Visualizing and Understanding Neural Machine Translation

Yanzhuo Ding[†] **Yang Liu**[†‡*] **Huanbo Luan**[†] **Maosong Sun**[†‡]
[†]State Key Laboratory of Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology
Department of Computer Science and Technology, Tsinghua University, Beijing, China
[‡]Jiangsu Collaborative Innovation Center for Language Competence, Jiangsu, China
djx133@yeah.net, liuyang2011@tsinghua.edu.cn
luanhuanbo@gmail.com, sms@tsinghua.edu.cn

Abstract

While neural machine translation (NMT) has made remarkable progress in recent years, it is hard to interpret its internal workings due to the continuous representations and non-linearity of neural networks. In this work, we propose to use layer-wise relevance propagation (LRP) to compute the contribution of each contextual word to arbitrary hidden states in the attention-based encoder-decoder framework. We show that visualization with LRP helps to interpret the internal workings of NMT and analyze translation errors.

1 Introduction

End-to-end neural machine translation (NMT), which leverages neural networks to directly map between natural languages, has gained increasing popularity recently (Sutskever et al., 2014; Bahdanau et al., 2015). NMT proves to outperform conventional statistical machine translation (SMT) significantly across a variety of language pairs (Junczys-Dowmunt et al., 2016) and becomes the new *de facto* method in practical MT systems (Wu et al., 2016).

However, there still remains a severe challenge: it is hard to interpret the internal workings of NMT. In SMT (Koehn et al., 2003; Chiang, 2005), the translation process can be denoted as a derivation that comprises a sequence of translation rules (e.g., phrase pairs and synchronous CFG rules). Defined on language structures with varying granularities, these translation rules are interpretable from a linguistic perspective. In contrast, NMT takes an end-to-end approach: all internal information is represented as real-valued vectors or matrices. It is challenging to associate hidden states in neural networks with interpretable language structures. As a result, the lack of interpretability makes it very difficult to understand translation process and debug NMT systems.

Therefore, it is important to develop new methods for visualizing and understanding NMT. Existing work on visualizing and interpreting neural models has been extensively investigated in computer vision (Krizhevsky et al., 2012; Mahendran and Vedaldi, 2015; Szegedy et al., 2014; Simonyan et al., 2014; Nguyen et al., 2015; Girshick et al., 2014; Bach et al., 2015). Although visualizing and interpreting neural models for natural language processing has started to attract attention recently (Karpathy et al., 2016; Li et al., 2016), to the best of our knowledge, there is no existing work on visualizing NMT models. Note that the attention mechanism (Bahdanau et al., 2015) is restricted to demonstrate the connection between words in source and target languages and unable to offer more insights in interpreting how target words are generated (see Section 4.5).

In this work, we propose to use layer-wise relevance propagation (LRP) (Bach et al., 2015) to visualize and interpret neural machine translation. Originally designed to compute the contributions of single pixels to predictions for image classifiers, LRP back-propagates relevance recursively from the output layer to the input layer. In contrast to visualization methods relying on derivatives, a major advantage of LRP is that it does not require neural activations to be differentiable or smooth (Bach et al., 2015). We adapt LRP to the attention-based encoder-decoder framework (Bahdanau et al., 2015) to calculate relevance that measures the association degree between two arbitrary neurons in neural networks. Case studies on Chinese-English translation show that visualization helps to interpret the internal workings of

*Corresponding author.

Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pages 1150–1159
Vancouver, Canada, July 30 - August 4, 2017.
https://doi.org/10.18653/v1/P17-1106

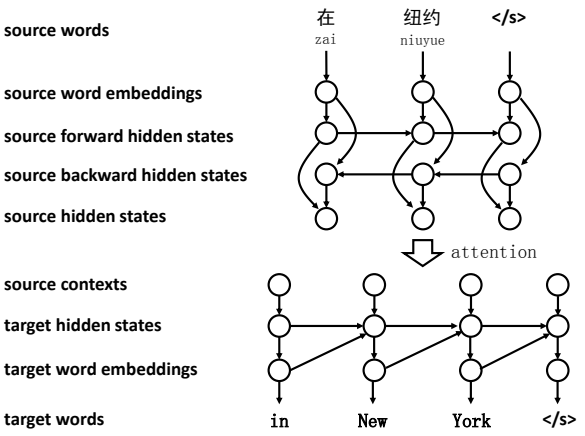

Figure 1: The attention-based encoder-decoder architecture for neural machine translation (Bahdanau et al., 2015).

NMT and analyze translation errors.

2 Background

Given a source sentence $\mathbf{x} = x_1, \ldots, x_i, \ldots, x_I$ with I source words and a target sentence $\mathbf{y} = y_1, \ldots, y_j, \ldots, y_J$ with J target words, neural machine translation (NMT) decomposes the sentence-level translation probability as a product of word-level translation probabilities:

$$P(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}) = \prod_{j=1}^{J} P(y_j|\mathbf{x}, \mathbf{y}_{<j};\boldsymbol{\theta}), \quad (1)$$

where $\mathbf{y}_{<j} = y_1, \ldots, y_{j-1}$ is a partial translation.

