Neurocomputing 000 (2018) 1–10

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Lattice-to-sequence attentional Neural Machine Translation models

Zhixing Tan[a], Jinsong Su[b], Boli Wang[a], Yidong Chen[a], Xiaodong Shi[a,*]

[a] School of Information Science and Engineering, Xiamen University, China
[b] Software School, Xiamen University, China

ARTICLE INFO

Article history:
Received 27 April 2017
Revised 15 December 2017
Accepted 1 January 2018
Available online xxx

Communicated by Dr. Tie-Yan Liu

Keywords:
Neural Machine Translation
Word lattice
Recurrent Neural Network
Gated Recurrent Unit

ABSTRACT

The dominant Neural Machine Translation (NMT) models usually resort to word-level modeling to embed input sentences into semantic space. However, it may not be optimal for the encoder modeling of NMT, especially for languages where tokenizations are usually ambiguous: On one hand, there may be tokenization errors which may negatively affect the encoder modeling of NMT. On the other hand, the optimal tokenization granularity is unclear for NMT. In this paper, we propose lattice-to-sequence attentional NMT models, which generalize the standard Recurrent Neural Network (RNN) encoders to lattice topology. Specifically, they take as input a word lattice which compactly encodes many tokenization alternatives, and learn to generate the hidden state for the current step from multiple inputs and hidden states in previous steps. Compared with the standard RNN encoder, the proposed encoders not only alleviate the negative impact of tokenization errors but are more expressive and flexible as well for encoding the meaning of input sentences. Experimental results on both Chinese–English and Japanese–English translations demonstrate the effectiveness of our models.

1. Introduction

Recently, NMT [1–4] has achieved great success and attracted much attention in the field of natural language processing (NLP). Different from the statistical machine translation approach that explicitly models latent structures such as word alignment, phrase segmentation and phrase reordering, NMT aims at training a unified encoder-decoder neural network [2,3] that directly maps a source-language sentence to a target-language sentence, where the encoder embeds the input sentence into a sequence of vectors, and then the decoder with attention mechanism produces a translation from the encoded vectors.

Although recent attention has been paid to character-level NMT [5–7] which learn sentence representations and perform generations at the character level, the dominant NMT systems make use of word boundary information to learn the semantic representations of the source sentence and generate the target sentence word by word, typically employing RNNs in both the encoder and the decoder. The reason is that words can encode semantic information more naturally than characters. The word-based models obtain good results for source languages like English, but it does not work equally well for languages without natural word delimiters such as Chinese. The reasons are two-fold. Firstly, the optimal tokenization granularity is not easy to determine for NMT because coarse granularity causes data sparseness while fine granularity results in the loss of useful information. Secondly, the 1-best tokenization may introduce errors that propagate to the later stage and thus negatively impact the learned source sentence representations. Therefore, we believe that it is necessary to learn from more tokenization alternatives for NMT systems to alleviate the sub-optimal tokenization granularity and tokenization error propagation problems.

In this paper, we propose lattice-to-sequence attentional NMT (L2SNMT) models to deal with the above tokenization issues. Word lattice, which compactly represents many tokenization alternatives, has been widely used in various NLP tasks [8–10]. In this paper, we present two lattice-based RNN encoders to simultaneously exploit multiple tokenizations for input sentence modeling. One is the *Lattice-based RNN Encoder with Pre-composition* which first combines the inputs and hidden states vectors derived from multiple tokenizations, and then feeds these vectors into standard RNN. The other is the *Lattice-based RNN Encoder with Post-composition* which learns and updates tokenization-specific vectors for gates, inputs and hidden states, and then generates hidden state vectors for current units. By extending the standard RNN encoder into lattice-based ones, we expect the proposed L2SNMT models not only reduce the negative impact of tokenization errors but also enhance the expressive power and flexibility of encoder in embedding source sentences.

* Corresponding author.
E-mail address: mandel@xmu.edu.cn (X. Shi).

https://doi.org/10.1016/j.neucom.2018.01.010
0925-2312/

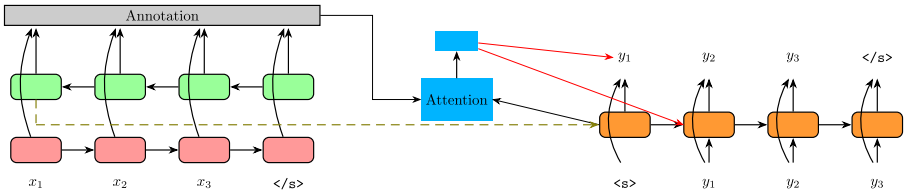

Fig. 1. Overview of attention-based NMT. It has two components: an encoder network and a decoder network with attention mechanism.

We conduct experiments on Chinese–English and Japanese–English translation tasks to investigate the effectiveness of the proposed L2SNMT models. From the experimental results, we conclude that: (1) word boundary information is helpful for NMT to accurately learn the semantics of input Chinese and Japanese sentences. (2) lattice-based RNN encoders are more effective than the standard RNN encoder in NMT. To the best of our knowledge, this is the first attempt to establish NMT encoder based on lattices.

The L2SNMT has been presented in our previous paper [11]. In this article, we make the following significant extensions to our previous work.

1. We integrate lattice posterior score into our proposed NMT models. We also propose a new variant of lattice-based RNN encoder which directly exploits weights associated with lattice input and compare it with other variants.
2. We show that our lattice-based RNN encoders can be further improved with deep RNNs. We also explore the effect of encoder depth that affects the model performance.
3. We investigate different ways to form source annotations from the output of lattice-based RNN.
4. We carry out new experiments on large-scale data to study the robustness of our model on different sizes of training data.
5. We conduct more experiments on Japanese–English translation to investigate the effectiveness of our model on different language pairs.

