Neural machine translation with Gumbel Tree-LSTM based encoder

Chao Su, Heyan Huang, Shumin Shi, Ping Jian, Xuewen Shi

PII: S1047-3203(20)30061-4
DOI: https://doi.org/10.1016/j.jvcir.2020.102811
Reference: YJVCI 102811

To appear in: *J. Vis. Commun. Image R.*

Received Date: 4 July 2019
Revised Date: 27 February 2020
Accepted Date: 3 April 2020

Journal of Visual Communication and Image Representation (2020)

Contents lists available at ScienceDirect

Journal of Visual Communication and Image Representation

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

Neural machine translation with Gumbel Tree-LSTM based encoder

Chao Su[a], Heyan Huang[a,b,*], Shumin Shi[a,b], Ping Jian[a,b], Xuewen Shi[a]

[a] *School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China*
[b] *Beijing Engineering Research Center of High Volume Language Information Processing and Cloud Computing Applications, Beijing 100081, China*

ARTICLE INFO

Article history:
Received 1 May 2013
Received in final form 10 May 2013
Accepted 13 May 2013
Available online 15 May 2013

Communicated by xxx

Keywords:
Neural machine translation
Tree to sequence
Gumbel Tree-LSTM

ABSTRACT

Neural machine translation has improved the translation accuracy greatly and received great attention of the machine translation community. Tree-based translation models aim to model the syntactic or semantic relation among long-distance words or phrases in a sentence. However, it faces the difficulties of expensive manual annotation cost and poor automatic annotation accuracy. In this paper, we focus on how to encode a source sentence into a vector in a unsupervised-tree way and then decode it into a target sentence. Our model incorporates Gumbel Tree-LSTM, which can learn how to compose tree structures from plain text without any tree annotation. We evaluate the proposed model on both spoken and news corpora, and show that the performance of our proposed model outperforms the attentional seq2seq model and the Transformer base model.

1. Introduction

Neural machine translation (NMT) models have made a big improvement over the traditional statistical ones. NMT usually uses a encoder-decoder framework. For example, the recurrent neural network (RNN) based NMT models [1, 2] typically uses RNN [3] or its variant, long short-term memory (LSTM) [4], to encode a sequential source sentence into a vector and then decode the vector to a sequential target sentence, which are called sequence-to-sequence (seq2seq) models. However, due to their sequential property, it is hard for seq2seq models to learn the syntactic and semantic relations among long-distance words. Vaswani et al. [5] proposed a self-attention based model, Transformer, which has been widely used in the NMT community to build a strong baseline. Transformer can process tokens in parallel rather than sequentially and better learn dependencies among words. However, it does not figure out how these individual words compose a sentence.

Natural language has its own tree-like underlying structure, and linguists agree that syntax rules dictate how single words compose a sentence in a tree-like way [6, 7]. Recent studies have pointed out that explicitly modeling tree structures may be helpful to improve natural language processing tasks. Parsing trees have been shown to be helpful for machine translation [8, 9, 10, 11], sentiment analysis [12, 13, 14], word sense disambiguation [15, 16, 17, 18] and other tasks. Eriguchi et al. [8] proposed a tree-to-sequence (tree2seq) model, which encodes a source sentence into a vector following its syntactic structure and then decodes the vector into target tokens. Since the annotation of syntactic trees is very expensive, they used trees generated from parsing tools rather than human annotation to perform their tree2seq translation. However, using automatically generated trees can lead to translation errors due to parsing errors. Therefore, it is necessary to study to improve the performance of neural machine translation models with unsupervised tree structures.

Learning the formal syntactic grammar without relying on any linguistic annotations has been studied in statistical machine translation (SMT) and has made a big improvement [19].

*Corresponding author.
e-mail: hhy63@bit.edu.cn (Heyan Huang)

In this paper, we propose to learn the source tree structure formally for NMT. We use Gumbel Tree-LSTM [20] to learn how to compose tree structures from words and encode the compositions into vectors, and then utilize the compositions as contextual information when decoding. Given some words as input, at each step, Gumbel Tree-LSTM computes all possible composition candidates and selects one from them through measuring the composition validity scores. This process is performed recursively until only one representation, i.e. the root of the tree, remains.

We apply the unsupervised tree encoder on two important models in the MT research community, Recurrent NMT (RNMT) and Self-Attention based one (SA). In both cases, the outputs of the LSTM or self-attention based encoder are fed into our tree encoder to learn a tree structure, and then all the original encoder outputs and our tree node representations are fed into the decoder as contextual information. The difference is that, the RNMT model uses a traditional attention mechanism to utilize the tree representations, as Eriguchi et al. [8] did, while the SA based model utilize a self-attention based context attention. The both attention mechanisms make NMT models give attention to both leaf (words) and non-leaf (phrases) nodes in the source tree and tells us which words or phrases are important when decoding a target word.

The main contributions of our work are summarized as follows:

- We adopt an unsupervised tree learning technology, Gumbel Tree-LSTM, to improve NMT models. The learned tree representations are fed into decoders as contextual information so that the model could learn to select which phrases or words are important when decoding.
- We show from experimental results that the proposed model outperforms both the attentional recurrent NMT and self-attention based Transformer models. It proves the effectiveness of unsupervised tree learning method for NMT.
- We analyze the learned trees in detail and show that the proposed approach is able to learn syntactic and semantic structures, which also helps to enhance the interpretability of the NMT methods.

In the remaining parts of this paper, we briefly introduce tree-structured encoders and decoders as related work in Section 2. We describe the proposed unsupervised tree-to-seq translation model in Section 3. Our experimental setup and results are presented in Section 4. Then we give and analyzed some tree examples learned by the model in Section 5. Lastly we conclude the paper and give some future work.

2. Tree-Structured Neural Models

Deep neural networks have been one of the most successful models for many kinds of artificial intelligent tasks, such as computer vision, natural language processing, and cross-modal retrieval [21, 22, 23, 24, 25]. These methods include convolutional neural networks (CNN) [26], recurrent neural networks (RNNs) [3], and Recursive neural network (RvNNs) [12]. RvNN is a classic tree-structured model. There are also some tree models based on RNN and long short-term memory (LSTM) [4]. Recently, researchers begin to design models those can learn tree structures from plain text and use them to improve down-stream tasks. Here we briefly introduce some recent variants of them which fall into three categories: encoders, decoders, and unsupervised trees.

2.1. Tree-Structured Encoders

Both Zhu et al. [13] and Tai et al. [14] extended LSTM to tree structures, or they added LSTM to RvNNs, in which each unit is able to incorporate information from multiple child units. They applied the model into predicting the sentiment of sentences and phrases, and outperformed the recursive network. Tai et al. [14] pointed out that sequence LSTM can be regarded as a special case of the Tree-LSTM. Chen et al. [27] proposed to use a hybrid model of chain-LSTM and tree-LSTM with syntactic information to improve the inference accuracy.

To compare the recursive (tree-structured) models with the recurrent (sequencial) models, Li et al. [28] conducted experiments on four tasks: sentiment classification, phrase matching, semantic relation classification, and discourse parsing. They concluded that tree structured models were better on tasks which need to process long distance dependency and long sequences. To fully utilize both low-level and high-level visual information or language context information, Gao et al. [29] proposed an adaptive attention model based on hierarchical LSTMs to generate image caption.

All of the above encoders are used in other tasks rather than machine translation. Eriguchi et al. [8] proposed a tree-to-sequence translation model to leverage source syntax and investigated the attentional mechanism on source trees. The unit at non-leaf tree nodes they used was Tree-LSTM proposed by Tai et al. [14]. Chen et al. [30] expanded the bidirectional sequential encoder into tree structured one and proposed a bidirectional tree encoder with GRU, in which each node was calculated according to both its children and its parent.

