

## A Novel Beam Search to Improve Neural Machine Translation for English-Chinese

Xinyue Lin<sup>1</sup>, Jin Liu<sup>1,\*</sup>, Jianming Zhang<sup>2</sup> and Se-Jung Lim<sup>3</sup>

**Abstract:** Neural Machine Translation (NMT) is an end-to-end learning approach for automated translation, overcoming the weaknesses of conventional phrase-based translation systems. Although NMT based systems have gained their popularity in commercial translation applications, there is still plenty of room for improvement. Being the most popular search algorithm in NMT, beam search is vital to the translation result. However, traditional beam search can produce duplicate or missing translation due to its target sequence selection strategy. Aiming to alleviate this problem, this paper proposed neural machine translation improvements based on a novel beam search evaluation function. And we use reinforcement learning to train a translation evaluation system to select better candidate words for generating translations. In the experiments, we conducted extensive experiments to evaluate our methods. CASIA corpus and the 1,000,000 pairs of bilingual corpora of NiuTrans are used in our experiments. The experiment results prove that the proposed methods can effectively improve the English to Chinese translation quality.

**Keywords:** Neural machine translation, beam search, reinforcement learning.

### 1 Introduction

Natural language processing is a comprehensive interdisciplinary subject integrating linguistics, mathematics, computer science and cognitive science. Machine Translation is a flagship of the recent successes and advances in the field of natural language processing. Its practical applications and use as a testbed for sequence transduction algorithms have spurred renewed interest in this topic.

Neural machine translation (NMT) has achieved state-of-the-art translation performances [Kalchbrenner and Blunsom (2013); Sutskever, Vinyals and Le (2014); Bahdanau, Cho and Bengio (2014)] in last several years. Furthermore, many recent studies have shown that neural networks [Zhang, Jin, Sun et al. (2018); Zhang, Wang, Lu et al. (2019)] can be successfully applied to many tasks in natural language processing (NLP). These include,

---

<sup>1</sup> College of Information Engineering, Shanghai Maritime University, Shanghai, 201306, China.

<sup>2</sup> School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha, 410114, China.

<sup>3</sup> Liberal Arts & Convergence Studies, Honam University, Gwangju, 62399, Korea.

\* Corresponding Author: Jin Liu. Email: jinliu@shmtu.edu.cn.

Received: 12 April 2020; Accepted: 12 May 2020.

but are not limited to, rule-based machine translation, memory-based translation, mechanic translation by analogy principle, language modeling, paraphrase detection and word embedding extraction [Mikolov, Corrado, Chen et al. (2013)], In the field of statistical machine translation (SMT), deep neural networks [Zhang, Xie, Sun et al. (2020)] have begun to show promising results [Schwenk, Udani, Gupta et al. (2012)], Schwenk et al. summarize the successful application of feedforward neural networks in the framework of phrase-based SMT (Statistical Machine Translation) systems. In 2016, instead of using SMT, Google used neural network to translate some languages, which introduced neural network machine translation from academia to the industry. Their solution requires no change in the model architecture from their base system but instead introduces an artificial token at the beginning of the input sentence to specify the required target language.

Based on the existing translation model architecture and summarized their advantages and disadvantages, we concluded that:

- Although there are many translation tools available to us today, these machine translations usually do not produce satisfactory results.
- While recent advances have reported near human-level performance on several language pairs using neural approaches, it still has some shortcomings. For example, the result of translation may duplicate content and some of the source statement is missing.

To solve these problems, we added two penalties in beam search to make it selects better translation, which are based on duplicate detection and length ratio respectively. Furthermore, we use reinforcement learning to train a translation evaluation system to select better candidate words for the translation result.

In summary our contributions are:

1. An innovative beam search evaluation function was used to select the better translation, which improved with duplicate detection and length ratio.
2. A novel reinforcement learning based method was used to build a selection model of translations. This method can make the translation evaluation system select better candidate words for generating translations.

The remaining of this paper is organized as the follows. Section 2 presents the related researches work of neural machine translation and beam search method, Section 3 the proposed method is illustrated in detail. Section 4 gives a detailed description of the experiments, analyzes the experiment results. And Section 5 draws some conclusion.

