对重要一些，单语语义特征和单语敏感度特征对于翻译性能几乎没有影响，因此也说明加入双语语义特征和双语敏感度特征的 HRNN 模型是与层次化短语翻译互补的，体现了语义特征的重要性.

表 1　不同语义特征的影响

方法	NIST05	NIST06	NIST08	ALL
baseline1	35.98	33.88	30.17	35.12
baseline2	36.02	34.22	30.17	35.27
HRNN(bssm)	**36.82**	**35.83**	30.80	**36.22**
HRNN(bssm+bssn)	**37.22**	**35.09**	31.02	**36.82**
HRNN(bssm+bssn+mssm+mssn)	**37.33**	**35.12**	**31.02**	**36.83**

为了探索语义向量维数对翻译性能的影响，设计了语义向量维数取不同值 $n=50,100,200$ 的实验. $n=50$ 是表 1 中 HRNN(bssm+bssn+mssm+mssn)中的结果，已在表 1 中显示这里不再列出. 在表 2 中可以看到，不管 n 取什么值，加入了双语和单语语义特征的 HRNN 模型的结果性能都优于基线系统 baseline1 和 baseline2，同时在 $n=100$ 达到了最好性能，在 3 个测试集合上分别又提高了 0.22，0.54，0.64 和 0.28 *BLEU* 分数，也就是说 $n=100$ 能够更好地区分好的和坏的翻译结果.

表 2　语义向量维数对于翻译性能的影响

方法	n	NIST05	NIST06	NIST08	ALL
HRNN	100	**37.55**	**35.64**	**31.66**	**37.11**
	200	**37.52**	35.52	**31.64**	**37.08**

(2) 单/双语自编码器对于短语重构的影响. 为了比较单语短语嵌入(Mono-lingual Phrase Embedding)和双语短语嵌入(Bi-lingual Phrase Embedding)，在下面的实验分别简写为 MPE 和 BPE. 单语短语嵌入的基本方法就是把计算双语语义向量的部分，单独进行无监督训练，对于生成的单语短语嵌入使用内积计算相似度. 表 3 中给出对一些造成性能差异的英文短语的分析，把单语相似度(即特征 $h_{e\text{-phr}}(e',e)$ 的数值)排名前三位列了出来，发现 MPE 生成的短语更加贴近于表面意思，而 BPE 生成的短语有更深层的意思，说明了 BRE 通过源语言更好地捕获了短语含义.

表 3　双语短语嵌入和单语短语嵌入异同

Phrase	MPE	BPE
work for the government	1. do for the office 2. work in the government 3. do and work in the office	1. work for the official 2. do something for the government 3. do things in the government
an office worker	1. an official worker 2. an office employer 3. a government worker	1. staff in the office 2. employer working for office 3. office staff
the same time	1. the same thing 2. in the meantime 3. same meantime	1. simultaneously 2. concurrently 3. meantime

(3)规则二叉化的影响. 我们在抽取规则的时候限制非终结符号 2～4 个,对于所有不同的非终结符号个数都归一化为 2 个,采用同一个投影矩阵. 因为不是所有规则都可以二叉化,我们把其中 3.24% 不可以二叉化的规则移除掉. 使用 removed 表示移除掉不可以二叉化的规则,使用 binarization 表示进行二叉化,进行二叉化是表 1 中 HRNN(bssm+bssn+mssm+mssn)中的结果. 表 4 说明了具体的性能,发现进行二叉化对于翻译性能几乎没有影响.

