文章编号：1003-0077（2011）00-0000-00

依存边转换翻译规则生成器*

陈宏申[1,2]，刘群[1,3]

（1. 中国科学院计算技术研究所，中国科学院智能信息处理重点实验室，北京市 100190; 2. 中国科学院大学，北京市 100190; 3. 爱尔兰都柏林城市大学，爱尔兰）

摘要：统计机器翻译模型，特别是基于句法的翻译模型，其翻译单元在保留足够的翻译信息以及翻译单元在翻译新句子时的泛化能力上始终存在着一个平衡。神经网络被成功用于统计机器翻译模型中的调序和语言生成中。本文提出了一个新颖的基于神经网络的句法翻译规则生成器——依存边转换翻译规则生成器（DETG），它利用一条转换翻译规则的源端以及源端的上下文作为输入，以依存边转换翻译规则的目标端作为输出。它不仅保留了依存边——这种最简单的句法翻译规则的灵活性，保证了翻译规则的泛化能力，同时通过上下文信息增强了转换翻译规则的匹配能力。生成器的结构非常简洁，它将翻译规则的源端作为输入，同时生成翻译规则目标端的对应翻译以及依存边的位置关系。我们使用生成器对解码时所用到的依存边转换翻译规则打分。我们在三个 NIST 测试集上的实验显示，相较于基线系统，平均有 1.39 个 BLEU 值的提升。

关键词：依存边；转换；翻译；神经网络；生成器

中图分类号：TP391　　**文献标识码：**A

A Dependency Edge Transfer Translation Rule Generator

Hongshen Chen[1,2], Qun Liu[1,3]

(1. Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China ; 2. University of Chinese Academy of Sciences, Beijing 100190 ,China; 3. Dublin City University, Dublin, Ireland)

Abstract: There always been a tradeoff between the amount of information a translation unit preserved and its ability to be generalized when translating new sentences in a statistical machine translation model, especially in syntax-based ones. Neural network has been successfully applied in reordering and language generation problems in SMT. In this paper, we propose a novel syntactic translation rule generator based on neural network, a dependency edge transfer rule generator (DETG), which leverage the source side of a transfer rule and local context as input and output the target side. It shares not only the benefit of dependency edge, the most relax syntactic constraint, as transfer unit, to ensure its generalization ability but also the local context as additional information to help it with a better matching ability. The structure of the generator is quite concise, with the source side of a translation rule as input, it generates the target side of a translation rule, namely the translation correspondence and position relations of the target edge simultaneously. We use the generator to score the transfer rules when decoding. We show experiments on three NIST test sets and it yields a significance performance with an averaged 1.39 BLEU higher over the baseline.

Key words: Dependency; Transfer; Translation; Neural Network; Generator

1 引言

近年来统计机器翻译见证了一系列的发展。从翻译单元的角度，可以分为基于词的翻译模型[2]，基于短语的翻译模型[22][15]和基于句法的翻译模型[33][12][5][21][13][20][8][26][28][31]。在统计机器翻

* **收稿日期：**　　**定稿日期：**

基金项目：自然科学基金（61379086）

译中，有两个重要因素，分别是规则抽取和解码的过程。前者关注如何定义翻译知识的形式，并从双语平行语料中抽取翻译知识。后者则利用翻译知识，进行翻译。翻译规则，作为翻译知识的表示形式，是核心组件之一，其在相当程度上决定了翻译模型的能力。

相较于其他的翻译模型，基于句法的模型在一些方面具有特别的吸引力。由于有句法信息作为指导，它能产生更符合句法结构的翻译，并且能更好地处理长距离依存以及调序关系。在基于句法的翻译模型当中，依存句法树结构被视为一种既包含了句法信息又包含了浅层语义的结构。研究人员基于依存句法树构建了很多翻译模型。