2.2. Tree-Structured Decoders

Compared with utilizing syntax information in encoders, it is harder for decoder to do that. The source syntax can be get by parsing tools, but prediction of target tree structures is usually hard.

Both Zhang et al. [31] and Zhou et al. [32] proposed generative neural machines for dependency tree structures. The main difference is that, Zhang et al. [31] redefined dependency paths, and predicted the tree nodes in an particular order with four RNNs. While Zhou et al. [32] firstly transformed the dependency tree into a ternary full tree, and then developed a model to capture the dependencies in K-ary full tree. They focused on the decoder only and assumed that the encoding has been finished.

Alvarez-Melis and Jaakkola [33] proposed a doubly recurrent neural network, in which each tree node receives information from its ancestor and previous sibling. We recently found that Chen et al. [34] used tree-to-tree networks for program translation, which provided evidence for the tree-to-tree model's effectiveness. But our preliminary experimental results did not show that linguistic tree-to-tree model is suitable for the natural language translation. It may be due to the abnormal phenomena on syntax and poor parsing accuracy. We focus on the tree encoder in this paper and leave the unsupervised tree decoder in the future.

2.3. *Unsupervised Trees*

In all of the above work, tree structures are given to guide the composition orders. Recently, researchers begin to train neural networks that predict the tree structure of a sentence and use the generated tree to represent the sentence, without any training parse trees given, which is called *unsupervised tree learning* or *latent tree learning*.

Choi et al. [20] applied the Straight Through Gumbel-Softmax estimator [35] to perform unsupervised tree learning and used the generated trees to improve natural language inference and sentiment analysis tasks. We use the Gumbel Tree-LSM proposed by them as the encoder unit to perform machine translation. Yogatama et al. [36] used reinforcement learning [37, 38] to get the best tree structures. Williams et al. [39] reimplemented and compared the two previous work in detail and suggested that though the learned trees are not like those in PTB, they are helpful in the semantic representation of sentences.

Shen et al. [40] proposed a Parsing-Reading-Predict-Networks (PRPN) language model, which consists of three parts, including Parsing Network, Reading Network, and Predict Network. First, the Parsing Network computes the syntactic distances between all pairs of words using a convolutional network. The syntactic distance represents the syntactic relationships among words. Then, the Reading network reads all memories that are related to the current token and computes a representation. At last, the Predict Network predicts the next token for language modeling. Le and Zuidema [41] proposed a Forest Convolutional Network to incorporate all semantic information from all child nodes rather than selecting a most possible parse.

Kim et al. [42] proposed unsupervised Recurrent Neural Network Grammars (URNNG). RNNG [43] is a effective supervised technique for language modeling and transition-based parsing. While URNNG samples binary trees from the outputs of its inference network rather than reading from annotated data. Shen et al. [44] added a master input gate and a master forget gate into LSTM to model the updating information in constituent trees, in which high-ranking neurons were updated less frequently while low-ranking neurons were more frequently updated.

Tran and Bisk [45] used structured self-attention to learn implicit dependency structures and feed them into a shared attention to improve translation results. Drozdov et al. [46] performed a bottom-up inside and a top-down outside algorithm for each tree node to learn a better representation.

In all these unsupervised tree learning techniques, we choose a promising one, Gumbel Tree-LSTM to learn semantic compositions for NMT.

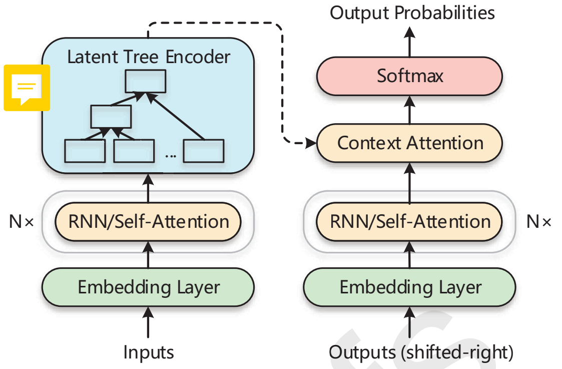


Fig. 1. Architecture of our tree-based translation model

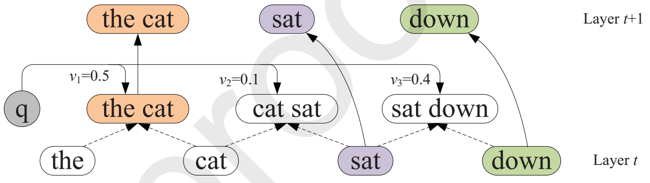


Fig. 2. An example of the parent selection from Refs. [20] and [39]

3. Gumbel Tree-LSTM for NMT

We propose an unsupervised tree2seq architecture for machine translation. We incorporate a Gumbel Tree-LSTM based encoder into both RNN and self-attention based translation models. In the left part of Figure 1 (encoder), the embeddings of an input source sequence are fed into the RNN/self-attention layers, then the output of the last layer are fed into the latent tree encoder to learn tree structures of the source sentence; In the right part (decoder), embeddings of a right-shifted target sequences are also fed into the RNN/self-attention layers, then both the leaf nodes (corresponding to encoder outputs in a standard encoder-decoder model) and the non-leaf nodes (providing additional tree structure information in this paper) from the tree encoder are fed into the decoder's attention module as context information for decoding. The output sequence is generated through a softmax layer at last.

3.1. *Latent Tree Encoding*

We use Gumbel Tree-LSTM [20] to perform our latent tree encoding. Gumbel Tree-LSTM is a tree-structured architecture that is able to learn how to compose tree structures only from plain text data without any annotation. It recursively select the best composition from all candidates in a bottom-up way. It uses Straight-Through (ST) Gumbel-Softmax estimator [35] to get discrete samples in the forward pass and propagate continuous error signals in the backward pass.

A core task of unsupervised parsing is to select the most possible composition from all candidates. An example of the parent selection process of Gumbel Tree-LSTM is shown in Figure 2. First, given a sequence of vector representations at layer t (*the*, *cat*, *sat*, *down*), parents of each two adjacent nodes is computed as candidates (*the cat*, *cat sat*, and *sat down*) using the Tree-LSTM. Then a trainable query vector $\mathbf{q}$ is used to calculate the validity score of each candidate and select the single best one. The selected parent (*the cat*) and not-selected nodes (*cat*, *down*) are copied to the corresponding positions at layer $t+1$. This parent selection procedure is repeated until only a single node is left and then the tree construction is completed.

Calculation details are described below. The Tree-LSTM has been proposed by Tai et al. [14] and Zhu et al. [13]. We use the same model setup as Choi et al. [20] here. The validity score is computed as:

$$v_i = \frac{\exp(\mathbf{q} \cdot \tilde{\mathbf{h}}_i^{t+1})}{\sum_{j=1}^{M_{t+1}} \exp(\mathbf{q} \cdot \tilde{\mathbf{h}}_j^{t+1})}, \tag{1}$$

To sample a parent, we need to use the ST Gumbel-Softmax estimator. Suppose that π_i is the validity score before normalization, i.e., $\pi_i = \exp(\mathbf{q} \cdot \tilde{\mathbf{h}}_i^{t+1})$. The Gumbel-Softmax distribution is calculated by

$$o_i = \frac{\exp((\log(\pi_i) + g_i)/\tau)}{\sum_{j=1}^{k} \exp((\log(\pi_j) + g_j)/\tau)}, \tag{2}$$

where g_i is the Gumbel noise and τ is a temperature parameter [20, 35, 47]. In the forward pass, the ST Gumbel-Softmax estimator discretizes the continuous probability vector $\mathbf{o}$ into the one-hot vector o^{ST},

$$o_i^{ST} = \begin{cases} 1, & i = \mathrm{argmax_j o_j} \\ 0, & \text{otherwise} \end{cases}. \tag{3}$$