表 4 二叉化对于翻译性能的影响

方法	NIST05	NIST06	NIST08	ALL
移除掉	37.32	**35.12**	**31.08**	**36.82**
二叉化	37.33	**35.12**	**31.02**	**36.83**

6 结论与展望

本文探索了深度递归的层次化翻译模型. 本模型更符合翻译过程,以对齐为指导生成源语言和目标语言带有结构信息的神经网络. 本模型中不仅考虑到含有全局信息的词向量,而且在训练神经网络的过程中考虑到了双语对齐信息. 在训练中采用了 3 个训练模块,对于不同模块使用不同的目标函数,更好地平衡每个模块的影响,且在无监督部分使用了分层的预训练,使得每层更快地达到训练目标函数. 训练数据使用了典型的双语训练语料,并进行了过滤,测试数据也采用了多组测试数据以反映方法的有效性. 通过实验发现我们的方法超过了经典的基线系统大约 1.84 个 BLEU 分数,且做了显著性测试以证明方法的统计意义.

从实验结果中可以看出,双语短语的嵌入对于 SMT 起到关键的作用. 我们希望在将来从 3 个方面对该模型进行扩展:(1)把该模型应用到更多的 SMT 模型中,进一步证明其有效性;(2)对于篇章级 SMT 使用本方法进行建模;(3)使用复述语料对单语的短语嵌入模型进行训练,把单语无监督自编码扩展成为既有无监督又有有监督,以提高单语自编码部分的性能.

致 谢 本文感谢开源解码系统 Joshua、基于 Hadoop 平台实现的规则抽取系统 Thrax 的相关人员!

参 考 文 献

[1] Onishi K H, Murphy G L, Bock K. Prototypicality in sentence production. Cognitive Psychology, 2010, 56(2): 103-141

[2] Scocher R, Bauer J, Manning C D, Ng A Y. Parsing with compositional vector grammars//Proceedings of the ACL 2013. Potsdam, Germany, 2013: 455-465

[3] Mikolov T, Karafiát M, Burget L, et al. Recurrent neural network based language model//Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH 2010). Makuhari, Chiba, Japan, 2010: 1045-1048

[4] Liu Shujie, Yang Nan, Li Mu, Zhou Ming. A recursive recurrent neural network for statistical machine translation//Proceedings of the ACL 2014. Baltimore, Maryland, USA, 2014: 1491-1500

[5] Li Peng, Liu Yang, Sun Maosong. Recursive autoencoder for ITG-based translation//Proceedings of the ACL 2013. Potsdam, Germany, 2013: 567-577

[6] Zhang Jiajun, Liu Shujie, Li Mu, Zong Chengqing. Mind the gap: Machine translation by minimizing the semantic gap in embedding space//Proceedings of the AAAI 2014. Québec City, Canada, 2014: 1657-1663

[7] Devlin J, Zbib R, Huang Z, et al. Fast and robust neural network joint models for statistical machine translation//Proceedings of the ACL 2014. Baltimore, Maryland, USA, 2014: 1370-1380

[8] Zhai Feifei, Zhang Jiajun, Zhou Yu, Zong Chengqing. RNN-based derivation structure prediction for SMT//Proceedings of the ACL 2014. Baltimore, Maryland, USA, 2014: 779-784

[9] Chiang D. Hierarchical phrase-based translation. Computational Linguistics, 2007, 33(2): 201-228

[10] Gao Jianfeng, He Xiaodong, Yih Wne-tau, Deng Li. Learning semantic representation for the phrase translation model. Microsoft Research, Redmond: Technical Report MSRTR2013-88, 2013

[11] Zou W Y, Socher R, Cer D, Manning C D. Bilingual word embeddings for phrase-based machine translation//Proceedings of the Empirical Methods in Natural Language Processing 2013. Seattle, Washington, USA, 2013: 120-128

[12] Liu L, Watanabe T, Sumita E, Zhao T. Additive neural networks for statistical machine translation//Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistic. Potsdam, Germany, 2013: 761-768

[13] Cui Lei, Zhang Dongdong, Liu Shujie, et al. Learning topic representation for SMT with neural networks//Proceedings of the ACL 2014. Baltimore, Maryland, USA, 2014: 133-143

[14] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to alignand translate//Proceedings of the ICLR 2015. San Diego, USA, 2015: 1-15