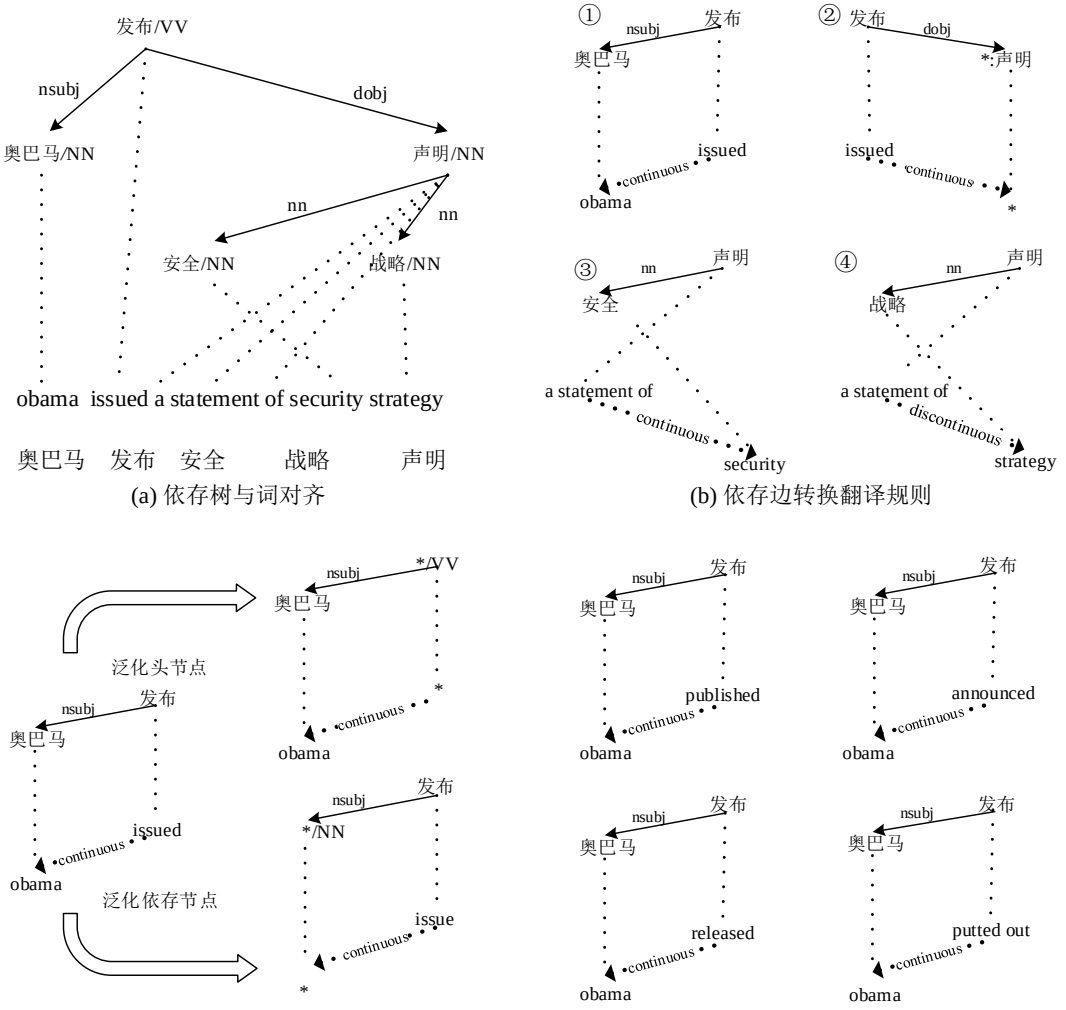
Lin [20]使用路径作为转换翻译结构。路径是依存树的片段，可以视为一个连续的或者非连续的短语。每条路径只有一个根节点。在翻译的时候，他通过将相同根节点的不同路径合并起来生成翻译。Quirk 等人[26]则使用树权(treelet)进行树到树的翻译，树权是一个联通的依存树的子树。Shen 等人[28]，则在层次短语翻译规则的目标端增加 fixed 和 floating 的依存树结构构建串到依存树的翻译模型，其中 fixed 结构是头节点-依存节点集合中，包含头节点的一颗依存子树，而 floating 结构是仅包含依存节点集合的连续短语子树。Xie 等人[31]使用头节点-依存节点集合作为基本的翻译单元，提出了基于依存树到串的翻译模型。研究人员探索了不同的方法用于解构依存树，采用同步文法（除了 Lin[20]）进行翻译。

Chen 等人[4] 14 年提出了一种基于依存边的非同步过程的翻译模型。他们将依存树最基本的单元—依存边—投射到目标端，然后通过组合目标端依存边来生成目标端的句子。他们提出的翻译单元相当简单，一条依存边在依存树中捕捉一个头节点和一个依存节点的修饰关系，在翻译的过程中具有相当的灵活性。在图 1（b）中，规则①的源端依存边，“奥巴马”是一个依存节点，它修饰头节点“发布”。相应地，目标端依存边，“obama”修饰“issue”。特别地，为了缓解翻译规则的数据稀疏问题，他们甚至泛化一条转换翻译规则中的头节点或者依存节点。在图 1（c）中，规则①中的依存节点“奥巴马”被泛化成了一个变量或者头节点“发布”被泛化成一个变量。在翻译的过程中，他们通过组合目标端依存边的方式来生成翻译句子，因而其调序更加灵活。

尽管 Chen 等人[4]所提出的依存边在翻译中有着较强的灵活性，然而一条依存边仅包含一个头节点和一个依存节点，因而翻译规则的所有上下文都被忽略了，其上下文匹配的能力则明显受限。特别地，这种上下文的匹配能力还由于头节点和依存节点的词被泛化而遭到进一步削弱。另一方面，依存边这种最基本的结构单元有时还会遭遇歧义过大的问题。图 1（d）中，由于“发布”在不同的上下文环境中存在不同的翻译，同一个源端依存边<发布，奥巴马>可以对应于多条目标端的依存边。而且，头节点和依存节点的泛化还会进一步恶化这个问题。在解码的过程中，过多的目标端依存边将极大地膨胀解码空间，导致合理的搜索变得较为困难。

然而，如果在规则表示上增加上下文信息，那么翻译规则的上下文匹配能力会因此得到加强，但却失去了翻译规则表示的灵活性，并且在双语平行语料较少时，低频的依存边上会更加稀疏。因此，有没有可能在增强翻译规则的上下文匹配能力同时又不牺牲它的知识表示的简洁性和应用上的灵活性呢？同时，由于一条依存边不仅描述了头节点的词和依存节点的词在源端和目标端翻译上的对应关系，还描述了它们之间的位置上的对应关系，如左右关系，相邻或者不相邻的连续关系，如何既增强模型翻译上的对应能力，同时增强位置关系上的区分能力呢？

