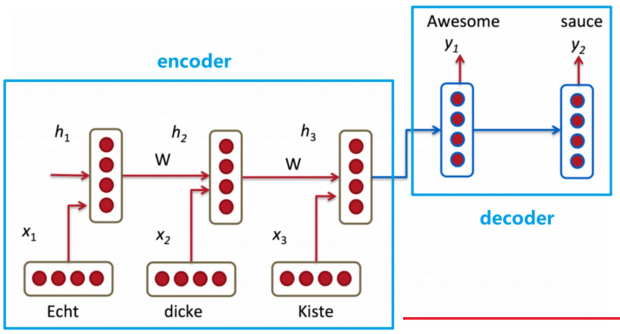
【论文笔记】Neural machine translation by jointly learning to align and translate

【论文】https://arxiv.org/pdf/1409.0473.pdf

论文背景

★ 机器翻译：在给定源句子x的情况下最大化目标句子y的条件概率，即$\arg\max p(y|x)$;

- **传统seq2seq模型（RNN Encoder-Decoder）[1]:**



根据上一个词、hidden state、和上下文表示c预测下一个词y_t:

$$p(y_t|\{y_1,\dots,y_{t-1}\},c)=g(y_t,s_t,c)$$

$$p(y)=\prod_{t=1}^{T}p(y_t|\{y_1,\dots,y_{t-1}\},c)$$

输入$x=(x_1,\dots,x_{T_x})$得到上下文表示c:

$$c=q(\{h_1,\dots,h_{T_x}\})$$

固定长度

- **存在的问题:**
  - encoder总是将所有输入信息编码为一个**固定长度**的向量，这可能会导致网络无法解决长句子，尤其是比训练语料长的句子，**当输入的句子长度上升时，传统的encoder-decoder模型的表现急剧下降。**

论文贡献

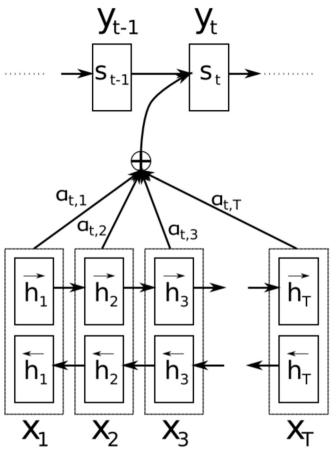
- 引入**注意力机制**;
- 提出了一种新的神经机器翻译模型;

注意力机制解决的问题

对于传统的seq2seq模型潜在的问题就是，在encode输入序列时，该模型框架需要压缩（compress）所有的信息在一个固定长度的向量中，这就会让该模型在面对长句的时候显得十分无力，而且随着序列的增长，句子越前面的词的信息就会丢失的越厉害，虽然有几个小trick来解决这些问题，比如倒序输入句子一遍，或者重复输入两遍，或者使用LSTM模型。但这些都治标不治本，对模型的提升微乎其微，因为在decode时，当前预测词对应的输入词的上下文信息，位置信息等基本都已丢失。

为了解决这个问题，才引入了注意力机制，该机制的本质其实就是引入了**当前预测词对应输入词的上下文信息以及位置信息。**用论文中比较易懂的说法就是，**我们现在这个模型不再是把输入序列编码为一个固定向量，再去解码这个固定向量；我们现在的解码本质上是自动从这个固定向量中抽取有用的信息来解码我的当前词。**

论文模型结构



Decoder

- 当前输出的词由**上一个词、当前hidden state、当前context vector**共同决定:

$$p(y_i|y_1,\dots,y_{i-1},x)=g(y_{i-1},s_i,c_i)$$

With the decoder state s_{i-1}, the context c_i and the last generated word y_{i-1}, we define the probability of a target word y_i as

$$p(y_i|s_i,y_{i-1},c_i)\propto\exp\left(y_i^\top W_o t_i\right),$$

where

$$t_i=\left[\max\left\{\tilde{t}_{i,2j-1},\tilde{t}_{i,2j}\right\}\right]^\top_{j=1,\dots,l}$$

and $\tilde{t}_{i,k}$ is the k-th element of a vector $\tilde{t}_i$ which is computed by

$$\tilde{t}_i=U_o s_{i-1}+V_o E y_{i-1}+C_o c_i.$$

$W_o\in\mathbb{R}^{K_y\times l}$, $U_o\in\mathbb{R}^{2l\times n}$, $V_o\in\mathbb{R}^{2l\times m}$ and $C_o\in\mathbb{R}^{2l\times 2n}$ are weight matrices. This can be understood as having a deep output (Pascanu *et al.*, 2014) with a single maxout hidden layer (Goodfellow *et al.*, 2013).