Note that $\mathbf{o}^{ST}$ in Equation (3) is discrete and thus can not be used to backpropagate. The trick is here. The ST Gumbel-Softmax estimator use the Gumbel-Softmax distribution $\mathbf{o}$ in Equation (2), which is continuous, to replace the discrete one in backward pass. It is also noteworthy that the Gumbel-Softmax distribution has been proved to be able to approximate the discrete one when $\tau \to 0$ [35, 47]. Suppose that the selected two nodes are $\mathbf{r}_i^t$ and $\mathbf{r}_{i+1}^t$, then the new nodes at $(t+1)$-th layer will be

$$\mathbf{r}_j^{t+1} = \begin{cases} \mathbf{r}_j^t, & j < i \\ \text{Tree-LSTM}_1(\mathbf{r}_j^t, \mathbf{r}_{j+1}^t), & j = i \\ \mathbf{r}_{j+1}^t, & j > i \end{cases}. \tag{4}$$

3.2. *Integrating Latent Trees into NMT*

When the latent tree learning is completed, we integrate the learned tree structures to guide the decoding in NMT. Specifically, we apply the tree model on two popular NMT models, an RNN based one (RNMT) and a self-attention based one (Transformer). Due to their different properties, the processes of integration are not exactly the same.

3.2.1. *RNMT Model*

Figure 3 shows the structure of our unsupervised tree-enhanced RNMT model. The model consists of an LSTM encoder (denoted as blue nodes $\mathbf{x}_i$), an LSTM decoder (red nodes $\mathbf{s}_i$), a latent tree encoder described in Section 3.1 (green nodes $\mathbf{r}_i^n$), and an attention module. Assume that a sequence of embeddings ($\mathbf{e}_1$, ..., $\mathbf{e}_N$) of source tokens is fed into the LSTM layer(s) and the blue nodes $\mathbf{x}_i$ denote the last layer's state at the i-th time step. When all time steps have been completed, all the states of the last layer are fed into the latent tree encoder to build a tree structure, denoted as green nodes $\mathbf{r}_i^n$. Then, like Eriguchi et al. [8] did, we initialize the decoder state $\mathbf{s}_1$ using both the tree root node $\mathbf{r}_1^N$ and the LSTM state at the last time step $\mathbf{x}_N$.When decoding the j-th step, both the encoder states (blue $\mathbf{x}_i$) and the non-leaf tree nodes (green $\mathbf{r}_i^{>1}$) are treated as context information to compute the attention weights $\alpha_j(i,t)$ and the attention vector $\mathbf{d}_j$. At last, both the original decoder hidden state $\mathbf{s}_j$ and the attention vector $\mathbf{d}_j$ are fused to generate the probability of $\mathbf{y}_j$.

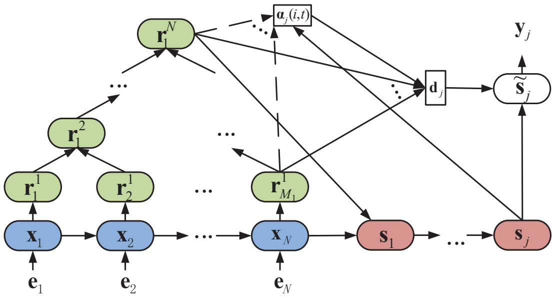


Fig. 3. Unsupervised tree-to-sequence attentional NMT model

Calculation details are described below. Assume that a source sentence has been fed into an embedding layer and then RNN or self-attention layers, and the outputs is a sequence of vectors ($\mathbf{x}_1$, ..., $\mathbf{x}_N$). We apply a linear transformation to initial the leaf nodes of the tree:

$$\mathbf{r}_i^1 = \begin{bmatrix} \mathbf{h}_i^1 \\ \mathbf{c}_i^1 \end{bmatrix} = \mathbf{W}_{leaf}\mathbf{x}_i + \mathbf{b}_{leaf}, \tag{5}$$

where $\mathbf{r}_i^t$ denotes the i-th node at t-th layer of the tree, and $\mathbf{h}_i^t$ and $\mathbf{c}_i^t$ denotes the corresponding hidden and cell state.

The initialization of decoder state used an special Tree-LSTM unit, which treats the last LSTM state $\mathbf{x}_N$ and the tree root $\mathbf{r}_1^N$ as left and right child respectively:

$$\mathbf{s}_1 = \text{Tree-LSTM}_2(\mathbf{x}_N, \mathbf{r}_1^N). \tag{6}$$

Note that the numbers of tree layers and source tokens are equal (N), because the Gumbel Tree-LSTM induce only two children at each layer.

Attention mechanism has been proved helpful to select which words or phrases are more important to the current decoding node [48]. We also incorporate the attention mechanism to do this. The attention is applied on all the nodes including leaf and non-leaf ones in the source composition tree. As shown in Figure 3, the attention score $\alpha_j(i,t)$ between the i-th node at layer t of source tree and the j-th target state is calculated as:

$$\alpha_j(i,t) = \frac{\exp(\mathbf{r}_i^t \cdot \mathbf{s}_j)}{\sum_{l=1}^{N} \sum_{k=1}^{M_l} \exp(\mathbf{r}_k^l \cdot \mathbf{s}_j)}, \tag{7}$$

where $\mathbf{r}_k^l \cdot \mathbf{s}_j$ is the inner product of the k-th source hidden state at layer l and the j-th target hidden state. The context vector is

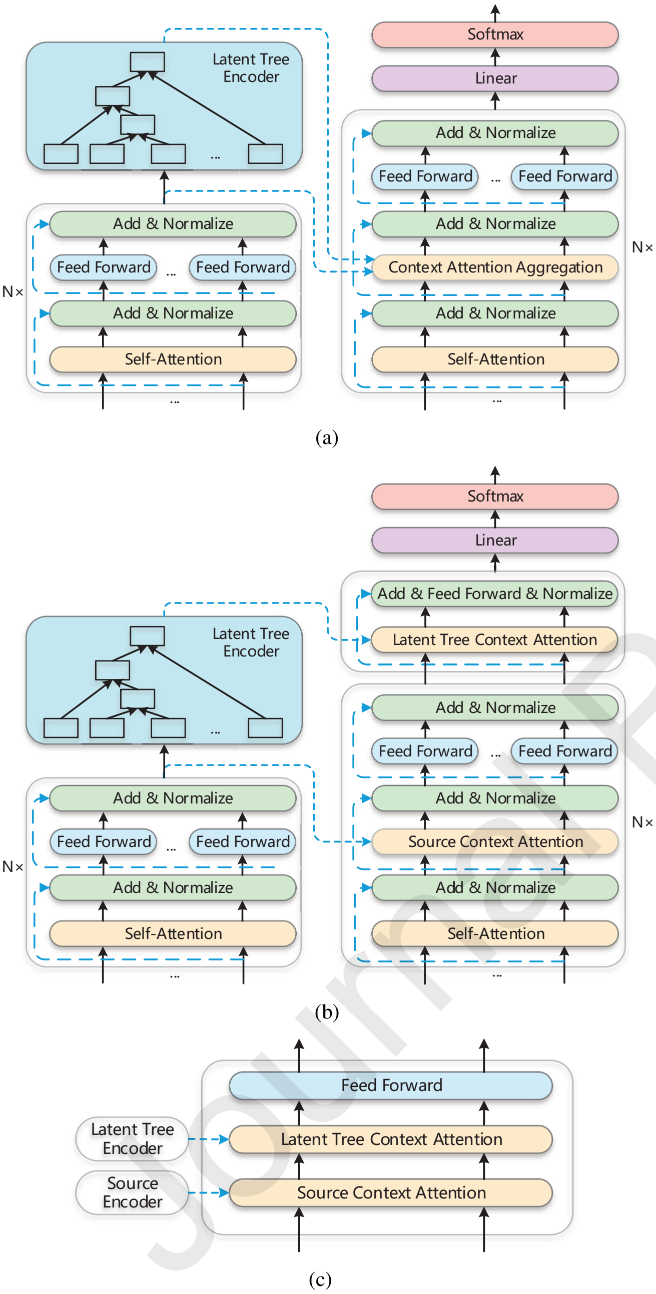


Fig. 4. Integrating latent tree encoder into Transformer model. (a) Aggregating tree memories with the original context attention (*agg*). (b) Leveraging tree memories through an individual attention above the decoder (*ind*). (c) Aggregation of source context attention and latent tree attention in (a).