## **2 Related work**

In this section, we discuss and compare common model architecture of machine translation system and application of language model in machine translation, focusing on heuristic search algorithm for translation results.

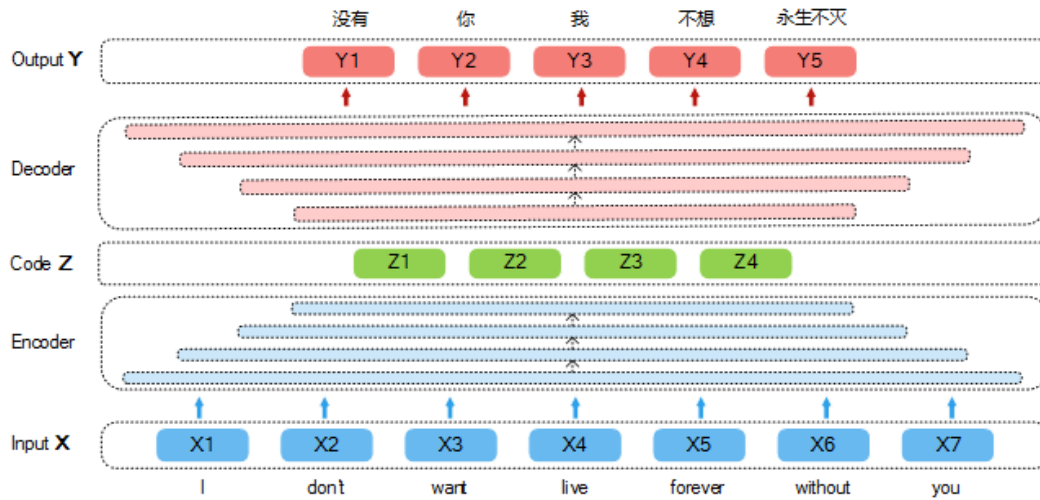
### ***2.1 Neural machine translation***

The neural machine translation system usually can be implemented as an encoder-decoder network with recurrent neural networks.

The encoder is a bidirectional neural network with gated recurrent units that reads an

input sequence  $x=(x_1, \dots, x_m)$  and calculates a forward sequence of hidden states  $(\vec{h}_1, \dots, \vec{h}_m)$ , and a backward sequence  $(\overleftarrow{h}_1, \dots, \overleftarrow{h}_m)$ . The hidden states  $\vec{h}_j$  and  $\overleftarrow{h}_j$  are concatenated to obtain the annotation vector  $h_j$ . The decoder is a recurrent neural network that predicts a target sequence  $y=(y_1, \dots, y_m)$ .

Fig. 1 shows the structure of a typical English-Chinese translation system, which consists of encoder-decoder network. The input is an English sentence, which is compiled into a vector  $z=(z_1, \dots, z_m)$  by the encoder, and then the vector  $z$  is decoded into a Chinese syntactic sentence by the decoder.



**Figure 1:** Structure of English-Chinese translation system

Machine translation for end-to-end learning was first proposed by Chrisman [Chrisman (1991); Kalchbrenner and Blunsom (2013)] mapped input sentences into vectors of fixed dimensions for the first time. Based on this, Cho et al. [Cho, Merrienboer, Gülçehre et al. (2014)] further proposed the RNN encoder-decoder model, which improved the performance of the machine translation system.

Later, Sutskever et al. [Sutskever, Vinyals, Le et al. (2014)] proposed a new end-to-end sequence learning method using a model combining two LSTM (Long short-term memory) networks. The decoder uses beam search to find the best translation result.

Based on Graves A’s LSTM equation, Sutskever et al. [Sutskever, Udani, Gupta et al. (2014)] proposed an improved model which is composed of two different LSTM networks, one of which takes reversed input sentence, to produce a “minimum time delay”. Unlike feature-based traditional approaches and long short-term memory network based models, Zeng’s work, Zeng et al. [Zeng, Dai, Li et al. (2019)], combines the strengths of linguistic resources and gating mechanism to propose an effective convolutional neural network based model for aspect-based sentiment analysis.