In this work, we focus on the attention-based encoder-decoder framework (Bahdanau et al., 2015). As shown in Figure 1, given a source sentence $\mathbf{x}$, the encoder first uses source word embeddings to map each source word x_i to a real-valued vector $\boldsymbol{x}_i$.[1]

Then, a forward recurrent neural network (RNN) with GRU units (Cho et al., 2014) runs to calculate source forward hidden states:

$$\overrightarrow{\mathbf{h}}_i = f(\overrightarrow{\mathbf{h}}_{i-1}, \boldsymbol{x}_i), \quad (2)$$

where $f(\cdot)$ is a non-linear function.

Similarly, the source backward hidden states can be obtained using a backward RNN:

$$\overleftarrow{\mathbf{h}}_i = f(\overleftarrow{\mathbf{h}}_{i+1}, \boldsymbol{x}_i). \quad (3)$$

[1]Note that we use $\mathbf{x}$ to denote a source sentence and $\boldsymbol{x}$ to denote the vector representation of a single source word.

To capture global contexts, the forward and backward hidden states are concatenated as the hidden state for each source word:

$$\mathbf{h}_i = [\overrightarrow{\mathbf{h}}_i; \overleftarrow{\mathbf{h}}_i]. \quad (4)$$

Bahdanau et al. (2015) propose an attention mechanism to dynamically determine the relevant source context $\mathbf{c}_j$ for each target word:

$$\mathbf{c}_j = \sum_{i=1}^{I+1} \alpha_{j,i}\mathbf{h}_i, \quad (5)$$

where $\alpha_{j,i}$ is an attention weight that indicates how well the source word x_i and the target word y_j match. Note that an end-of-sentence token is appended to the source sentence.

In the decoder, a target hidden state for the j-th target word is calculated as

$$\mathbf{s}_j = g(\mathbf{s}_{j-1}, \boldsymbol{y}_j, \mathbf{c}_j), \quad (6)$$

where $g(\cdot)$ is a non-linear function, $\boldsymbol{y}_{j-1}$ denotes the vector representation of the $(j-1)$-th target word.

Finally, the word-level translation probability is given by

$$P(y_j|\mathbf{x}, \mathbf{y}_{<j};\boldsymbol{\theta}) = \rho(\boldsymbol{y}_{j-1}, \mathbf{s}_j, \mathbf{c}_j), \quad (7)$$

where $\rho(\cdot)$ is a non-linear function.

Although NMT proves to deliver state-of-the-art translation performance with the capability to handle long-distance dependencies due to GRU and attention, it is hard to interpret the internal information such as $\overrightarrow{\mathbf{h}}_i$, $\overleftarrow{\mathbf{h}}_i$, $\mathbf{h}_i$, $\mathbf{c}_j$, and $\mathbf{s}_j$ in the encoder-decoder framework. Though projecting word embedding space into two dimensions (Faruqui and Dyer, 2014) and the attention matrix (Bahdanau et al., 2015) shed partial light on how NMT works, how to interpret the entire network still remains a challenge.

Therefore, it is important to develop new methods for understanding the translation process and analyzing translation errors for NMT.

3 Approach

3.1 Problem Statement

Recent efforts on interpreting and visualizing neural models has focused on calculating the contribution of a unit at the input layer to the final decision at the output layer (Simonyan et al., 2014; Mahendran and Vedaldi, 2015; Nguyen et al., 2015;

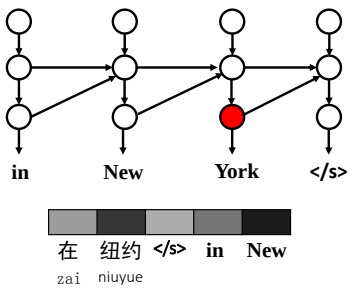

Figure 2: Visualizing the relevance between the vector representation of a target word "New York" and those of all source words and preceding target words.

Girshick et al., 2014; Bach et al., 2015; Li et al., 2016). For example, in image classification, it is important to understand the contribution of a single pixel to the prediction of classifier (Bach et al., 2015).

In this work, we are interested in calculating the contribution of source and target words to the following internal information in the attention-based encoder-decoder framework:

1. $\overrightarrow{\mathbf{h}}_i$: the i-th source forward hidden state,
2. $\overleftarrow{\mathbf{h}}_i$: the i-th source backward hidden state,
3. $\mathbf{h}_i$: the i-th source hidden state,
4. $\mathbf{c}_j$: the j-th source context vector,
5. $\mathbf{s}_j$: the j-th target hidden state,
6. $\boldsymbol{y}_j$: the j-th target word embedding.