calculated as:

$$\mathbf{d}_j = \sum_{t=1}^{N}\sum_{i=1}^{M_t}\alpha_j(i,t)\mathbf{r}_i^t. \tag{8}$$

When decoding, an additional hidden layer $\tilde{\mathbf{s}}_j$ is used to predict the j-th word:

$$\tilde{\mathbf{s}}_j = \tanh(\mathbf{W}_d[\mathbf{s}_j;\mathbf{d}_j] + \mathbf{b}_d), \tag{9}$$

The j-th word is finally predicted by:

$$p(y_i|\mathbf{y}_{<j},\mathbf{x}) = \mathrm{softmax}(\mathbf{W}_s\tilde{\mathbf{s}}_j + \mathbf{b}_s). \tag{10}$$

Our training object is to minimize the cross entropy loss, or to maximizing the likelihood:

$$J(\theta) = \frac{1}{|\mathcal{D}|}\sum_{(\mathbf{x},\mathbf{y})\in\mathcal{D}}\log p(\mathbf{y}|\mathbf{x}). \tag{11}$$

where θ is the whole parameters to train and $\mathcal{D}$ is the training set.

3.2.2. Transformer Model

The latent tree encoder can also be used in the self-attention based translation model, Transformer (see Figure 4). Similar to the RNN based model, the outputs of the Transformer encoder are fed into a linear transformation and then used as the hidden and cell states of the first layer of the tree, and then a tree structure is learned based on them. When the tree construction is completed, the learned compositions (phrasal vectors) are used as outputs of our latent tree encoder, i.e., our tree memory bank. At last, both the original encoder outputs and our tree memory bank are fed into the decoder.

In the decoder, we propose two ways to use the tree memory bank. First, the tree memory bank can be aggregated with the original context attention in each Transformer decoder layer (*agg*, Figure 4 (a)). Second, it can also be used independently after all Transformer decoder layers computed, i.e., above all the Transformer decoder layers (*ind*, Figure 4 (b)).

agg: In this setting, each decoder layer is forced to attend to both the original source memory bank and our latent tree memory bank. The new context attention is showed in Figure 4 (c). In Figure 4 (c), we have two context attention modules, latent tree context attention module and the original source context attention module, which attend to the latent tree encoder and source encoder respectively. The two modules are cascaded. It means that the inputs are fed into the source attention for attending on source words, and then fed into our latent tree attention for augmenting target words selection through attending on source tree constituents. At last the outputs are fed into a fully-connected feed-forward layer.

ind: We also try to position the aggregation of the tree attention above the Transformer decoder (shown in 4 (b)). In this setting, all the original Transformer decoder layers are not inflected by the tree memory bank, and tree context attention is performed with outputs from the last layer of Transformer decoder as queries. By doing this, the model only attend the tree attention only once. Thus, this will use less parameters. And experiments show that this leads to better results.

4. Experiments

4.1. Data

Table 1. Statistics of datasets.

Data	# sent pairs	# src tokens	# tgt tokens
BOLT (train)	121,078	844,552	1,122,261
BOLT (dev)	4,935	37,727	49,148
BOLT (test)	4,977	31,315	37,730
ASPEC (train)	1,500,000	39,642,405	47,987,387
ASPEC (dev)	1,790	49,148	53,960
ASPEC (test)	1,812	45,035	53,719
LDC Zh-En	1,678,061	47,575,832	54,859,204
MT02	878	22,350	25,339
MT03	919	23,992	25,999
MT04	1,597	43,128	46,952
MT05	1,082	29,475	30,882
MT06	1,664	37,822	41,014
MT08	1,357	32,042	37,307
MT12	820	21,321	25,316
MT08-12	1,370	30,935	36,043
WMT14 En-De	4,520,620	117,013,968	110,233,786
newstest2013	3,000	64,807	63,412
newstest2014	2,737	61,752	57,987

To test the performance of the proposed architecture, we apply our approach on four datasets, BOLT, ASPEC, LDC Zh-En, and WMT14 En-De. Statistics of the datasets are shown in Table 1.

BOLT: We use BOLT (Broad Operational Language Translation) corpora[1] [49] to verify the model performance on Chinese-English spoken language. Those corpora come from the NIST OpenMT'15 Challenge, including chat messages (CHT), short message service (SMS), and conversational telephone speech (CTS) from DARPA BOLT Project. We extract the CHT part from all the corpora and split them into three parts, train (121,078 sentence pairs), dev (4,935 sentence pairs), and test (4,977 sentence pairs) for training, validation, and test respectively.

ASPEC: ASPEC (Asian Scientific Paper Excerpt Corpus) is a parallel corpus adopted by the 2nd Workshop on Asian Translation (WAT'15)[2] for En-Ja and Zh-Ja translation in scientific papers domain. The En-Ja corpus contains 3.0 million sentence pairs now. We followed Eriguchi et al. [8] to compare the effectiveness on the first 1.5 million sentence pairs. We segmented all the Japanese sentences using KyTea[3] [50], and cleaned out the training sentence pairs which contained above 50 tokens, as described in Eriguchi et al. [8]. Other pre-processing steps were performed as recommended in WAT'15[4]. Statistics of the data can be seen in Table 1.

LDC Zh-En: We also evaluate the proposed approach on Chinese-English news datasets extracted from LDC corpora[5]. The training corpora totally contains 1.68 million sentence pairs. The Chinese sentences were word-segmented by a segmentor from LTP [51]. Since part of the corpora (LDC2003E14) is not sentence-aligned, we sentence-aligned the corpus using the Champollion Toolkit (CTK)[6] [52]. We used the NIST 2002 (MT02) testset for validation. And all other testsets of NIST, including NIST 2003 (MT03), NIST 2004 (MT04), NIST 2005 (MT05), NIST part of NIST 2006 (MT06), NIST 2008 (MT08), NIST 2012 General (MT12), and NIST 2008-2012 Progress (MT08-12) were used for testing.

WMT14 En-De: The En-De corpus provided by WMT'14[7] contains 4.5 millions sentence pairs. We use the newstest2013 for validation and the newstest2014 for testing, which contains 3,000 and 2,737 sentence pairs respectively.

4.2. Evaluation

We evaluated translation results through three automatic metrics: case-insensitive BLEU score [53] for all translation results, RIBES score [54] for En-Ja results, and Meteor score [55] for Zh-En and En-De results. BLEU score is a widely used language-independent metric for machine translation, which is based purely on *n*-gram counting. Meteor is a language specific metric. It takes stemming, synonym, and paraphrase into account. Thus, it needs language specific resources, such as the WordNet, to provide the above information. RIBES, which considers the word order into account, is developed to better evaluate translation between distant language pairs such as Japanese and English.

4.3. Training Details

We employed our approach on two popular baselines in NMT: an recurrent NMT model with attention mechanism (RNMT) and a self-attention based Transformer model. The RNMT model contained an encoder with 3-layer bidirectional LSTMs and a 3-layer LSTM decoder. The Transformer model was a base version in which both encoder and decoder contain 6-layer self-attention networks.