Bahdanau et al. [Bahdanau, Cho and Bengio (2014)] conjecture that base on encoder-decoder architecture, the use of a fixed-length vector is a bottleneck in improving the performance, and propose to extend this by allowing a model to automatically (soft-)

search for parts of a source sentence which are relevant to predicting a target word, without having to form these parts into hard segments explicitly. The attentional mechanism was improved by Luong et al. [Luong, Pham, Manning et al. (2015)], The authors examine two effective kinds of attention mechanism. The former approach is similar to Bahdanau's method but simpler, and the latter approach can be seen as an interesting mixture between hard attention and soft attention. Similar method can be also applied in other tasks such as question classification, Liu et al. [Liu, Yang, Lv et al. (2019)] proposed an attention-based encoder-decoder model that can extract the features of Chinese questions effectively. Furthermore, Kaiser et al. [Kaiser, Gomez, Chollet et al. (2017)] applied depthwise separable convolutions [Chollet (2017)] to neural machine translation. Lample et al. [Lample, Ott, Conneau et al. (2018)] propose two model variants, a neural and a phrase-based model.

Similarly, Generative Adversarial Networks (GANs) is also used to do distant supervised relation extraction by Zeng et al. [Zeng, Dai, Li et al. (2018); Moryossef, Aharoni, Goldberg et al. (2019)] propose a black-box approach for injecting the missing information to a pre-trained neural machine translation system, allowing to control the morphological variations in the generated translations without changing the underlying model or training data.

## 2.2 Beam search

Here we briefly review neural text generation and then review existing beam search algorithms.

Beam search is a widely used approximate search strategy for neural network decoders, and it generally outperforms simple greedy decoding on tasks like machine translation. Beam search is the optimization of the best priority search, which can reduce its memory requirements.

Assume the input is embedded into a vector  $x$ , from which we generate the output sentence  $y$  which is a completed hypothesis:

$$y^* = \underset{y:comp(y)}{\operatorname{argmax}} p(y|x) = \underset{y:comp(y)}{\operatorname{argmax}} \prod_{i \leq |y|} p(y_i|x, y_{<i}) \quad (1)$$

where  $y_{<i}$  is a popular shorthand notation for the prefix  $y_0, y_1, \dots, y_{i-1}$ . We say that a hypothesis  $y$  is completed, notated as  $comp(y)$ , if its last word is  $\langle /s \rangle$ , i.e.,

$$comp(y) \triangleq (y_{|y|} = \langle /s \rangle) \quad (2)$$

in which case it will not be further expanded.

Wiseman et al. [Wiseman and Rush (2016)] introduce a model and beam-search training scheme, that extends seq2seq to learn global sequence scores.

Furthermore, Google's Neural Machine Translation system (GNMT) [Wu, Schuster, Chen et al. (2016)] uses beam search technique which employs a length-normalization procedure and uses a coverage penalty, which encourages generation of an output sentence that is most likely to cover all the words in the source sentence.

Sennrich et al. [Sennrich, Haddow and Birch (2016)] introduces a simpler and more effective approach, making the NMT model capable of open-vocabulary translation by encoding rare and unknown words as sequences of sub word units. Furthermore, part-of-

speech (POS) feature is used to improve the language model used in translation. Liu et al. [Liu, Lin, Ren et al. (2018)] proposed a negative sampling algorithm based on POS tagging, which can optimize the negative sampling process and improve the quality of the final language model. In view of the long execution time and low execution efficiency of Support Vector Machine in large-scale training samples, Chen et al. [Chen, Xiong, Xu et al. (2019)] proposed the online incremental and decremental learning algorithm based on variable support vector machine (VSVM). Williams et al. [Williams and Aleksic (2017)] present a technique to adapt the beam search algorithm to preserve hypotheses when they may benefit from rescoring. This technique makes it feasible to use one base language model, but still achieve high-accuracy speech recognition results in all contexts.