近年来神经网络被成功地运用于统计翻译模型的调序[18][19]和语言生成[1][7][6][14][27]之中。本文提出了一种新颖的基于神经网络结构的依存边转换翻译规则生成器去解决这些问题。依存边转换翻译规则生成器采用一个转换翻译规则的源端依存边作为输入，以转换翻译规则的目标端依存边作为输出。为了增强其上下文匹配能力，我们同时将源端头节点和依存节点的上下文输入到生成器中。当解码时，我们使用它对转换翻译规则的候选目标端依存边进行打



(a) 依存树与词对齐　(b) 依存边转换翻译规则

(c) 依存边转换规则的泛化　(d) 依存边转换规则的歧义

图 1 依存边转换翻译规则

分并重排序，在三个 NIST 的测试集上，依存边转换规则生成器平均有 1.39 个 BLEU 的提升。

2 基于依存边转换的翻译

依存边转换翻译规则从一棵源端依存树及其目标端译文和它们的词对齐中抽取。图 1（b）是图 1（a）中抽取的依存边转换翻译规则。基于依存边转换的翻译使用依存边作为其翻译单元。一条边描述了头节点和依存节点之间的关系。一条转换翻译规则由两条边组成，一条源端的边和一条目标端的边。源端的边是一个四元组$\langle H_{src}, D_{src}, P_{src}, R\rangle$，$H_{src}$ 源端依存边的头节点，D_{src} 是源端依存边的依存节点，P_{src} 是源端依存边头节点与依存节点的左右相对位置关系标记，R 是头节点与依存节点的依存关系标记。在图 1（b）中，转换翻译规则①的源端依存边的头节点是“发布”，其依存节点是“奥巴马”。其依存关系标记“nsubj”表示一个名词作为主语。“奥巴马”出现在“发布”的左边。目标端依存边是另外一个四元组$\langle H_{tgt}, D_{tgt}, P_{tgt}, C\rangle$，$H_{tgt}$ 是目标端依存边的头节点，D_{tgt} 是目标端依存边的依存节点，P_{tgt} 是头节点与依存节点的左右相对位置关系标记，C 是头节点与依存节点相邻或者不相邻的连续位置关系标记。对应地，第一条转换翻译规则的头节点的目标端边的头节点是“issued”，依存节点是“obama”。“obama”出现在“issued”的左边，且二者连续。

依存边转换翻译规则还会通过泛化头节点或者依存节点来缓解数据稀疏的问题。图 1（c）中分别泛化了头节点和依存节点。其中，泛化头节点意味着“奥巴马”（obama）可

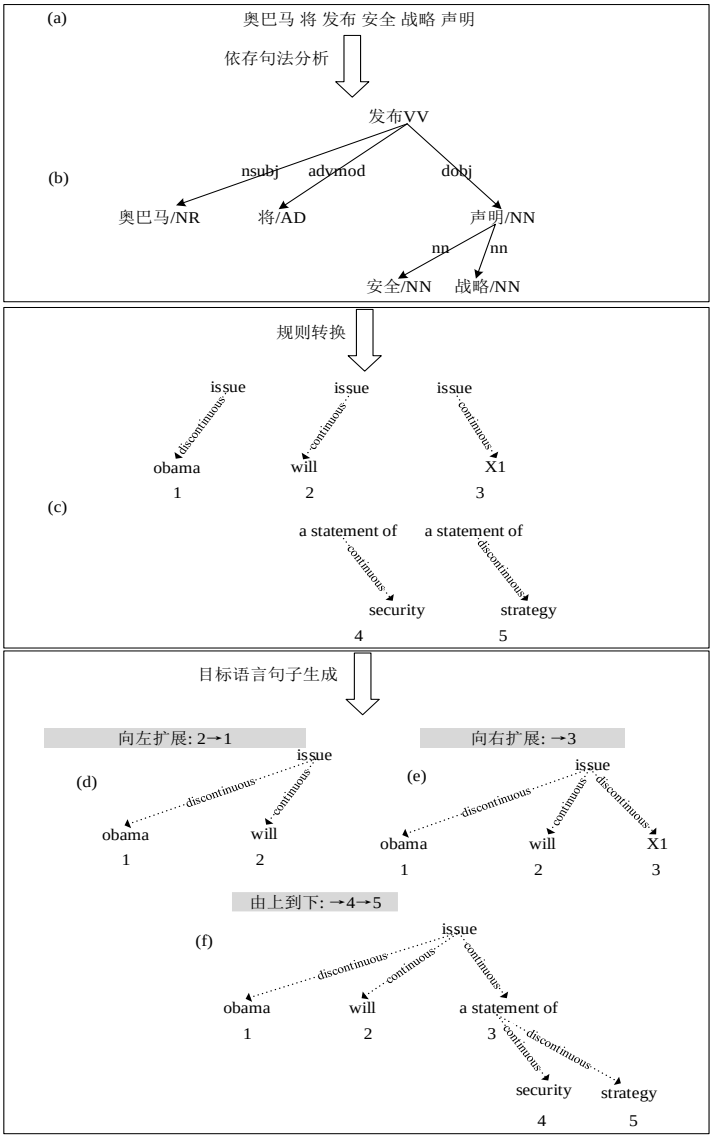


图 2 依存边转换翻译过程

以修饰任何动词，这些依存边共享了相同的结构。而泛化依存节点则表示任何一个名词都能修饰“发布”(issued)。

依存边转换翻译模型的翻译过程如图 2 包括三个部分。对于一条输入的句子，首先通过依存句法分析器得到其句法分析树。然后，利用依存边转换翻译规则，将源端依存边投射到目标端。最后，通过组合目标端依存边，生成目标端译文。在每一个内部节点使用 Beam-search 算法，枚举搜索目标端依存边的最优组合。