Both models were set with 512 embedding dimension, 512 hidden dimension, and were trained with Adam optimizer [56] with $\beta_1 = 0.9$ and $\beta_2 = 0.998$. The learning rate was scheduled as described in [5] with $warmup\_steps = 8000$. Dropout probability was set to be 0.1 to avoid over-fitting. For the BOLT, ASPEC, LDC Zh-En, and WMT14 En-De corpora, we used the top frequency 20K, 60K, 30K, and 50K words as their vocabulary respectively. Batch size was set to 2048 tokens for the former two corpora, and 4096 tokens for the latter two corpora. When learning a Gumbel tree, we use the same temperature $\tau = 1.0$ with Choi et al. [20].

All of the models were trained with 200K steps, and the one with highest accuracy on validation set was selected to perform

[1] LDC2013E80, LDC2013E81, LDC2013E83, LDC2013E85, LDC2013E118, LDC2013E125, LDC2013E132, LDC2014E08, LDC2014E69, LDC2013E119

[2] `http://orchid.kuee.kyoto-u.ac.jp/WAT/WAT2015/`

[3] `http://www.phontron.com/kytea/`

[4] `http://orchid.kuee.kyoto-u.ac.jp/WAT/WAT2015/baseline/dataPreparationJE.html`

[5] LDC2002E18, LDC2003E07, LDC2003E14, LDC2004T07, Hansards portion of LDC2004T08, and LDC2005T06

[6] `https://www.ldc.upenn.edu/language-resources/tools`

[7] `http://www.statmt.org/wmt14/translation-task.html`

Table 2. Experimental results on BOLT, ASPEC, and WMT14 En-De corpora

Model	BOLT		ASPEC		WMT14 En-De	
	BLEU	Meteor	BLEU	RIBES	BLEU	Meteor
RNMT	13.56	21.57	37.50	82.53	20.83	41.35
+GTree	13.75	21.90*	37.67	82.72	21.62*	42.28*
Transformer base	15.05	23.12	39.22	83.43	21.59	42.26
+GTree-agg	**15.59**$^{\#}$	**23.42**$^{\#}$	**39.97**$^{\#}$	**83.85**	**22.56**$^{\#}$	**43.14**$^{\#}$
+GTree-ind	15.43$^{\#}$	23.24	39.78$^{\#}$	83.76	21.97$^{\#}$	42.95$^{\#}$

Table 3. Results of base model on the Zh-En training data

Test set	BLEU score (%)					Meteor score (%)				
	RNMT	+GTree	Transformer	+GTree-agg	+GTree-ind	RNMT	+GTree	Transformer	+GTree-att	+GTree-ind
MT03	36.89	37.47* (+0.58)	41.32	**41.80**$^{\#}$ (+0.48)	41.24 (-0.08)	24.94	25.49* (+0.55)	27.00	**27.47**$^{\#}$ (+0.47)	27.21 (+0.21)
MT04	37.95	39.19* (+1.24)	42.28	**42.73**$^{\#}$ (+0.45)	42.55 (+0.27)	24.18	24.62* (+0.44)	25.66	**26.13**$^{\#}$ (+0.47)	26.07 (+0.41)
MT05	34.59	34.36 (-0.23)	39.14	39.76 (+0.62)	**39.98**$^{\#}$ (+0.84)	24.39	24.53 (+0.14)	26.68	**26.82** (+0.14)	26.74 (+0.06)
MT06	31.91	32.20 (+0.29)	35.38	35.53 (+0.15)	**36.09**$^{\#}$ (+0.71)	23.40	23.54 (+0.14)	25.08	25.58$^{\#}$ (+0.50)	**25.66**$^{\#}$ (+0.58)
MT08	25.62	25.58 (-0.04)	29.24	**30.07**$^{\#}$ (+0.83)	29.41 (+0.17)	19.73	19.96 (+0.23)	20.92	**21.58**$^{\#}$ (+0.66)	21.39$^{\#}$ (+0.47)
MT08-12	23.01	24.02* (+1.01)	26.78	27.09 (+0.31)	**27.33**$^{\#}$ (+0.55)	19.18	19.75* (+0.57)	20.78	21.32$^{\#}$ (+0.54)	**21.46**$^{\#}$ (+0.68)
MT12	20.28	20.65 (+0.37)	22.43	**23.61**$^{\#}$ (+1.18)	22.70 (+0.27)	17.55	17.93 (+0.38)	18.35	19.14$^{\#}$ (+0.79)	**19.23**$^{\#}$ (+0.88)
Avg	30.04	30.50* (+0.46)	33.80	**34.37** (+0.57)	34.19 (+0.39)	21.91	22.26 (+0.35)	23.50	**24.01**$^{\#}$ (+0.51)	23.97$^{\#}$ (+0.47)

our testing. In testing phase, we performed a beam search algorithm with a beam size of 5. To test whether a performance difference was statistically significant, we conducted significance tests following Reference [57] for BLEU and Reference [58] for Meteor.

Our code is available at `https://github.com/chao-su/gumbel-nmt`.

4.4. Results

Results on BOLT, ASPEC, and WMT14 En-De datasets. Table 2 shows the results on BOLT, ASPEC, and WMT14 En-De datasets. For Chinese-English spoken language translation (BOLT), our approach gets +0.19 BLEU and + 0.33 Meteor improvement on RNMT model, and +0.54 BLEU and +0.30 Meteor improvements than Transformer base model. For the English-Japanese scientific translation (ASPEC), our approach gets +0.75 BLEU and +0.42 RIBES improvements than Transformer base model. For the English-German translation, our approach gets +0.79 BLEU and +0.93 Meteor improvements than RNMT model, and +0.97 BLEU and +0.88 Meteor improvements than Transformer base model. The signs * and # represent denote that the result is significantly better than RNMT and Transformer respectively (at significance level $p < 0.05$).

Results on LDC Zh-En dataset. Tabel 3 shows the result on LDC Zh-En dataset. For Chinese-English news translation, our approach gets +0.46 BLEU and +0.35 Meteor improvements than RNMT model in average, and gets +0.57 BLEU and +0.51 Meteor improvements than Transformer base model in average. The signs * and # denote that the result is significantly better than RNMT and Transformer respectively (at significance level $p < 0.05$).

5. Analysis

5.1. Learned English Trees

We show two learned tree structures of English sentences in Figures 5 and 6. Their corresponding binary trees parsed by Enju [59] can be seen in Eriguchi et al. [8]. Through comparison analysis, we found that they have some same points.

One of the similar point is that, in the example of Figure 5, both of them split the whole sentence into two parts: "SiO2 films ... or less" and "and the ... confirmed". This may be easily learned through the explicit comma. We also find some phrases which have linguistic meaning, such as "430 °C or less", "memory effect" etc. These results support the analysis in Williams et al. [39]. The model tends to combine the initial or final two words.