- 当前hidden state由**上一个hidden state、上一个词、当前的context vector**决定:

$$s_i=f(s_{i-1},y_{i-1},c_i)$$

The hidden state s_i of the decoder given the annotations from the encoder is computed by

$$s_i = (1 - z_i) \circ s_{i-1} + z_i \circ \tilde{s}_i,$$

where

$$\tilde{s}_i = \tanh\left(WEy_{i-1} + U\left[r_i \circ s_{i-1}\right] + Cc_i\right)$$
$$z_i = \sigma\left(W_z Ey_{i-1} + U_z s_{i-1} + C_z c_i\right)$$
$$r_i = \sigma\left(W_r Ey_{i-1} + U_r s_{i-1} + C_r c_i\right)$$

E is the word embedding matrix for the target language. $W, W_z, W_r \in \mathbb{R}^{n\times m}$, $U, U_z, U_r \in \mathbb{R}^{n\times n}$, and $C, C_z, C_r \in \mathbb{R}^{n\times 2n}$ are weights. Again, m and n are the word embedding dimensionality and the number of hidden units, respectively. The initial hidden state s_0 is computed by $s_0 = \tanh\left(W_s \overleftarrow{h}_1\right)$, where $W_s \in \mathbb{R}^{n\times n}$.

- 当前的context vector是由**annotation vector加权**得到：

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})}$$

$$e_{ij} = a(s_{i-1}, h_j)$$

- a表示**对齐模型**，**用来评估当前预测词，与输入词每一个词的相关度。**直观的理解就是在decode的时候，希望decode的信息是直接与预测词相关的，或者说更集中注意于与预测词相关的部分。而这种思想反馈到对齐模型上就是，在生成一个输出词的时候，会考虑每一个输入词与当前词的对齐关系，对齐越好的词，应该享有更大的权重，自然对预测当前词会产生更大的影响；
- 常见的对齐关系计算方式有，点乘（Dot product），权值网络映射（General）和concat映射（multilayer perceptron）几种方式；
- 论文中使用concat映射（多层感知机）方法：$e_{ij} = v_a^T \tanh\left(W_a s_{i-1} + U_a h_j\right)$
- 对齐模型的计算方式如下图：

Encoder

- 使用BiRNN，每个时刻的输出就是一个annotation vector：

$$h_j = \left[\overrightarrow{h}_j^T; \overleftarrow{h}_j^T\right]^T$$

First, the forward states of the bidirectional recurrent neural network (BiRNN) are computed:

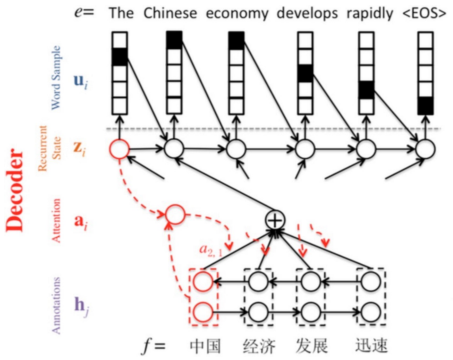
$$\overrightarrow{h}_i = \begin{cases} (1 - \overrightarrow{z}_i) \circ \overrightarrow{h}_{i-1} + \overrightarrow{z}_i \circ \overrightarrow{\underline{h}}_i & , \text{if } i > 0 \\ 0 & , \text{if } i = 0 \end{cases}$$

where

$$\overrightarrow{\underline{h}}_i = \tanh\left(\overrightarrow{W} \overline{E} x_i + \overrightarrow{U}\left[\overrightarrow{r}_i \circ \overrightarrow{h}_{i-1}\right]\right)$$
$$\overrightarrow{z}_i = \sigma\left(\overrightarrow{W}_z \overline{E} x_i + \overrightarrow{U}_z \overrightarrow{h}_{i-1}\right)$$
$$\overrightarrow{r}_i = \sigma\left(\overrightarrow{W}_r \overline{E} x_i + \overrightarrow{U}_r \overrightarrow{h}_{i-1}\right).$$

$\overline{E} \in \mathbb{R}^{m\times K_x}$ is the word embedding matrix. $\overrightarrow{W}, \overrightarrow{W}_z, \overrightarrow{W}_r \in \mathbb{R}^{n\times m}$, $\overrightarrow{U}, \overrightarrow{U}_z, \overrightarrow{U}_r \in \mathbb{R}^{n\times n}$ are weight matrices. m and n are the word embedding dimensionality and the number of hidden units, respectively. $\sigma(\cdot)$ is as usual a logistic sigmoid function.