在基于依存边转换的翻译模型中，其转换规则相当灵活。因此，目标端的句子生成在调序上非常灵活。在目标端的边上，其仅有的位置约束包括头节点和依存节点的左右相对位置关系以及连续位置关系。然而，由于源端和目标端的对应关系是仅通过依存边建立起来的，如此一来，源端的上下文信息被忽略了。在实际的应用之中，一条源端的依存边可能对应到数十乃至数百个目标端的边。图 1（d）展示了与图 1（b）中规则①的源端依存边相同，但目标端依存边不同的四条依存边转换翻译规则。在规则抽取中，它们的头节点“发布”在不同的上下文环境下存在四个不同的翻译候选。但是，在解码的过程中，过多的（且经常是不合适的）候选目标端依存边，给解码的过程中选择上下文匹配的目标端的边大大增加了难度。因而导致在目标端句子的生成过程中面临了一个巨大的搜索空间。此外，如图 1（c）中的

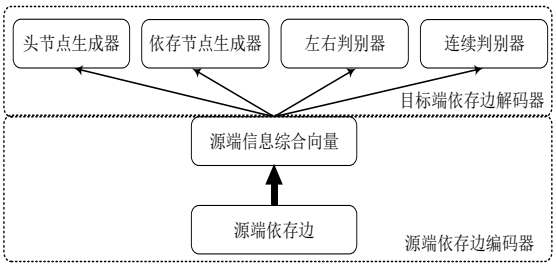


图 3 依存边转换规则生成器

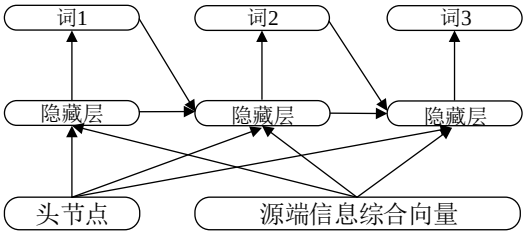


图 4 目标端头节点生成器

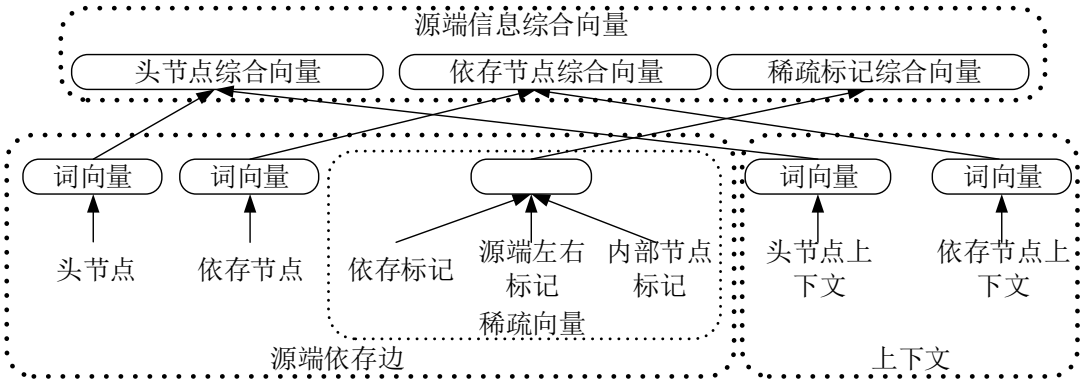


图 5 源端依存边编码器

泛化了头节点或者依存节点的翻译规则虽然缓解了数据稀疏的问题，但却是依存边转换翻译规则的歧义性的另一处来源。

3 依存边转换翻译规则生成器

本文提出了依存边转换翻译规则生成器（DETG），直接学习将源端的依存边匹配投射到目标端。它不仅能增强依存边转换规则的上下文匹配能力，而且由于它直接使用稠密特征生成目标端依存边，因而不存在因源端依存边在双语平行语料中不存在或出现的次数较少导致的数据稀疏问题，与此同时，它还保存了依存边转换翻译规则的灵活性，不增加翻译规则本身在信息存储上的负担。

依存边转换翻译规则生成器包含了一个编码器和一个解码器。编码器以一条转换翻译规则的源端的边的所有信息作为输入。

解码器则包含了四个部分，对应于目标端的边的四个组成元素（目标端的头节点，目标端的依存节点，左右相对位置标记，连续标记），包括目标端头节点生成器，目标端依存节点生成器，左右判别器和连续判别器。图 3 展示了依存边翻译规则生成器的主体结构。

给定一条源端的边 e_{src}（以及其上下文信息），我们希望得到目标端的边 e_{tgt}。由于目标端的边包含了四个元素，目标端头节点（$head_{tgt}$），目标端依存节点（dep_{tgt}），左右位置标记（lr_{tgt}）和连续标记（cd_{tgt}）。假设四个元素相互独立，只条件依赖于源端的依存边，那么依存边转换规则生成器可以被定义为：

$$p(e_{tgt} \mid e_{src}) = p(head_{tgt} \mid e_{src}) * p(dep_{tgt} \mid e_{src}) * p(lr_{tgt} \mid e_{src}) * p(cd_{tgt} \mid e_{src})$$