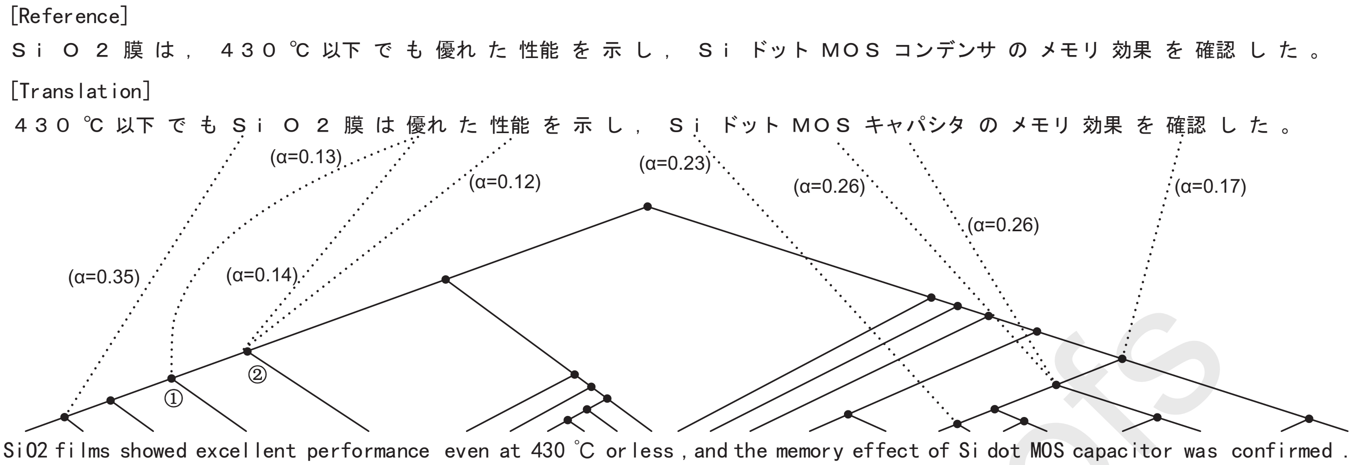

[Reference]
Ｓｉ Ｏ ２ 膜 は， ４３０ ℃ 以下 で も 優れ た 性能 を 示 し， Ｓｉ ドット ＭＯＳ コンデンサ の メモリ 効果 を 確認 し た 。
[Translation]
４３０ ℃ 以下 で も Ｓｉ Ｏ ２ 膜 は 優れ た 性能 を 示 し， Ｓｉ ドット ＭＯＳ キャパシタ の メモリ 効果 を 確認 し た 。
(α=0.13)
(α=0.12)
(α=0.23)
(α=0.26)
(α=0.26)
(α=0.17)
(α=0.35)
(α=0.14)
①
②
SiO2 films showed excellent performance even at 430 ℃ or less , and the memory effect of Si dot MOS capacitor was confirmed .

Fig. 5. Example of learned English tree structure and attention scores on ASPEC corpus.

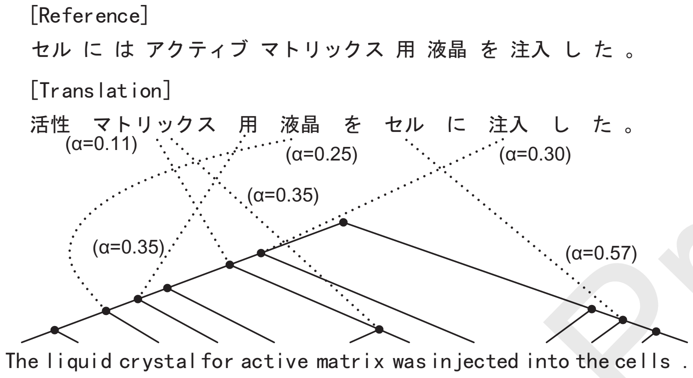

[Reference]
セル に は アクティブ マトリックス 用 液晶 を 注入 し た 。
[Translation]
活性 マトリックス 用 液晶 を セル に 注入 し た 。
(α=0.11)
(α=0.25)
(α=0.30)
(α=0.35)
(α=0.35)
(α=0.57)
The liquid crystal for active matrix was injected into the cells .

Fig. 6. Example of learned tree structure and attention scores on ASPEC corpus.

In Figure 6, we also find that there are some similarities between the learned and the Enju parsed trees. For example, both of them group the phrase "into the cells". But the Gumbel Tree-LSTM model often combines the first two words of a sentence, which is also presented in Williams et al. [39]. This may results in ungrammatical tree like "the liquid" rather than "liquid crystal" in Figure 6. It is hard for the Gumbel model to process prepositions. The prepositions sometimes form constituents with the noun phrases that follow them, as "into the cells" in Figure 6. But in other times, they forms constituents with the word that precede them, as "the liquid crystal for" in Figure 6.

5.2. Learned Chinese Trees

We also show three learned trees on the BOLT spoken corpus in Figure 7. At the left top of Figure 7, we can see that the model group the phrases "这个 价" (at this price) and "买 不 到" (can not buy). At the right top of Figure 7, the model combines the auxiliary word "了" (used after the verb or adj. to indicate completion) with its preceding adjective word "懒", and then connects the phrase with the following degree adverb "很多" (much more) to construct a new phrase "懒 了 很多" (become much more lazy). At the bottom of Figure 7, we also find some common phrases, such as "在 画" (be drawing) and "原理 图" (schematic diagram). It is noteworthy that the model often connect the initial two words although the left top of Figure 7 is not the case.

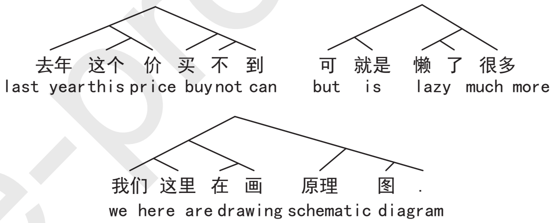

去年 这个 价 买 不 到
last year this price buy not can
可 就是 懒 了 很多
but is lazy much more
我们 这里 在 画 原理 图 .
we here are drawing schematic diagram

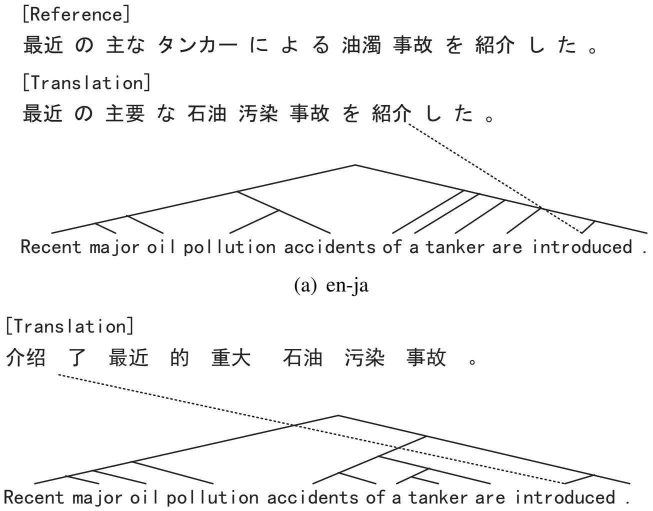
Fig. 7. Learned Chinese trees on BOLT corpora.

5.3. Learned Trees on Different Tasks

According to Choi et al. [20], each Gumbel Tree-LSTM model generates a distinct tree structure based on properties of the task. The proposed model may demonstrates different behavior features when training on different translation tasks, such as en-ja and en-zh. To test this, we used the FBIS corpus to train a small en-zh translation model, and compared the English tree structures it generated with the ones generated from the former en-ja model. Figure 8 shows an example. In Figure 8(a), the en-ja model group the words backward from tail of the sentence, while this is not so serious for the en-zh model in Figure 8(b). We hypothesize that the reason is that Chinese is an SVO language and the verb "介绍" (introduced) has to be reordered to the head of the sentence, while in Japanese this is not the case. Keeping the verb independent of the phrase "accidents of a tanker" may be helpful to complete long distance reordering.

5.4. Attention Scores on Learned Trees

We analyze two English-Japanese translation examples by our *GTree-att* model. The examples are also used by [8]. We show the attention score α, which means how much a Japanese word is related to English phrases.


[Reference]
最近 の 主な タンカー に よ る 油濁 事故 を 紹介 し た 。
[Translation]
最近 の 主要 な 石油 汚染 事故 を 紹介 し た 。
Recent major oil pollution accidents of a tanker are introduced .
[Translation]
介绍 了 最近 的 重大 石油 污染 事故 。
Recent major oil pollution accidents of a tanker are introduced .

(a) en-ja

(b) en-zh

Fig. 8. Tree structures built by models trained for en-ja and en-zh tasks

In Figure 5, two Japanese words “Ｓｉ” (silicon) have high attention scores with the English phrase “SiO2 films” (α = 0.35) and “of Si” (α = 0.23). The Japanese words “れ” (excellent) “性能” (performance) are softly aligned with the English phrases “SiO2 films showed excellent performance”. This is because the tree nodes ① and ② are very close to words “excellent” and “performance” in the tree.