For example, as shown in Figure 2, the generation of the third target word "York" depends on both the source context (i.e., the source sentence "zai niuyue </s>") and the target context (i.e., the partial translation "in New"). Intuitively, the source word "niuyue" and the target word "New" are more relevant to "York" and should receive higher relevance than other words. The problem is how to quantify and visualize the relevance between hidden states and contextual word vectors.

More formally, we introduce a number of definitions to facilitate the presentation.

Definition 1 *The* **contextual word set** *of a hidden state* $\mathbf{v} \in \mathbb{R}^{M\times 1}$ *is denoted as* $\mathcal{C}(\mathbf{v})$*, which is a set of source and target contextual word vectors* $\mathbf{u} \in \mathbb{R}^{N\times 1}$ *that influences the generation of* $\mathbf{v}$.

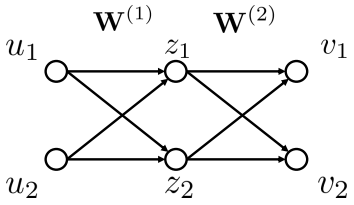

Figure 3: A simple feed-forward network for illustrating layer-wise relevance propagation (Bach et al., 2015).

For example, the context word set for $\overrightarrow{\mathbf{h}}_i$ is $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_i\}$, for $\overleftarrow{\mathbf{h}}_i$ is $\{\boldsymbol{x}_i, \ldots, \boldsymbol{x}_{I+1}\}$, and for $\mathbf{h}_i$ is $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{I+1}\}$. The contextual word set for $\mathbf{c}_j$ is $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{I+1}\}$, for $\mathbf{s}_j$ and $\boldsymbol{y}_j$ is $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{I+1}, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{j-1}\}$.

As both hidden states and contextual words are represented as real-valued vectors, we need to factorize vector-level relevance at the neuron level.

Definition 2 *The* **neuron-level relevance** *between the* m*-th neuron in a hidden state* $v_m \in \mathbb{R}$ *and the* n*-th neuron in a contextual word vector* $u_n \in \mathbb{R}$ *is denoted as* $r_{u_n \leftarrow v_m} \in \mathbb{R}$*, which satisfies the following constraint:*

$$v_m = \sum_{\mathbf{u}\in\mathcal{C}(\mathbf{v})} \sum_{n=1}^{N} r_{u_n \leftarrow v_m} \tag{8}$$

Definition 3 *The* **vector-level relevance** *between a hidden state* $\mathbf{v}$ *and one contextual word vector* $\mathbf{u} \in \mathcal{C}(\mathbf{v})$ *is denoted as* $R_{\mathbf{u}\leftarrow\mathbf{v}} \in \mathbb{R}$*, which quantifies the contribution of* $\mathbf{u}$ *to the generation of* $\mathbf{v}$*. It is calculated as*

$$R_{\mathbf{u}\leftarrow\mathbf{v}} = \sum_{m=1}^{M} \sum_{n=1}^{N} r_{u_n \leftarrow v_m} \tag{9}$$

Definition 4 *The* **relevance vector** *of a hidden state* $\mathbf{v}$ *is a sequence of vector-level relevance of its contextual words:*

$$R_{\mathbf{v}} = \{R_{\mathbf{u}_1\leftarrow\mathbf{v}}, \ldots, R_{\mathbf{u}_{|\mathcal{C}(\mathbf{v})|}\leftarrow\mathbf{v}}\} \tag{10}$$

Therefore, our goal is to compute relevance vectors for hidden states in a neural network, as shown in Figure 2. The key problem is how to compute neuron-level relevance.

3.2 Layer-wise Relevance Propagation

We follow (Bach et al., 2015) to use layer-wise relevance propagation (LRP) to compute neuron-level relevance. We use a simple feed-forward network shown in Figure 3 to illustrate the central idea of LRP.

Input: A neural network G for a sentence pair and a set of hidden states to be visualized $\mathcal{V}$.
Output: Vector-level relevance set $\mathcal{R}$.

for $u \in G$ *in a forward topological order* **do**
  for $v \in \mathrm{OUT}(u)$ **do**
    calculating weight ratios $w_{u \to v}$;
  end
end
for $\mathbf{v} \in \mathcal{V}$ **do**
  for $v \in \mathbf{v}$ **do**
    $r_{v \leftarrow v} = v$; `// initializing neuron-level relevance`
  end
  for $u \in G$ *in a backward topological order* **do**
    $r_{u \leftarrow v} = \sum_{z \in \mathrm{OUT}(u)} w_{u \to z} r_{z \leftarrow v}$; `// calculating neuron-level relevance`
  end
  for $\mathbf{u} \in \mathcal{C}(\mathbf{v})$ **do**
    $R_{\mathbf{u} \leftarrow \mathbf{v}} = \sum_{u \in \mathbf{u}} \sum_{v \in \mathbf{v}} r_{u \leftarrow v}$; `// calculating vector-level relevance`
    $\mathcal{R} = \mathcal{R} \cup \{R_{\mathbf{u} \leftarrow \mathbf{v}}\}$; `// Update vector-level relevance set`
  end
end

Algorithm 1: Layer-wise relevance propagation for neural machine translation.