3．1 源端依存边编码器

源端依存边编码器使用前馈神经网络将源端的边编码成一个固定长度的源端信息综合向量（$\vec{s}$）。在一条依存边转换翻译规则中，一条源端的边仅包含四组信息。此外，我们还将源端头节点和依存节点的上下文作为输入，增强转换翻译规则的上下文匹配能力。我们通过一个位于源端头节点和依存节点的文本窗口，取固定数量的上下文的词作为额外的输入。在实

验中，我们将检视看上下文在多大程度上增强了转换翻译规则的上下文匹配能力。

由于编码器的源端的边的四元组包含不同信息类型。头节点和依存节点是词，而左右位置关系标记和依存关系标记则是标记信息。因而，我们区分这两种类型的信息作为输入。图5是编码器的结构示意图。

对于头节点$head_{src}$和依存节点dep_{src}，我们分别映射成头节点词向量$\vec{e}_{head}$和依存节点词向量$\vec{e}_{dep}$。假设头节点在源句子词序列（$w_1, w_2, \ldots, w_{n-1}, w_n$）是第$i$个词，如果上下文窗口大小为$c$，则取词序列$w_{i-c}, \ldots, w_{i-1}, w_{i+1}, \ldots, w_{i+c}$组成头节点的上下文$head_{con}$。然后将其映射成头节点上下文词向量$\vec{e}_{headcon}$。最后将头节点上下文词向量与头节点的词向量连接成$\vec{x}_1$。最后，经过非线性变换，得到头节点综合向量$\vec{h}_{\text{head}}$：

$$\vec{x}_1 = \vec{e}_{\text{head}} \oplus \vec{e}_{\text{headcon}}, \quad \vec{h}_{\text{head}} = g(w_1\vec{x}_1 + b_1),$$

其中g是sigmoid函数，$\oplus$是连接操作。

类似地，依存节点词向量$\vec{e}_{dep}$以及其上下文词向量$\vec{e}_{depcon}$也被连接成$\vec{x}_2$。经过非线性变换，得到依存节点综合向量$\vec{h}_{\text{dep}}$：

$$\vec{x}_2 = \vec{e}_{\text{dep}} \oplus \vec{e}_{\text{depcon}}, \quad \vec{h}_{\text{dep}} = g(w_2\vec{x}_2 + b_2)$$

而对于标记信息，则组合成0/1稀疏向量作为输入。

其中，对于依存标记***deplabel***，我们定义映射特征函数为：

$$f_{deplabel} = \begin{cases} 0, \text{如果} label = deplabel \\ 1, \text{其他} \end{cases}$$

对于左右相对位置关系(*lrtag*)，定义映射特征函数为：

$$f_{left/right} = \begin{cases} 0, \text{如果} lrtag = left \\ 1, \text{其他} \end{cases}$$

此外，我们还将依存节点是内部节点或者是叶节点也作为稀疏向量的组成部分，定义其映射函数为：

$$f_{inner/leaf} = \begin{cases} 0, \text{如果依存节点是叶节点} \\ 1, \text{其他} \end{cases}$$

稀疏向量$\vec{x}_{\text{sparse}}$被直接转化成稀疏综合向量$\vec{h}_{\text{sparse}}$：$\vec{h}_{\text{sparse}} = g(w_3\vec{x}_{\text{sparse}} + b_3)$。

然后，这三个向量被连接成源端综合信息向量$\vec{s}$：$\vec{s} = \vec{h}_{\text{head}} \oplus \vec{h}_{\text{dep}} \oplus \vec{h}_{\text{sparse}}$。

3．2 依存边转换规则解码器

给定源端信息综合向量$\vec{s}$，依存边转换翻译规则解码器产生目标端依存边。目标端的边的四个元素同时产生。与源端的边对应，目标端的边同样的四个元素也包含两种类型。目标端头节点（$head_{tgt}$），目标端依存节点（dep_{tgt}）为字符串，左右相对位置标记(lr_{tgt})和连续位置标记(cd_{tgt})可以视为二元分类的标记。

我们使用两种不同的网络产生目标端。对于目标端头节点和目标端依存节点，我们使用两个循环神经网络（RNN)。因为目标端头节点和依存节点的词的个数不确定，可能包含多个词。对于两个位置关系标记，我们采用两个前馈神经网络进行判定。

目标端头节点和依存节点的生成器结构一致。以目标端头节点的生成为例，图 4 中循环神经网络的隐藏状态可以计算为：

$$h_i = g(w_h h_{i-1} + w_s\vec{s} + w_{\text{head}}\vec{e}_{\text{head}} + w_y\vec{y}_{i-1} + b_h),$$

其中，h_{i-1}是上一个词对应的隐藏状态，$\vec{s}$是源端信息综合向量，$\vec{e}_{head}$是源端头节点词向量，$\vec{y}_{i-1}$是上一个产生的词的词向量。g是 sigmoid 函数。

根据h_i，经过 softmax 分类器即得到目标端头节点的第 i 个词的概率y_{head_i}。假设目标端头节点有 n 个词，那么，头节点的概率定义为：$y_{head} = \prod_{i}^{n} y_{head_i}$。