In Figure 6, the Japanese word “液晶” (liquid crystal) mainly aligns with the English phrase “the liquid crystal” (α = 0.25). The Japanese word “用” (used for) mainly aligns with “the liquid crystal for” (α = 0.35), because that tree node is close to the English word “for” and previous context information may also be helpful to the alignment. The Japanese word “セル” (cell) aligns to the English phrase “the cells” (α = 0.57).

These examples illustrate the process that our models help to enhance proper alignments between target words and source phrases and thus improve translation performance.

6. Conclusion and Future Work

In this paper, we explore an approach to perform tree-to-sequence NMT with unsupervised trees. The tree structures are learned from the plain text rather than parsed by a tool. Thus the proposed method can compose different tree structures according to different tasks or training objectives and move around the poor parsing accuracy and expensive annotation cost.

Though we show that the proposed model gets better translation than the RNMT and Transformer models, it has room to be improved in the future. For example, since we utilize the source tree only, how to get the target trees in an unsupervised way and even utilize the both sides together are left to the future. Another remaining problem is that, the model learns to combine adjacent words only. If expanded to learn nonconsecutive phrases, such as “与 X_1 有 X_2” (“have X_2 with X_1”), it is expected to be more powerful.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant Nos. 61732005 and 61671064) and the National Key Research and Development Program of China (Grant No. 2017YFB1002103).

References

[1] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, 2014, pp. 1724–1734. URL: http://aclweb.org/anthology/D14-1179. doi:10.3115/v1/D14-1179.

[2] I. Sutskever, O. Vinyals, Q. V. Le, Sequence to sequence learning with neural networks, in: Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, K. Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada, 2014, pp. 3104–3112. URL: http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.

[3] J. L. Elman, Finding structure in time, Cognitive Science 14 (1990) 179–211.

[4] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Computation 9 (1997) 1735–1780.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, I. Polosukhin, Attention is all you need, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems 30, Curran Associates, Inc., 2017, pp. 5998–6008. URL: http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

[6] N. Chomsky, Three models for the description of language, IRE Transactions on Information Theory 2 (1956) 113–124.

[7] N. Chomsky, Aspects of the Theory of Syntax, The MIT press, Cambridge, 1965.

[8] A. Eriguchi, K. Hashimoto, Y. Tsuruoka, Tree-to-sequence attentional neural machine translation, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, 2016, pp. 823–833. URL: http://aclweb.org/anthology/P16-1078. doi:10.18653/v1/P16-1078.

[9] J. Li, D. Xiong, Z. Tu, M. Zhu, M. Zhang, G. Zhou, Modeling source syntax for neural machine translation, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 688–697. URL: https://www.aclweb.org/anthology/P17-1064. doi:10.18653/v1/P17-1064.

[10] K. Yamada, K. Knight, A syntax-based statistical translation model, in: Association for Computational Linguistic, 39th Annual Meeting and 10th Conference of the European Chapter, Proceedings of the Conference, July 9-11, 2001, Toulouse, France., Morgan Kaufmann Publishers, 2001, pp. 523–530. URL: http://www.aclweb.org/anthology/P01-1067. doi:10.3115/1073012.1073079.

[11] Y. Liu, Q. Liu, S. Lin, Tree-to-string alignment template for statistical machine translation, in: N. Calzolari, C. Cardie, P. Isabelle (Eds.), ACL 2006, 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, Sydney, Australia, 17-21 July 2006, The Association for Computer Linguistics, 2006. URL: http://aclweb.org/anthology/P06-1077. doi:10.3115/1220175.1220252.

[12] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, C. Potts, Recursive deep models for semantic compositionality over a sentiment treebank, in: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2013, pp. 1631–1642. URL: http://aclweb.org/anthology/D13-1170.

[13] X. Zhu, P. Sobhani, H. Guo, Long short-term memory over recursive structures, in: F. R. Bach, D. M. Blei (Eds.), Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France,

6-11 July 2015, volume 37 of *JMLR Workshop and Conference Proceedings*, JMLR.org, 2015, pp. 1604–1612. URL: http://proceedings.mlr.press/v37/zhub15.html.

[14] K. S. Tai, R. Socher, C. D. Manning, Improved semantic representations from tree-structured long short-term memory networks, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, 2015, pp. 1556–1566. URL: http://aclanthology.coli.uni-saarland.de/pdf/P/P15/P15-1150.pdf. doi:10.3115/v1/P15-1150.

[15] P. Chen, W. Ding, C. Bowes, D. Brown, A fully unsupervised word sense disambiguation method using dependency knowledge, in: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, Boulder, Colorado, 2009, pp. 28–36. URL: https://www.aclweb.org/anthology/N09-1004.

[16] D. S. Chaplot, P. Bhattacharyya, A. Paranjape, Unsupervised word sense disambiguation using markov random field and dependency parser, in: B. Bonet, S. Koenig (Eds.), Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA, AAAI Press, 2015, pp. 2217–2223. URL: http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9684.

[17] F. Meng, W. Lu, X. Zhang, C. Jinyong, Word sense disambiguation method based on HowNet and graph model, Journal of Qilu University of Technology 32 (2018) 66–73.

[18] W. Lu, H. Wu, P. Jian, Y. Huang, H. Huang, An empirical study of classifier combination based word sense disambiguation, IEICE Transactions 101-D (2018) 225–233.

[19] D. Chiang, A hierarchical phrase-based model for statistical machine translation, in: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05), Association for Computational Linguistics, 2005, pp. 263–270. URL: http://aclweb.org/anthology/P05-1033.

[20] J. Choi, K. M. Yoo, S. Lee, Learning to compose task-specific tree structures, in: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018, AAAI Press, 2018, pp. 5094–5101.

[21] H. Lu, Y. Li, M. Chen, H. Kim, S. Serikawa, Brain intelligence: Go beyond artificial intelligence, MONET 23 (2018) 368–375.

[22] H. Lu, Y. Li, T. Uemura, H. Kim, S. Serikawa, Low illumination underwater light field images reconstruction using deep convolutional neural networks, Future Gener. Comput. Syst. 82 (2018) 142–148.

[23] Y. Zhang, W. Lu, W. Ou, G. Zhang, X. Zhang, J. Cheng, W. Zhang, Chinese medical question answer selection via hybrid models based on CNN and GRU, Multimedia Tools and Applications (2019).

[24] X. Xu, H. Lu, J. Song, Y. Yang, H. T. Shen, X. Li, Ternary adversarial networks with self-supervision for zero-shot cross-modal retrieval, IEEE Transactions on Cybernetics (2019) 1–14.

[25] X. Xu, L. He, H. Lu, L. Gao, Y. Ji, Deep adversarial metric learning for cross-modal retrieval, World Wide Web 22 (2019) 657–672.

[26] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, L. D. Jackel, Backpropagation applied to handwritten zip code recognition, Neural Computation 1 (1989) 541–551.

[27] Q. Chen, X. Zhu, Z. Ling, S. Wei, H. Jiang, Enhancing and combining sequential and tree LSTM for natural language inference, CoRR abs/1609.06038 (2016).

[28] J. Li, T. Luong, D. Jurafsky, E. Hovy, When are tree structures necessary for deep learning of representations?, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2015, pp. 2304–2314. URL: http://aclanthology.coli.uni-saarland.de/pdf/D/D15/D15-1278.pdf. doi:10.18653/v1/D15-1278.

[29] L. Gao, X. Li, J. Song, H. T. Shen, Hierarchical LSTMs with adaptive attention for visual captioning, IEEE Transactions on Pattern Analysis and Machine Intelligence (2019) 1–1.