### 3 Method

In this section, aiming at the two problems that often occur in NMT, two corresponding methods to improve the sequence to sequence machine translation model (seq2seq) are introduced in detail. We propose a neural machine translation improvement based on beam search, firstly. Secondly, reinforcement learning was used in machine translation to train a translation evaluation system to select better candidate words of the model.

#### **3.1 Improved beam search with duplicate detection and length ratio control**

##### *3.1.1 Issues in traditional beam search algorithm*

There are two problems that often appear in neural machine translation: 1. Traditional machine translation model always selects the word with the highest probability at the current time step, which is not necessarily the best choice for the whole translation. So, in the translation model, due to duplicate translation of some fragments in the source statement, one fragment may be translated twice or more. 2. Beam Search prefers short results over long ones, that caused the translation model may miss part of the source statement. As show in the following example:

**Source statement:** *There are people living in squatter huts, temporary housing areas, “cage” apartments or private rental housing for which they pay painfully expensive rents; people whose names are on the public housing Waiting List; and people who want to buy Home Ownership Scheme flats but who are disappointed every time because they fail to get drawn in the balloting. These kinds of people, too, are likely to ask: How much of the money in the Budget will be spent on improving my life?*

**Target language standard translation:** *居住于木屋、临屋、笼屋和在私人楼宇捱贵租，正在轮候公屋，或每次在居屋抽签中都不能中签，失望而回的居民同样会问，今年的财政预算案中，有多少金钱用於改善他们的生活呢？*

**Translation generated by Translation model:** *他们在寮屋、临屋、笼屋或私人租住公屋单位、临屋或私人租住公屋单位，他们有多少时间可以用来改善我的生活？*

Both types of problems have appeared in the generated test by the translation model. Among them, “临屋” and “或私人租住公屋单位” have been translated twice, and a section “正在轮候公屋，或每次在居屋抽签中都不能中签，失望而回的居民同样会问，今年的财政预算案中” in the middle of the standard translation of the target

language is completely omitted. These situations decrease the quality and readability of the generated translation.

As we described above, beam search prefers short results over long ones. So, the appearance of duplicate translation will increase the possibility of missing words.

We add two penalty terms to the beam search evaluation function. The penalty items for duplicate translations are added to alleviate the first problem, and the penalty items based on length ratios are added to alleviate the second one.

### 3.1.2 Beam search with duplicate detection and length ratio control

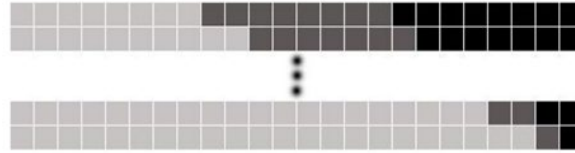
The original evaluation function of the beam search is as follows:

$$s(Y, X) = \log(P(Y|X)) \quad (3)$$

where,  $P(Y|X)$  represents the probability that  $Y$  will appear under the condition that  $X$  appears. In this paper, we modified the beam search evaluation function, added a penalty item  $d(x)$  based on the duplicate detection and a penalty term based on the length ratio  $l(x)$ :

$$s(Y, X) = \log(P(Y|X)) + d(x) + l(x) \quad (4)$$

For the problem of duplicate translation, we propose the penalty  $d(x)$  that based on duplicate detection. We obtained the similarity by comparing a range of translations. The specific comparison method is as shown in Fig. 2:



**Figure 2:** The selection of the translation fragment that has been generated

We compared adjacent fragment of different lengths one by one. That is, shows in Fig. 2, the black part is compared with the dark gray part to find the same part. If the same item appears in the translation fragment, then we multiply the number of identical words by the corresponding penalty factor. Finally, we used weighted summation to get the penalty for duplicate translation. The equation of  $d(x)$  is as Eq. (5):

$$d(x) = \sum_{i=1}^{\delta} \frac{\varepsilon}{i} \sum_{j=0}^i (|y_{c-j} - y_{c-i-j}| == |0|) \quad (5)$$

where,  $c$  is the index of the current translated word,  $\delta$  is the range of duplicate detection, and  $\varepsilon$  is the coefficient of the penalty term. It can be seen from the Eq. (5) that the closer the duplicate is, the larger the corresponding coefficient is, and the corresponding penalty is larger. In Eq. (5),  $\delta$  is the range between two word in the translation from 1 to  $\delta$ . And for each range, we count the number of times that every two word in this range is equal, then multiply it by the penalty coefficient. Finally, we add the weighted terms to the evaluation function of the beam search.