LRP first propagates the relevance from the output layer to the intermediate layer:

$$r_{z_1 \leftarrow v_1} = \frac{\mathbf{W}^{(2)}_{1,1} z_1}{\mathbf{W}^{(2)}_{1,1} z_1 + \mathbf{W}^{(2)}_{2,1} z_2} v_1 \quad (11)$$

$$r_{z_2 \leftarrow v_1} = \frac{\mathbf{W}^{(2)}_{2,1} z_2}{\mathbf{W}^{(2)}_{1,1} z_1 + \mathbf{W}^{(2)}_{2,1} z_2} v_1 \quad (12)$$

Note that we ignore the non-linear activation function because Bach et al. (2015) indicate that LRP is *invariant* against the choice of non-linear function.

Then, the relevance is further propagated to the input layer:

$$\begin{aligned} r_{u_1 \leftarrow v_1} = & \frac{\mathbf{W}^{(1)}_{1,1} u_1}{\mathbf{W}^{(1)}_{1,1} u_1 + \mathbf{W}^{(1)}_{2,1} u_2} r_{z_1 \leftarrow v_1} + \\ & \frac{\mathbf{W}^{(1)}_{1,2} u_1}{\mathbf{W}^{(1)}_{1,2} u_1 + \mathbf{W}^{(1)}_{2,2} u_2} r_{z_2 \leftarrow v_1} \quad (13) \\ r_{u_2 \leftarrow v_1} = & \frac{\mathbf{W}^{(1)}_{2,1} u_2}{\mathbf{W}^{(1)}_{1,1} u_1 + \mathbf{W}^{(1)}_{2,1} u_2} r_{z_1 \leftarrow v_1} + \\ & \frac{\mathbf{W}^{(1)}_{2,2} u_2}{\mathbf{W}^{(1)}_{1,2} u_1 + \mathbf{W}^{(1)}_{2,2} u_2} r_{z_2 \leftarrow v_1} \quad (14) \end{aligned}$$

Note that $r_{u_1 \leftarrow v_1} + r_{u_2 \leftarrow v_1} = v_1$.

More formally, we introduce the following definitions to ease exposition.

Definition 5 *Given a neuron u, its* **incoming neuron set** $\mathrm{IN}(u)$ *comprises all its direct connected preceding neurons in the network.*

For example, in Figure 3, the incoming neuron set of z_1 is $\mathrm{IN}(z_1) = \{u_1, u_2\}$.

Definition 6 *Given a neuron u, its* **outcoming neuron set** $\mathrm{OUT}(u)$ *comprises all its direct connected descendant neurons in the network.*

For example, in Figure 3, the incoming neuron set of z_1 is $\mathrm{OUT}(z_1) = \{v_1, v_2\}$.

Definition 7 *Given a neuron v and its incoming neurons $u \in \mathrm{IN}(v)$, the* **weight ratio** *that measures the contribution of u to v is calculated as*

$$w_{u \to v} = \frac{\mathbf{W}_{u,v} u}{\sum_{u' \in \mathrm{IN}(v)} \mathbf{W}_{u',v} u'} \quad (15)$$

Although the NMT model usually involves multiple operators such as matrix multiplication, element-wise multiplication, and maximization, they only influence the way to calculate weight ratios in Eq. (15).

For matrix multiplication such as $\mathbf{v} = \mathbf{W}\mathbf{u}$, its basic form that is calculated at the neuron level is given by $v = \sum_{u \in \mathrm{IN}(v)} \mathbf{W}_{u,v} u$. We follow Bach et al. (2015) to calculate the weight ratio using Eq. (15).

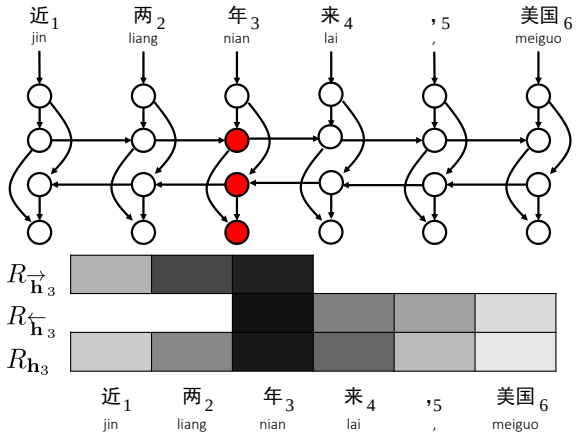

Figure 4: Visualizing source hidden states for a source content word "nian" (*years*).