左右关系判定器和连续关系判别器的结构同样类似。给定源端信息综合向量$\vec{s}$，由于左右位置关系和连续关系都是二元分类，我们只需要知道两个分类结果中的其中一个的概率便可得知另外一个分类的概率，因而通过下面的式子进行预测：

$$f_{lr} = g(w_{lr}g(w_{lrs}\vec{s} + b_{lrs}) + b_{lrs}),$$

其中，g 是 sigmoid 函数。

3．3 依存边转换规则生成器的训练

依存边转换规则的生成器的训练损失函数是由四个子损失函数组合而成，这四个子损失函数，对应于依存边转换规则解码器的四个组成部分。我们定义损失函数如下：

$$J_\theta = J_{head} + J_{dep} + J_{lr} + J_{cd},$$

其中，J_{head}是目标端头节点生成器的损失函数，J_{dep}是目标端依存节点生成器的损失函数。J_{lr}和J_{cd}是左右位置关系和连续关系判别器的损失函数。

J_{head}和J_{dep}采用负的对数似然函数：

$$J_{head} = \sum_{n} \log y_{head}^{n*}$$

$$J_{dep} = \sum_{n} \log y_{dep}^{n*},$$

其中，y_{head}^{n*}和y_{dep}^{n*}分别是目标端头节点和依存节点的概率。

J_{lr}和J_{cd}则是间隔最大化函数，但是这里我们定义间隔为 0：

$$J_{lr} = \max(0, y_{lr}^{*} - y_{lr})$$

$$J_{cd} = \max(0, y_{cd}^{*} - y_{cd}),$$

其中，y_{lr}和y_{cd}是正确预测的概率，y_{lr}^{*}和y_{cd}^{*}是错误预测的概率。

间隔为 0 意味着一旦位置标记预测正确了，即便是正确的概率只比错误的概率高一点如（0.51 比 0.49），那么我们也认为预测正确，损失为 0。这么定义间隔距离是因为在训练样例中，训练实例的比例并不均衡。如左右关系位置判定，大部分训练实例源端和目标端的左右关系是对应一致的，源端和目标端左右出现调序的实例比例相对较少。同样地，大多数训练实例，目标端的头节点和目标端依存节点是不相邻的。我们认为，这些占比相对较少的训练实例所包含的信息量特别大，而采用间隔距离函数则能在一定程度上避免这种问题。特别地，我们定义间隔距离为 0，这是因为我们认为只要能够给出一个模糊的正确的判定就已经足够。

4 解码

解码过程中，我们使用依存边转换规则生成器来增强模型选择合适的目标端的边的能力。具体而言，为了使用该生成器，我们在依存边转换翻译模型中增加一个新的特征。

依存边转换使用对数线性[23]框架，寻找目标端的边的最佳组合。它包含 13 个特征：

- 依存边转换翻译规则的前向和后向翻译概率
- 依存边转换翻译规则的前向和后向词汇化翻译概率

系统	MT03	MT04	MT05	均值
Moses	32.3	33.43	31.44	32.39
DEBT	32.57	35.06	31.36	32.99
+DETG	**33.8***	**36.58***	**32.76***	34.38

表 1 BLEU-4 分数（%） 在 NIST MT03-05 测试集上。“Moses”采用默认设置，“DEBT”是基线系统。“+DETG”也即 DEBT 翻译模型增加了本工作。“*”表示结果显著好于基线系统($p<0.01$).

系统	MT03	MT04	MT05	均值
DEBT	32.57	35.06	31.36	32.99
+nocon	33.56	36.06	32.37	33.99
+con1	33.8	**36.58**	**32.76**	34.38
+con2	33.73	36.52	32.45	34.23
+con3	**33.94**	36.24	32.49	34.22

表 2 BLEU-4 分数（%）在 NIST MT03-05 测试机上，采用不同的上下文作为输入。“nocon”表示没有上下文作为输入。“con1,con2,con3”分别表示上下文窗口大小为 1,2,3

- 源端依存子树 fixed 短语片段的前向和后向翻译概率
- 源端依存子树 fixed 短语片段的前向和后向词汇化翻译概率
- 依存边转换翻译规则和源端依存子树 fixed 短语的规则数量惩罚
- 虚构的依存边转换翻译规则
- 目标端的词个数惩罚
- 语言模型

给定一条源端的依存边，以及其对应的候选目标端的边。我们将源端的边（及其上下文）输入到生成器中，然后获取几个所有候选目标端的边的损失值，将其作为特征加入到模型中。

我们采用最小错误率训练[25]来调整翻译模型的权重参数。

5 实验

我们在中文-英文的翻译上设计了一系列实验以验证下列问题：

1. 依存边转换规则生成器能够帮助模型更好地匹配源端和目标端的依存边吗？
2. 上下文信息在多大程度上能够增强模型的匹配能力？

5．1 步骤

我们的训练语料包含了 260K LDC（LDC2002E18, LDC2003E07, LDC2003E14, LDC2004T07, LDC2005T06）数据的平行句对。我们使用 NIST MT 评测的测试集 2002 作为开发集，2003-2005 NIST 测试集作为最终的测试集。

我们使用斯坦福句法分析器[17][3]来分析源端的句子，其输出为投射的源端依存树，依存树中的各个节点的词被标注了词性标记，并且依存树的各个边也被标记了依存标签。我们使用 GIZA++，在源端到目标端和目标端到源端两个方向上同时训练词对齐，然后使用“grow-diag-and”精炼词对齐[16]。我们利用 Gigaword 语料 Xinhua 的部分，使用 SRILM[29]训练 4 元语言模型，同时采用 Knser-Ney 平滑。译文的质量评价则采用大小写不敏感的 BLEU-4 脚本。MERT 用于在开发集上调整参数权重，以最大化 BLEU 值。

为了训练依存边转换规则生成器，我们将源端和目标端的词汇表限制为两万个出现频率最高的词，分别覆盖了大约 97.41%和 99.13 个中文和英文词。所有的 OOV 词都被映射成特殊的标签 UNK。我们采用随机梯度下降，以及 Adadelta[34]训练模型。Batch 的大小设置为 64，所有的参数维度设置为 300.