To avoid missing words in translation, we use the current candidate word, cumulative length distribution function  $F_X(x)$ , and the penalty factor as the input of the algorithm to get the penalty for Length Ratio. To get a proper range of sentence length ratios, firstly we count the length of 1,000,000 pairs of parallel corpus translated by NiuTrans

[NiuTrans (2011)], secondly we divide the length of the target language sentence (Chinese) into different sets by the length of the corresponding source language sentence (English), which obtain a series of ratios, then remove the data which have obvious errors, for example, the ratio is too small and too large. After excluding these extreme data, we use linear regression to fit the cumulative distribution function of the remaining data to obtain  $F_X(x)$  which is the probability that the translation has ended:

$$F_X(x) = P(X < x) \quad (6)$$

here  $x$  can be defined as:  $x = \frac{\text{target\_length}}{\text{source\_length}}$ , which is the ratio of the length of the target language sentence to the length of the source language sentence. When the *EOS* (Sentence Mark) and the ordinary word appear simultaneously in the candidate words of the beam search,  $F_X(x)$  and  $1-F_X(x)$  will add to their evaluation functions separately.  $F_X(x)$  is the probability that the translation has ended and  $1 - F_X(x)$  is the probability that the translation has not ended.

$$l(x) = \begin{cases} \theta F_X(x), & \text{not\_EOS} \\ \theta(1 - F_X(x)), & \text{EOS} \end{cases} \quad (7)$$

In Eq. (7), when candidate word is *EOS*, the untranslated probability is multiplied by the penalty factor  $\theta$  as a penalty term. And when the candidate word is not *EOS*, the untranslated probability is multiplied by a penalty factor  $\theta$  as a penalty term. We can get different scores through penalty terms, thus beam search can make better choices by choosing different  $l(x)$ .

### 3.1.3 Algorithm description

---

#### **Algorithm I: Determining Penalty Items for Duplicate Detection**

---

**Input:** *mat*, *delta*, *epsilon*.

*mat* is the matrix of translation.

*delta* is the range of duplicate detection.

*epsilon* is the coefficient of penalty.

**Output:** *Score*

1. *Scores* = *tensor(mat.rows)*
  2. *Zeros* = *tensor(mat.rows)*
  3. *Scores.fill(0)*
  4. *Zeros.fill(0)*
  5. **while**  $i \geq 1 \& i \leq \text{delta}$  **do**
  6.  $\text{tmp\_scores} = \text{sum} \left( \text{abs} \left( \begin{matrix} \text{mat}[c-i-1:c] - \\ \text{mat}[c-2i-1:c-i] \end{matrix} \right) \right) == \text{Zeros}$
  7.  $\text{Scores} += \left( \frac{\text{epsilon}}{i} \right) * \text{tmp\_scores}$
  8. **return** *Scores*
  9. **end while**
-

Based on the construction principle of the duplicate detection penalty items that were described in Section 3.1.3, the computation process is given in Algorithm I. We take the candidate words of the entire beam search, the parameters  $\delta$  and  $\varepsilon$  as the input to the algorithm. After dividing into multiple fragments of different sizes to compare, we calculated the penalty terms respectively, and finally sum with weight together.

---

**Algorithm II: Determining Penalty Items for Length Ratio Control**

---

**Input:** *words, fx, theta.*

*word* is the matrix of the candidate words.

*fx* is the value of the cumulative distribution function obtained by calculation.

*theta* is the parameters in the equation.