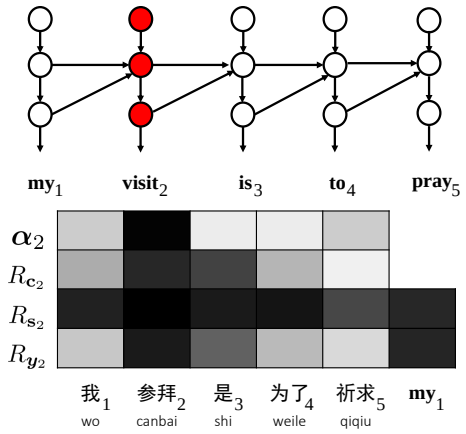

Figure 5: Visualizing target hidden states for a target content word "visit".

For element-wise multiplication such as $\mathbf{v} = \mathbf{u}_1 \circ \mathbf{u}_2$, its basic form is given by $v = \prod_{u \in \mathrm{IN}(v)} u$. We use the following method to calculate its weight ratio:

$$w_{u \to v} = \frac{u}{\sum_{u' \in \mathrm{IN}(v)} u'} \tag{16}$$

For maximization such as $v = \max\{u_1, u_2\}$, we calculate its weight ratio as follows:

$$w_{u \to v} = \begin{cases} 1 & \text{if } u = \max_{u' \in \mathrm{IN}(v)}\{u'\} \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

Therefore, the general local redistribution rule for LRP is given by

$$r_{u \leftarrow v} = \sum_{z \in \mathrm{OUT}(u)} w_{u \to z} r_{z \leftarrow v} \tag{18}$$

Algorithm 1 gives the layer-wise relevance propagation algorithm for neural machine translation. The input is an attention-based encoder-decoder neural network for a sentence pair after decoding G and a set of hidden states to be visualized $\mathcal{V}$. The output is a set of vector-level relevance between intended hidden states and their contextual words $\mathcal{R}$. The algorithm first computes weight ratios for each neuron in a forward pass (lines 1-4). Then, for each hidden state to be visualized (line 6), the algorithm initializes the neuron-level relevance for itself (lines 7-9). After initialization, the neuron-level relevance is back-propagated through the network (lines 10-12). Finally, vector-level relevance is calculated based on neuron-level relevance (lines 13-16). The time complexity of Algorithm 1 is $\mathcal{O}(|G| \times |\mathcal{V}| \times O_{\max})$, where $|G|$ is the number of neuron units in the neural network G, $|\mathcal{V}|$ is the number of hidden states to be visualized and $O_{\max}$ is the maximum of out-degree for neurons in the network. Calculating relevance is more computationally expensive than computing attention as it involves all neurons in the network. Fortunately, it is possible to take advantage of parallel architectures of GPUs and relevance caching for speed-up.

4 Analysis

4.1 Data Preparation

We evaluate our approach on Chinese-English translation. The training set consists of 1.25M pairs of sentences with 27.93M Chinese words and 34.51M English words. We use the NIST 2003 dataset as the development set for model selection and the NIST 2004 dataset as test set. The BLEU score on NIST 2003 is 32.73.

We use the open-source toolkit GROUNDHOG (Bahdanau et al., 2015), which implements the attention-based encoder-decoder framework. After model training and selection on the training and development sets, we use the resulting NMT model to translate the test set. Therefore, the visualization examples in the following subsections are taken from the test set.

4.2 Visualization of Hidden States

4.2.1 Source Side

Figure 4 visualizes the source hidden states for a source content word "nian" (*years*). For each word in the source string "jin liang nian lai , meiguo" (*in recent two years, USA*), we attach a number

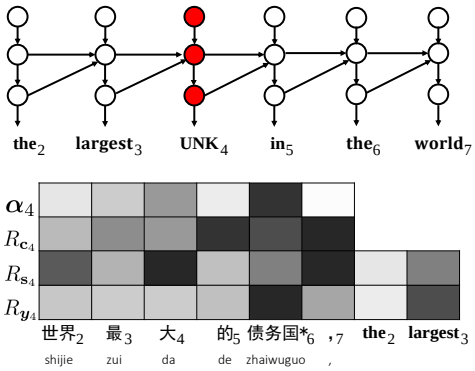

Figure 6: Visualizing target hidden states for a target UNK word.

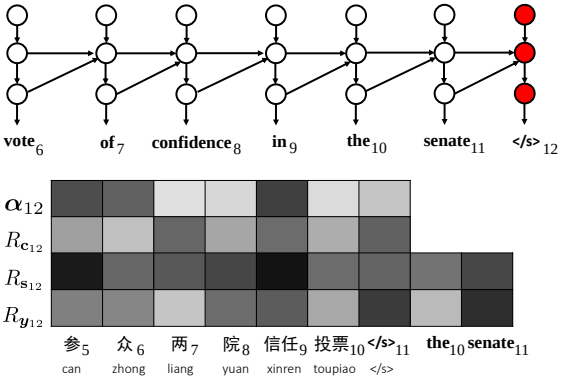

Figure 7: Analyzing translation error: word omission. The 6-th source word "zhong" is untranslated incorrectly.

to denote the position of the word in the sentence. For example, "nian" (*years*) is the third word.