5．2 系统

我们采用开源的基于短语的系统 Moses[16]（采用默认的配置）作为我们的基线系统。基于依存边转换的翻译模型基线系统和增加了依存边转换规则生成器的模型采用相同的实验设置。beam 的阈值，beam 的大小和规则数量分别设为 10^{-3},100,100.

5．3 实验结果

表 1展示了三个系统的BLEU值。基线系统“DEBT”的BLEU好于开源的基于短语的系统Moses。在采用依存边转换规则生成器作为辅助后，系统性能在MT03、MT04、MT05测试集上分别提升了+1.23,+1.5和+1.4 BLEU值。这表明，依存边转换规则生成器在解码时能够帮助解码器在解码的过程中选择更匹配的目标端的边。

我们进一步研究上下文信息在多大程度上增强了依存边转换规则生成器的性能。从表 2我们看到，即使没有任何上下文信息，依存边转换规则生成器仍能在一定程度上取得性能提升（均值：33.99比32.99）。这在很大程度上是转换规则生成器的泛化能力所带来的。在解码时，翻译解码器经常会碰到数据稀疏的问题，此时在抽取的规则中找不到完全匹配的源端依存边。DEBT 通过使用前面所提到的泛化后的依存边转换规则来进行翻译。而泛化后的依存边，其头节点或者依存节点被泛化成变量，这进一步削弱了转换规则的上下文匹配能力，且为翻译规则的选择增加了噪音和困难。而依存边转换规则生成器则不存在这个问题，神经网络的数值化特征使得模型在面对未曾出现在训练语料中的依存边时仍能正常处理。

当我们增加上下文的词汇作为输入，在上下文窗口大小变为1时，性能得到了预期中的提升。然而，当我们把上下文的窗口开的更大一点，上下文窗口大小变为2和3，性能反而没有进一步提升。在一定程度上，这表明了上下文对翻译规则选择的重要性，但是也这并不意味着上下文信息越多，规则的匹配能力越强。在我们的实验中，上下文窗口为1，头节点和依存节点左右两边各选1个词时的性能已经足够好了。而且，更大的上下文窗口也意味着计算量的增加。窗口为1时的计算代价也不大。

6 相关工作

Schwenk[27]提出了一个前馈神经网络对源端到目标端的短语片段进行打分。Devlin等人[7]也使用了一个前馈神经网络，每次预测目标端短语的一个词。Auli等人[1]和Cho等人[6]则使用RNN 编码器-解码器结构，学习短语/句子之间的表示。Kalchbrenner 等人[14]提出使用卷积n-gram模型作为编码器和一个混编反向的CGM与RNN作为解码器。这些模型都关注短语或者句子层面的建模。他们的模型设计较为复杂。与他们的模型相比，我们则关注基于句法的翻译规则的表示学习。

Xiong 等人[32]提出了最大熵模型用于预测两个将合并的短语块之间的调序关系。与Xiong等人[32]类似，Van[30]等人提出了用于层次短语调序的模型。Li等人[18][19]分别提出了用于括号转录语法（Inversion Transduction Grammar）和基于短语的翻译的神经网络调序模型。我们的模型主要关注一条依存边中的头节点和依存几点之间的位置关系。我们不仅关注左右相对位置关系，同时关注它们之间是否连续。

7 结论

本文展示了一个基于神经网络的依存边转换翻译规则生成器。我们提出使用编码器-解码器的结构，利用前馈神经网络作为编码器，将源端依存边以及上下文压缩成一个综合向量，然后利用解码器生成目标端的依存边。解码器同时产生翻译对应信息（目标端头节点和目标端依存节点）以及调序信息（目标端头节点左右相对位置关系以及连续位置关系）。采用依存边转换规则生成器后，基于依存边转换规则的翻译基线模型提升了平均+1.39个BLEU点。此外，本文还展示了对于我们的实验设置，少量的上下文已经足够获得可观的性能提升。

参考文献

[1] Michael Auli, Michel Galley, Chris Quirk, and Geoffrey Zweig. Joint language and translation modeling with recurrent neural networks. [C]//Proceedings of the Conference on EMNLP, 2013: 1044-1054.

[2] Peter F Brown, Vincent J Della Pietra, Stephen A Della Pietra, and Robert L Mercer. The mathematics of statistical machine translation: Parameter estimation. [J]//Computational linguistics, 1993. 19 (2) : 263-311.