[30] H. Chen, S. Huang, D. Chiang, J. Chen, Improved neural machine translation with a syntax-aware encoder and decoder, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, 2017, pp. 1936–1945. URL: http://www.aclweb.org/anthology/P17-1177. doi:10.18653/v1/P17-1177.

[31] X. Zhang, L. Lu, M. Lapata, Top-down tree long short-term memory networks, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, 2016, pp. 310–320.

[32] G. Zhou, P. Luo, R. Cao, Y. Xiao, F. Lin, B. Chen, Q. He, Tree-structured neural machine for linguistics-aware sentence generation, in: AAAI, AAAI Press, 2018, pp. 5722–5729.

[33] D. Alvarez-Melis, T. S. Jaakkola, Tree-structured decoding with doubly-recurrent neural networks, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, OpenReview.net, 2017. URL: https://openreview.net/forum?id=HkYhZDqxg.

[34] X. Chen, C. Liu, D. Song, Tree-to-tree neural networks for program translation, in: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track Proceedings, OpenReview.net, 2018. URL: https://openreview.net/forum?id=H1xSvMh8M.

[35] E. Jang, S. Gu, B. Poole, Categorical reparameterization with gumbel-softmax, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, OpenReview.net, 2017. URL: https://openreview.net/forum?id=rkE3y85ee.

[36] D. Yogatama, P. Blunsom, C. Dyer, E. Grefenstette, W. Ling, Learning to compose words into sentences with reinforcement learning, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, OpenReview.net, 2017. URL: https://openreview.net/forum?id=Skvgqgqxe.

[37] R. J. Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning, Machine Learning 8 (1992) 229–256.

[38] H. Lu, Y. Li, S. Mu, D. Wang, H. Kim, S. Serikawa, Motor anomaly detection for unmanned aerial vehicles using reinforcement learning, IEEE Internet of Things Journal 5 (2018) 2315–2322.

[39] A. Williams, A. Drozdov, S. R. Bowman, Do latent tree learning models identify meaningful structure in sentences?, Transactions of the Association for Computational Linguistics 6 (2018) 253–267.

[40] Y. Shen, Z. Lin, C. Huang, A. C. Courville, Neural language modeling by jointly learning syntax and lexicon, in: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings, OpenReview.net, 2018. URL: https://openreview.net/forum?id=rkgOLb-0W.

[41] P. Le, W. Zuidema, The forest convolutional network: Compositional distributional semantics with a neural chart and without binarization, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Lisbon, Portugal, 2015, pp. 1155–1164. URL: https://www.aclweb.org/anthology/D15-1137. doi:10.18653/v1/D15-1137.

[42] Y. Kim, A. Rush, L. Yu, A. Kuncoro, C. Dyer, G. Melis, Unsupervised recurrent neural network grammars, 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), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 1105–1117. URL: https://www.aclweb.org/anthology/N19-1114. doi:10.18653/v1/N19-1114.

[43] C. Dyer, A. Kuncoro, M. Ballesteros, N. A. Smith, Recurrent neural network grammars, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, San Diego, California, 2016, pp. 199–209. URL: https://www.aclweb.org/anthology/N16-1024. doi:10.18653/v1/N16-1024.

[44] Y. Shen, S. Tan, A. Sordoni, A. C. Courville, Ordered neurons: Integrating tree structures into recurrent neural networks, in: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019, OpenReview.net, 2019. URL: https://openreview.net/forum?id=B1l6qiR5F7.

[45] Y. Bisk, K. Tran, Inducing grammars with and for neural machine translation, in: Proceedings of the 2nd Workshop on Neural Machine Translation and Generation, Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 25–35. URL: https://www.aclweb.org/anthology/W18-2704. doi:10.18653/v1/W18-2704.

[46] A. Drozdov, P. Verga, M. Yadav, M. Iyyer, A. McCallum, Unsu-

pervised latent tree induction with deep inside-outside recursive autoencoders, 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), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 1129–1141. URL: https://www.aclweb.org/anthology/N19-1116. doi:10.18653/v1/N19-1116.

[47] C. J. Maddison, A. Mnih, Y. W. Teh, The concrete distribution: A continuous relaxation of discrete random variables, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, OpenReview.net, 2017. URL: https://openreview.net/forum?id=S1jE5L5gl.

[48] T. Luong, H. Pham, C. D. Manning, Effective approaches to attention-based neural machine translation, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2015, pp. 1412–1421. URL: http://aclweb.org/anthology/D15-1166. doi:10.18653/v1/D15-1166.

[49] Z. Song, S. Strassel, H. Lee, K. Walker, J. Wright, J. Garland, D. Fore, B. Gainor, P. Cabe, T. Thomas, B. Callahan, A. Sawyer, Collecting natural SMS and chat conversations in multiple languages: The BOLT Phase 2 Corpus, in: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), European Language Resources Association (ELRA), Reykjavik, Iceland, 2014.

[50] G. Neubig, Y. Nakata, S. Mori, Pointwise prediction for robust, adaptable japanese morphological analysis, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, 2011, pp. 529–533. URL: http://aclweb.org/anthology/P11-2093.

[51] W. Che, Z. Li, T. Liu, LTP: A chinese language technology platform, in: Proceedings of the 23rd International Conference on Computational Linguistics: Demonstrations, Association for Computational Linguistics, 2010, pp. 13–16.

[52] X. Ma, Champollion: A robust parallel text sentence aligner, in: Proceedings of the Fifth International Conference on Language Resources and Evaluation, LREC 2006, Genoa, Italy, May 22-28, 2006., 2006, pp. 489–492.

[53] K. Papineni, S. Roukos, T. Ward, W. Zhu, BLEU: a method for automatic evaluation of machine translation, in: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA., ACL, 2002, pp. 311–318. URL: http://www.aclweb.org/anthology/P02-1040.pdf. doi:10.3115/1073083.1073135.

[54] H. Isozaki, T. Hirao, K. Duh, K. Sudoh, H. Tsukada, Automatic evaluation of translation quality for distant language pairs, in: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP '10, Association for Computational Linguistics, Stroudsburg, PA, USA, 2010, pp. 944–952. URL: http://dl.acm.org/citation.cfm?id=1870658.1870750.

[55] M. Denkowski, A. Lavie, Meteor Universal: Language specific translation evaluation for any target language, in: Proceedings of the Ninth Workshop on Statistical Machine Translation, WMT@ACL 2014, June 26-27, 2014, Baltimore, Maryland, USA, The Association for Computer Linguistics, 2014, pp. 376–380. URL: http://aclweb.org/anthology/W/W14/W14-3348.pdf.

[56] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: Y. Bengio, Y. LeCun (Eds.), 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015. URL: http://arxiv.org/abs/1412.6980.

[57] P. Koehn, Statistical significance tests for machine translation evaluation, in: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing , EMNLP 2004, A meeting of SIGDAT, a Special Interest Group of the ACL, held in conjunction with ACL 2004, 25-26 July 2004, Barcelona, Spain, ACL, 2004, pp. 388–395. URL: http://www.aclweb.org/anthology/W04-3250.

[58] J. Clark, D. Dyer, A. Lavie, N. Smith, Better hypothesis testing for statistical machine translation: Controlling for optimizer instability, in: The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA - Short Papers, The Association for Computer Linguistics, 2011, pp. 176–181. URL: http://www.aclweb.org/anthology/P11-2031.

[59] Y. Miyao, J. Tsujii, Feature forest models for probabilistic HPSG parsing, Computational Linguistics 34 (2008).

Declaration of interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

1. An unsupervised tree-to-sequence neural machine translation method is proposed.

2. Learning latent trees for improving neural machine translation.

3. Comparisons between Gumbel Tree-based model and the baselines, recurrent neural machine translation and self-attention based Transformer.

4. Analysis on the learned tree structures and attention scores on the structures.