We are interested in visualizing the relevance between the third source forward hidden state $\overrightarrow{\mathbf{h}}_3$ and all its contextual words "jin" (*recent*) and "liang" (*two*). We observe that the direct preceding word "liang" (*two*) contributes more to forming the forward hidden state of "nian" (*years*). For the third source backward hidden state $\overleftarrow{\mathbf{h}}_3$, the relevance of contextual words generally decreases with the increase of the distance to "nian" (*years*). Clearly, the concatenation of forward and backward hidden states $\mathbf{h}_3$ capture contexts in both directions.

The situations for function words and punctuation marks are similar but the relevance is usually more concentrated on the word itself. We omit the visualization due to space limit.

4.2.2 Target Side

Figure 5 visualizes the target-side hidden states for the second target word "visit". For comparison, we also give the attention weights $\boldsymbol{\alpha}_2$, which correctly identifies the second source word "canbai" ("visit") is most relevant to "visit".

The relevance vector of the source context $\mathbf{c}_2$ is generally consistent with the attention but reveals that the third word "shi" (*is*) also contributes to the generation of "visit".

For the target hidden state $\mathbf{s}_2$, the contextual word set includes the first target word "my". We find that most contextual words receive high values of relevance. This phenomenon has been frequently observed for most target words in other sentences. Note that relevance vector is not normalized. This is an essential difference between attention and relevance. While attention is defined to be normalized, the only constraint on relevance is that the sum of relevance of contextual words is identical to the value of intended hidden state neuron.

For the target word embedding $\mathbf{y}_2$, the relevance is generally consistent with the attention by identifying that the second source word contributes more to the generation of "visit". But $R_{\mathbf{y}_2}$ further indicates that the target word "my" is also very important for generating "visit".

Figure 6 shows the hidden states of a target UNK word, which is very common to see in NMT because of limited vocabulary. It is interesting to investigate whether the attention mechanism could put a UNK in the right place in the translation. In this example, the 6-th source word "zhaiwuguo" is a UNK. We find that the model successfully predicts the correct position of UNK by exploiting surrounding source and target contexts. But the ordering of UNK usually becomes worse if multiple UNK words exist on the source side.

4.3 Translation Error Analysis

Given the visualization of hidden states, it is possible to offer useful information for analyzing translation errors commonly observed in NMT such as word omission, word repetition, unrelated words and negation reversion.

4.3.1 Word Omission

Given a source sentence "bajisitan zongtong muxialafu yingde can zhong liang yuan xinren toupiao" (*pakistani president musharraf wins votes of confidence in senate and house*), the NMT model pro-

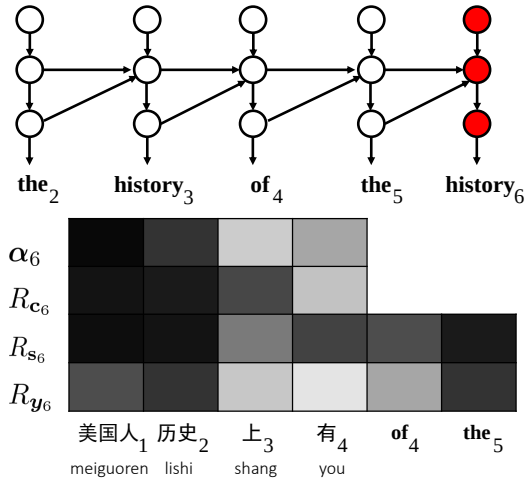

Figure 8: Analyzing translation error: word repetition. The target word "history" occurs twice in the translation incorrectly.

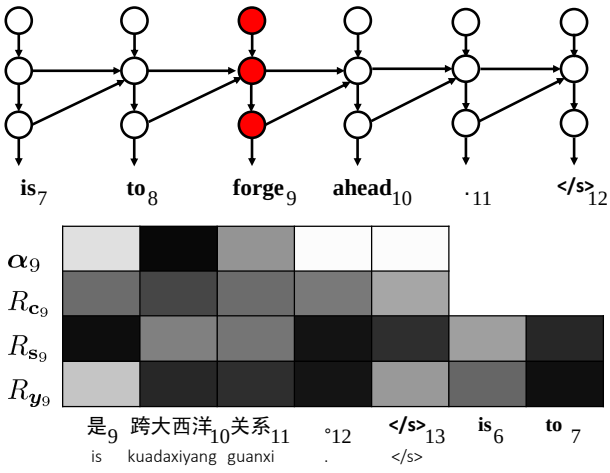

Figure 9: Analyzing translation error: unrelated words. The 9-th target word "forge" is totally unrelated to the source sentence.

duces a wrong translation "pakistani president win over democratic vote of confidence in the senate". One translation error is that the 6-th source word "zhong" (*house*) is incorrectly omitted for translation.