[3] Pi-Chuan Chang, Huihsin Tseng, Dan Jurafsky, and Christopher D Manning. Discriminative reordering with Chinese grammatical relations features. [C]//Proceedings of the Third

Workshop on Syntax and Structure in Statistical Translation, 2009 : 51-59.
[4] Hongshen Chen, Jun Xie, Fandong Meng, Wenbin Jiang, and Qun Liu. A dependency edge-based transfer model for statistical machine translation. [C]//In Proceedings of COLING, 2014: 23-29.
[5] David Chiang. A hierarchical phrase-based model for statistical machine translation. [C]//Proceedings of ACL, 2005:263-270.
[6] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. [C]//Proceedings of the Conference on EMNLP, 2014:1724-1734
[7] Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. 2014. Fast and robust neural network joint models for statistical machine translation. [C]//Proceedings of ACL, volume 1, 2014:1370-1380.
[8] Yuan Ding and Martha Palmer. Synchronous dependency insertion grammars: A grammar formalism for syntax based statistical MT. [C]//Workshop on Recent Advances in Dependency Grammars (COLING), 2004: 90-97.
[9] Yuan Ding and Martha Palmer. Machine translation using probabilistic synchronous dependency insertion grammars. [C]//Proceedings of ACL, 2005:541-548.
[10] Michel Galley and Christopher D. Manning. A simple and effective hierarchical phrase reordering model. [C]//Proceedings of the Conference on EMNLP, 2008:848-856
[11] Kevin Gimpel and Noah A Smith. Phrase dependency machine translation with quasi-synchronous tree-to-tree features. [J]//Computational Linguistics, 2014
[12] Jonathan Graehl and Kevin Knight. Training tree transducers. [C]//HLT-NAACL,2004: 2004:105-112.
[13] Liang Huang, Kevin Knight, and Aravind Joshi. Statistical syntax-directed translation with extended domain of locality. [C]//Proceedings of AMTA, 2006:66-73.
[14] Nal Kalchbrenner and Phil Blunsom. Recurrent continuous translation models. [C]//Proceedings of the 2013 Conference on EMNLP, 2013:1700-1709
[15] Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. [C]//Proceedings of the North American Chapter of the ACL on Human Language Technology NAACL, 2003:48-54.
[16] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. Moses: Open source toolkit for statistical machine translation. [C]//Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions, 2007:177-180.
[17] Roger Levy and Christopher Manning. Is it harder to parse Chinese, or the Chinese treebank? [C]//Proceedings of ACL, 2003:439-446.
[18] Peng Li, Yang Liu, and Maosong Sun. Recursive autoencoders for ITG-based translation. [C]//Proceedings of EMNLP, 2013:567-577.
[19] Peng Li, Yang Liu, Maosong Sun, Tatsuya Izuha, and Dakun Zhang. A neural reordering model for phrase-based translation. [C]//Proceedings of COLING, 2014:1897-1907.
[20] Dekang Lin. A path-based transfer model for machine translation. [C]//Proceedings of Coling 2004, 2004 : 625-630.
[21] Yang Liu, Qun Liu, and Shouxun Lin. Tree-to-string alignment template for statistical machine translation. [C]//Proceedings of the 21st COLING and the 44th annual meeting of the ACL, 2006:609-616.
[22] Daniel Marcu and William Wong. A phrasebased, joint probability model for statistical machine translation. [C]//Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10, 2002:133-139.
[23] Franz Josef Och and Hermann Ney. Discriminative training and maximum entropy models for statistical machine translation. [C]//Proceedings of the 40th Annual Meeting on ACL, 2002:295-302.
[24] Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. [J]//Computational linguistics, 2003:19-51.
[25] Franz Josef Och. Minimum error rate training in statistical machine translation. [C]//Proceedings of the 41st Annual Meeting on ACL-Volume 1, 2003:160-167.
[26] Chris Quirk, Arul Menezes, and Colin Cherry. Dependency treelet translation: Syntactically informed phrasal SMT. [C]//Proceedings of the 43rd Annual Meeting of the ACL, 2005:271-279.
[27] Holger Schwenk. Continuous space translation models for phrase-based statistical machine

translation. [C]//COLING (Posters), 2012:1071-1080.
[28] Libin Shen, Jinxi Xu, and Ralph Weischedel. A new string-to-dependency machine translation algorithm with a target dependency language model. [C]//Proceedings of ACL, 2008:577-585.
[29] Andreas Stolcke. Srilm-an extensible language modeling toolkit. [C]//Proceedings of ICSLP, 2002:901-904.
[30] Vinh Van Nguyen, Akira Shimazu, Minh Le Nguyen, and Thai Phuong Nguyen. Improving a lexicalized hierarchical reordering model using maximum entropy. [C]//MT Summit XII, Ottawa, Canada, August. 2009
[31] Jun Xie, Haitao Mi, and Qun Liu. A novel dependency-to-string model for statistical machine translation. [C]// Proceedings of EMNLP, 2011:216-226.
[32] Deyi Xiong, Qun Liu, and Shouxun Lin. Maximum entropy based phrase reordering model for statistical machine translation. [C]// Proceedings of the 21st COLING and 44th ACL, 2006:521-528.
[33] Kenji Yamada and Kevin Knight. A syntax-based statistical translation model. [C]// Proceedings of ACL, 2001:523-530.
[34] Matthew D Zeiler. Adadelta: An adaptive learning rate method. [DB]. [2012]. arXiv preprint arXiv:1212.5701.

陈宏申（1991——），男，博士研究生，主要研究领域为自然语言处理，机器翻译。Email:chenhongshen@ict.ac.cn；

刘群（1966——），男，研究员，主要研究领域为自然语言处理，机器翻译。Email:liuqun@ict.ac.cn