The backward states $(\overleftarrow{h}_1, \cdots, \overleftarrow{h}_{T_x})$ are computed similarly. We share the word embedding matrix $\overline{E}$ between the forward and backward RNNs, unlike the weight matrices.



实验分析

- **数据集：**英法翻译数据集ACL WMT'14；
- **模型：**
  - **Baseline：**传统的seq2seq模型，RNNenc；
  - **新模型：**RNNsearch；
- **参数设置：**
  - 每种模型训练两次，第一次训练的句子最长为30个词，第二次训练句子最长为50个词；
- **实验结果：**
  - **对齐模型的表现（有效性）**

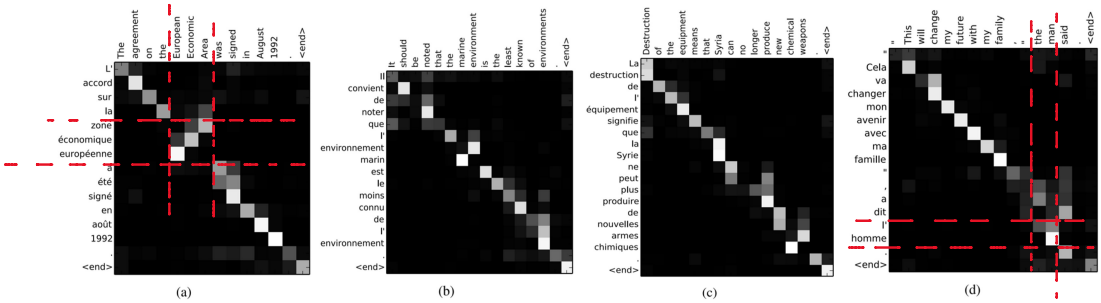


Figure 3: Four sample alignments found by RNNsearch-50. The x-axis and y-axis of each plot correspond to the words in the source sentence (English) and the generated translation (French), respectively. Each pixel shows the weight α_{ij} of the annotation of the j-th source word for the i-th target word (see Eq. (6)), in grayscale (0: black, 1: white). (a) an arbitrary sentence. (b–d) three randomly selected samples among the sentences without any unknown words and of length between 10 and 20 words from the test set.

不考虑什么语法情况的话，理论上应该是这个矩阵的对角线享有最大的权重；

  - **长句翻译的表现**

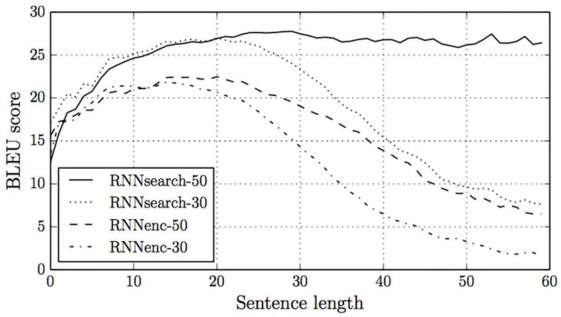


Figure 2: The BLEU scores of the generated translations on the test set with respect to the lengths of the sentences. The results are on the full test set which includes sentences having unknown words to the models.

Model	All	No UNK°
RNNencdec-30	13.93	24.19
RNNsearch-30	21.50	31.44
RNNencdec-50	17.82	26.71
RNNsearch-50	26.75	34.16
RNNsearch-50*	28.45	36.15
Moses	33.30	35.63

Table 1: BLEU scores of the trained models computed on the test set. The second and third columns show respectively the scores on all the sentences and, on the sentences without any unknown word in themselves and in the reference translations. Note that RNNsearch-50* was trained much longer until the performance on the development set stopped improving. (○) We disallowed the models to generate [UNK] tokens when only the sentences having no unknown words were evaluated (last column).

➢ 相关文献：

【1】On the Properties of Neural Machine Translation: Encoder-Decoder Approaches