As the end-of-sentence token "</s>" occurs early than expected, we choose to visualize its corresponding target hidden states. Although the attention correctly identifies the 6-th source word "zhong" (*house*) to be important for generating the next target word, the relevance of source context $R_{\mathbf{c}_{12}}$ attaches more importance to the end-of-sentence token.

Finally, the relevance of target word $R_{\boldsymbol{y}_{12}}$ reveals that the end-of-sentence token and the 11-th target word "senate" become dominant in the softmax layer for generating the target word.

This example demonstrates that only using attention matrices does not suffice to analyze the internal workings of NMT. The values of relevance of contextual words might vary significantly across different layers.

4.3.2 Word Repetition

Given a source sentence "meiguoren lishi shang you jiang chengxi de chuantong , you fancuo rencuo de chuantong" (*in history , the people of america have the tradition of honesty and would not hesitate to admit their mistakes*), the NMT model produces a wrong translation "in the history of the history of the history of the americans , there is a tradition of faith in the history of mistakes". The translation error is that "history" repeats four times in the translation.

Figure 8 visualizes the target hidden states of the 6-th target word "history". According to the relevance of the target word embedding $R_{\boldsymbol{y}_6}$, the first source word "meiguoren" (*american*), the second source word "lishi" (*history*) and the 5-th target word "the" are most relevant to the generation of "history". Therefore, word repetition not only results from wrong attention but also is significantly influenced by target side context. This finding confirms the importance of controlling source and target contexts to improve fluency and adequacy (Tu et al., 2017).

4.3.3 Unrelated Words

Given a source sentence "ci ci huiyi de yi ge zhongyao yiti shi kuadaxiyang guanxi" (*one the the top agendas of the meeting is to discuss the cross-atlantic relations*), the model prediction is "a key topic of the meeting is to forge ahead". One translation error is that the 9-th English word "forge" is totally unrelated to the source sentence.

Figure 9 visualizes the hidden states of the 9-th target word "forge". We find that while the attention identifies the 10-th source word "kuadaxiyang" (*cross-atlantic*) to be most relevant, the relevance vector of the target word $R_{\boldsymbol{y}_9}$ finds that multiple source and target words should contribute to the generation of the next target word.

We observe that unrelated words are more likely to occur if multiple contextual words have high

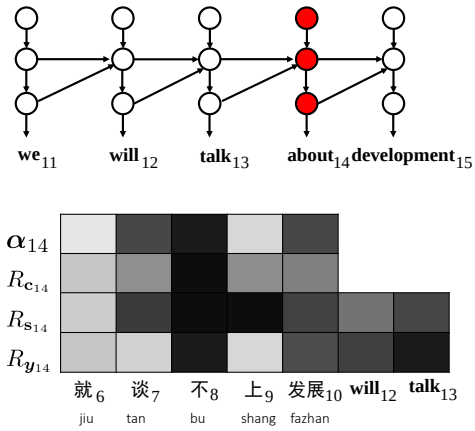

Figure 10: Analyzing translation error: negation. The 8-th negation source word "bu" (*not*) is not translated.

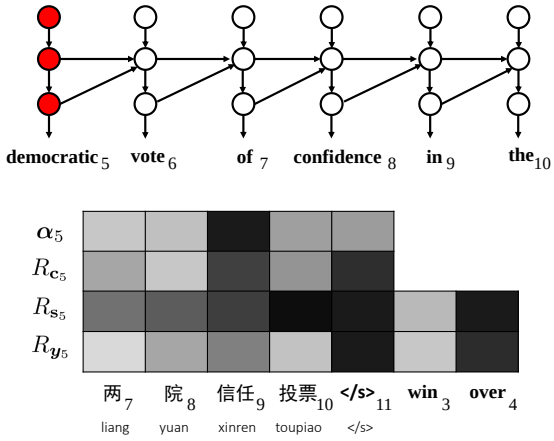

Figure 11: Analyzing translation error: extra word. The 5-th target word "democratic" is an extra word.

values in the relevance vector of the target word being generated.

4.3.4 Negation Reversion

Given a source sentence "bu jiejue shengcun wenti , jiu tan bu shang fa zhan , geng tan bu shang ke chixu fazhan" (*without solution to the issue of subsistence , there will be no development to speak of , let alone sustainable development*), the model prediction is "if we do not solve the problem of living , we will talk about development and still less can we talk about sustainable development". The translation error is that the 8-th negation source word "bu" (*not*) is untranslated. The omission of negation is a severe translation error it reverses the meaning of the source sentence.

As shown in Figure 10, while both attention and relevance correctly identify the 8-th negation word "bu" (*not*) to be most relevant, the model still generates "about" instead of a negation target word. One possible reason is that target context words "will talk" take the lead in determining the next target word.