The remainder of this article is organized as follows: Section 2 briefly describes the conventional sequence-to-sequence attentional NMT, which is the basis of our work. Section 3 gives details of the proposed L2SNMT models. Section 4 reports the experimental results on Chinese–English and Japanese–English translation tasks and studies how the learned encoders improve translation quality. Section 5 summarizes the related work and highlights the differences of our model from previous studies. Finally, we conclude in Section 6 with future directions.

2. Neural Machine Translation

As depicted in Fig. 1, the dominant NMT model is an attention-based NMT [4], which consists of an encoder network and a decoder network with attention mechanism.

Generally, the encoder is modeled as a bidirectional RNN. Given a source sentence $\boldsymbol{x} = x_1, x_2, \ldots, x_S$, the forward RNN reads $\boldsymbol{x}$ word by word in a left-to-right way and uses the recurrent activation function f to learn semantic representation of word sequence $x_{1:i}$ as $\overrightarrow{\mathbf{h}}_i = f(\mathbf{x}_i, \overrightarrow{\mathbf{h}}_{i-1})$, where $\mathbf{x}_i$ is the word embedding of x_i. In a similar way, the backward RNN reversely scans the source sentence and learns the semantic representation $\overleftarrow{\mathbf{h}}_i$ of the word sequence $x_{i:S}$. Then, the hidden states learned by the two RNNs are concatenated to form the *source annotation* $\mathbf{h}_i = [\overrightarrow{\mathbf{h}}_i^T, \overleftarrow{\mathbf{h}}_i^T]^T$, which encodes all the surrounding words of the i-th word.

The decoder is a forward RNN which generates the translation $\boldsymbol{y}$ using a nonlinear function $g(\cdot)$:

$$p(y_t|y_{<t}, \boldsymbol{x}) = g(\mathbf{y}_{t-1}, \mathbf{s}_t, \mathbf{c}_t), \tag{1}$$

where $\mathbf{y}_{t-1}$ denotes the word embedding of word y_{t-1}, $\mathbf{s}_t$ and $\mathbf{c}_t$ are the decoding state and the context vector at the time step t, respectively. Formally, $\mathbf{s}_i$ can be computed using a function $f(\cdot)$, such as Long Short-Term Memory (LSTM [12]) or Gated Recurrent Unit (GRU [2]), as follows:

$$\mathbf{s}_t = f(\mathbf{y}_{t-1}, \mathbf{c}_{t-1}, \mathbf{s}_{t-1}), \tag{2}$$

Furthermore, to accurately capture context, the attention mechanism learns the context vector $\mathbf{c}_t$ as the weighted sum of the source annotations $\{\mathbf{h}_i\}$:

$$\mathbf{c}_t = \sum_{i=1}^{S} \alpha_{t,i} \cdot \mathbf{h}_i, \tag{3}$$

where $\alpha_{t,i}$ measures how well $\mathbf{s}_t$ and $\mathbf{h}_i$ matches as below:

$$\alpha_{t,i} = \frac{\exp(e_{t,i})}{\sum_{i'=1}^{S} \exp(e_{t,i'})}, \tag{4}$$

$$e_{t,i} = \mathbf{v}_a^T \tanh(\mathbf{W}_a \mathbf{s}_t + \mathbf{U}_a \mathbf{h}_i). \tag{5}$$

where $\mathbf{W}_a$, $\mathbf{U}_a$ and $\mathbf{v}_a$ are the weight matrices of attention model. In this way, the relevant source words can be automatically selected to predict target words.

3. The proposed models

In this section, we give a detailed description to the L2SNMT models. We first describe how to build the word lattice of each input sentence. Then we briefly review the standard GRU, which is chosen as the basic unit of our encoder. Next, we describe lattice-based RNN encoders in detail. Finally, we describe the decoder network of our L2SNMT.

3.1. Word lattice

As shown in Fig. 2, a word lattice can be represented as a directed graph $G = \langle V, E \rangle$, where V is the node set and E is the edge set. Given the word lattice of a character sequence $c_{1:N} = c_1, \ldots, c_N$, the node $v_i \in V$ denotes the position between c_i and c_{i+1} and the edge $e_{i:j} \in E$ departs from v_i and arrives at v_j from left to right, covering the subsequence $c_{i+1:j}$ that is recognized as a possible word.

Intuitively, different segmentation results should have different effects when combining inputs and hidden states derived from multiple tokenizations. To do this, we apply *forward–backward algorithm* [13] to calculate the posterior scores of edges in the lattice. Given an edge $e_{i:j}$ in the word lattice, we denote the number of paths from v_0 to v_i with α_i, which is iteratively computed as $\alpha_i = \sum_{k:e_{k,i} \in E} \alpha_k$ and $\alpha_1 = 1$. Similarly, the number of paths from v_N to v_j is denoted as β_j, which is also computed as $\beta_j = \sum_{k:e_{j,k} \in E} \beta_k$ and $\beta_N = 1$.

Finally, we calculate the weight of $e_{i:j}$ as

$$w_{i,j} = \frac{\alpha_i \beta_j}{\sum_{k:e_{k,j} \in E} \alpha_k \beta_j} \tag{6}$$

Please cite this article as: Z. Tan et al., Lattice-to-sequence attentional Neural Machine Translation models, Neurocomputing (2018), https://doi.org/10.1016/j.neucom.2018.01.010