[15] Auli M, Galley M, Quirk C, Zweig G. Joint language and translation modeling with recurrent neural networks//Proceedings of the EMNLP 2013. Seattle, Washington, USA, 2013: 1044-1054

[16] Zhang Hao, Huang Liang, Gildea D, Knight K. Synchronous binarization for machine translation//Proceedings of the 2006 Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-06). New York, USA, 2006: 256-263

[17] Socher R, Pennington J, Huang E H, et al. Semi-supervised recursive autoencoders for predicting sentiment distributions// Proceedings of the Empirical Methods in Natural Language 2011. Edinburgh, UK, 2011: 1011-1022

[18] Och F J. Minimum error rate training in statistical machine translation//Proceedings of the ACL2003. Morristown, USA, 2003: 160-167

[19] Mi Haitao, Huang Liang, Liu Qun. Forest-based translation //Proceedings of the ACL 2008. Columbus, USA, 2008: 192-199

[20] Hopkins M, May J. Tuning as ranking. Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing. Edinburgh, UK, 2011: 1352-1362

[21] Bicici E, Yuret D. Instance selection for machine translation using feature decay algorithm//Proceedings of the 6th Workshop on Statistical Machine Translation of Association for Computational Linguistics. Portland, Oregon, USA, 2011: 272-283

[22] Pauls A, Klein D. Faster and smaller N-gram language models//Proceedings of the ACL 2011. Portland, Oregon, USA, 2011: 258-267

[23] Papineni K, Roukos S, Ward T, Zhu Wei-Jing. BLEU: A method for automatic evaluation of machine translation// Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. Philadelphia, USA, 2002: 311-318

[24] Koehn P. Statistical significance tests for machine translation evaluation//Proceedings of the EMNLP 2004. Barcelona, Spain, 2004: 388-395

[25] Weese J, Ganitkevitch J, Callison-Burch C, et al. Joshua 3.0: Syntax-based machine translation with the thrax grammar extractor//Proceedings of the WMT11. Edinburgh, UK, 2011: 478-484

[26] Post M, Ganitkevitch J, Orland L, et al. Joshua 5.0: Sparser, better, faster, server//Proceedings of the WMT13. Sofia, Bulgaria, 2013: 206-212

LIU Yu-Peng, born in 1978, Ph. D., associate professor. His research interests focus on natural language processing, machine translation.

MA Chun-Guang, born in 1974, Ph. D., professor, Ph. D. supervisor. His research interests focus on natural language processing, machine translation.

ZHANG Ya-Nan, born in 1980, Ph. D., lecturer. His research interests focus on recommend system.

Background

Deep learning is a new area of machine learning research, which has been introduced with the objective of moving machine learning closer to one of its original goals: artificial intelligence. Recently, there are a lot of research papers in the domain. Deep Learning is about learning multiple levels of representation and abstraction that help to make sense of data such as images, sound, and text. For natural language processing, deep learning solve the insufficient learning problem, which less layers of original neural network learning cause. Semantic information is important to machine translation task. Many effort is devoted to this direction. Current machine translation system often capures the semantic feature of bilingual corpus during the entire translation procedure including training and decoding. The main research significance of this paper is that the alignment is introduced into the neural model, and we construct the target function to better balance semantics in deep network training procedure. The target function includes three parts, which are the word-level semantic errors, monolingual phrase/rule semantic errors and bilingual phrase/rule semantic errors. In order to measure the semantic information in the machine translation system, the decoder join not only the bilingual semantic features, but also the two monolingual semantic features of the source and target language.

The work is supported by the National Natural Science Foundation of China (61300115), the China Postdoctoral Science Foundation (2014M561331), and the Science and Technology Research Project of Education Department of Heilongjiang Province (12521073). The work mentioned by this paper is a part of this project. The author looks forward to give a better model, which describes the semantics in translation model. Of course, this research direction is important in many current machine translation researches.