4.4 Extra Words

Given a source sentence "bajisitan zongtong muxialafu yingde can zhong liang yuan xinren toupiao"(*pakistani president musharraf wins votes of confidence in senate and house*), the model prediction is "pakistani president win over democratic vote of confidence in the senate" The translation error is that the 5-th target word "democratic" is extra generated.

Figure 11 visualizes the hidden states of the 9-th target word "forge". We find that while the attention identifies the 9-th source word "xinren"(*confidence*) to be most relevant, the relevance vector of the target word R_{y_9} indicates that the end-of-sentence token and target words contribute more to the generation of "democratic".

4.5 Summary of Findings

We summarize the findings of visualizing and analyzing the decoding process of NMT as follows:

1. Although attention is very useful for understanding the connection between source and target words, only using attention is not sufficient for deep interpretation of target word generation (Figure 9);

2. The relevance of contextual words might vary significantly across different layers of hidden states (Figure 9);

3. Target-side context also plays a critical role in determining the next target word being generated. It is important to control both source and target contexts to produce correct translations (Figure 10);

4. Generating the end-of-sentence token too early might lead to many problems such as word omission, unrelated word generation, and truncated translation (Figures 7 and 9).

5 Related Work

Our work is closely related to previous visualization approaches that compute the contribution of a unit at the input layer to the final decision at the output layer (Simonyan et al., 2014; Mahendran and Vedaldi, 2015; Nguyen et al., 2015; Girshick et al., 2014; Bach et al., 2015; Li et al., 2016). Among them, our approach bears most resemblance to (Bach et al., 2015) since we adapt layer-wise relevance propagation to neural machine translation. The major difference is that word vectors rather than single pixels are the basic units in NMT. Therefore, we propose vector-level relevance based on neuron-level relevance for NMT. Calculating weight ratios has also been carefully designed for the operators in NMT.

The proposed approach also differs from (Li et al., 2016) in that we use relevance rather than partial derivative to quantify the contributions of contextual words. A major advantage of using relevance is that it does not require neural activations to be differentiable or smooth (Bach et al., 2015).

The relevance vector we used is significantly different from the attention matrix (Bahdanau et al., 2015). While attention only demonstrates the association degree between source and target words, relevance can be used to calculate the association degree between two arbitrary neurons in neural networks. In addition, relevance is effective in analyzing the effect of source and target contexts on generating target words.

6 Conclusion

In this work, we propose to use layer-wise relevance propagation to visualize and interpret neural machine translation. Our approach is capable of calculating the relevance between arbitrary hidden states and contextual words by back-propagating relevance along the network recursively. Analyses of the state-of-art attention-based encoder-decoder framework on Chinese-English translation show that our approach is able to offer more insights than the attention mechanism for interpreting neural machine translation.

In the future, we plan to apply our approach to more NMT approaches (Sutskever et al., 2014; Shen et al., 2016; Tu et al., 2016; Wu et al., 2016) on more language pairs to further verify its effectiveness. It is also interesting to develop relevance-based neural translation models to explicitly control relevance to produce better translations.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No.61522204), the 863 Program (2015AA015407), and the National Natural Science Foundation of China (No.61432013). This research is also supported by the Singapore National Research Foundation under its International Research Centre@Singapore Funding Initiative and administered by the IDM Programme.

References

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE* .

Dzmitry Bahdanau, KyungHyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of ICLR*.

Davie Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of ACL*.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of EMNLP*.

Mannal Faruqui and Chris Dyer. 2014. Improving vector space word representations using multilingual correlation. In *Proceedings of EACL*.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of CVPR*.

Marcin Junczys-Dowmunt, Tomasz Dwojak, and Hieu Hoang. 2016. Is neural machine translation ready for deployment? a case study on 30 translation directions. arXiv:1610.01108v2.

Andrej Karpathy, Justin Johnson, and Fei-Fei Li. 2016. Visualing and understanding recurrent networks. In *Proceedings of ICLR Workshop*.

Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of NAACL*.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. 2012. Imagenet classification with deep convolutional nerual networks. In *Proceedings of NIPS*.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016. Visualizing and understanding neural models in nlp. In *Proceedings of NAACL*.

Aravindh Mahendran and Andrea Vedaldi. 2015. Understanding deep image representations by inverting them. In *Proceedings of CVPR*.

Anh Nguyen, Jason Yosinski, and Jeff Clune. 2015. Deep neural networks are easily fooled: High confidence predictions for unrecignizable images. In *Proceedings of CVPR*.

Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Minimum risk training for neural machine translation. In *Proceedings of ACL*.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: Visualizing image classification models and saliency maps. In *Proceedings of ICLR Workshop*.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of NIPS*.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *Proceedings of ICLR*.

Zhaopeng Tu, Yang Liu, Lifeng Shang, Xiaohua Liu, and Hang Li. 2017. Context gates for neural machine translation. *Transactions of the ACL* .

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. In *Proceedings of ACL*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv:1609.08144v2.