**Output:** *result*

1. *Scores=tensor(words.length)*
  2. *eos=tensor(words.length)*
  3. *Scores.fill(0)*
  4. *eos.fill(0)*
  5.  $is\_eos = abs(mat == eos)$
  6.  $result = theta * (1 - fx) * is\_eos + theta * (1 - is\_eos)$
- 

Based on the construction principle of the length ratio control penalty items described in Section 3.1.3, the computation process is given in Algorithm II. We take the current candidate word, the cumulative distribution function calculated by the current length, and the parameter  $\theta$  as the input to the computation. First, we use the vector operation to get whether the candidate word is *EOS*, if it is *EOS*, the value is 1, otherwise it is 0. Then by the form of dot multiplication, we got the value of  $l(x)$  in the equation.

### 3.2 Reinforcement learning based translation evaluation

The quality of translations generated by machine translation is generally evaluated by BLEU scores, which compares the similarity between generated translation and target translation. Because this evaluation has to wait until the translation is fully generated, so it has some limitations, for example, it cannot improve the accuracy performance during the translation process. If the BLEU score could be used in the process of beam search generating translation, better candidate words can be selected, and the quality of the generated translation will be improved accordingly.

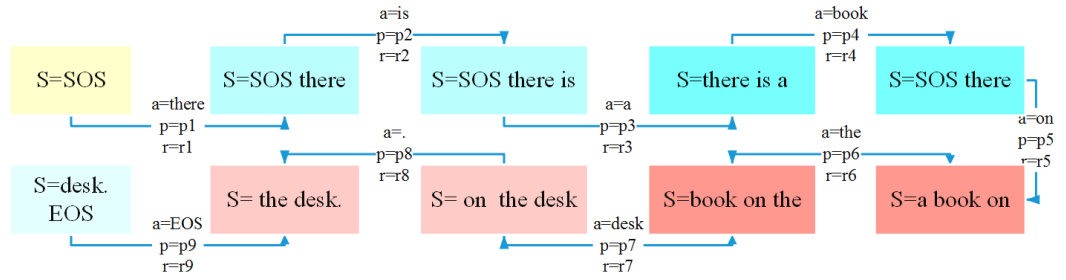
#### 3.2.1 Model description

To improve the translation text evaluation system, making it choose better candidate words. this paper use reinforcement learning based method to train an evaluation system. This method can make use of the maximize long-term cumulative rewards to improve the generated text during the target generation process. We first use the generated text by the



model and the corresponding target translation as corpus, then put the corpus through the  $Q$  learning algorithm. The  $Q$  learning algorithm evaluates the original strategy, executes the  $\epsilon$ -greedy strategy, and  $Q$  table is used to record the state in the algorithm. First, we postulate that the beginning of each sentence is the Start of Sentence (*SOS*) that is the initial state of each sentence (state). Then, the first word after the initial state is taken as a selectable action. After selecting an action, the current state is switched to the selected action (the selected word), and each sentence in the corpus represents a choice of route. As show in Fig. 3 below. The equation for the reward  $r$  is as follows:

$$r = \sum_i r_i \tag{8}$$



**Figure 3:** State transition diagram

The reward score in the reward matrix  $R$  is defined as: for the route generated by the translation model (the translation), the reward uses the negative score, and the target translation uses the positive score. Since the length of each sentence is different, the reward for each word is the sentence score divided by the length of the sentence. After defining the reward matrix  $R$ , the  $Q$  table can be obtained by the  $Q$  learning algorithm. The values of  $Q$  table are updated as follow:

$$Q(s, a) = R(s, a) + \gamma \cdot \max_{\tilde{a}} \{Q(\tilde{s}, \tilde{a})\} \tag{9}$$

Among them,  $s$  and  $a$  respectively represent the current state and behavior,  $\tilde{s}$ ,  $\tilde{a}$  respectively represent the next state and behavior of  $s$ , and  $\gamma$  is the value of the learning parameter which is between  $[0, 1]$ . Through Eq. (9), we can get the score for each route, that is, the score for each sentence.

### 3.2.2 Algorithm description

---

**Algorithm III: Use Q learning to find the table Q Based on the Reward matrix**

**Input:**  $\gamma$ , reward, epoch, end

- $\gamma$  is learning parameters.
- reward is the matrix of reward scores.
- epoch is the length of the sentence.
- end is the end mark.

**Output:**  $q$

**Begin**

---

---

```

1.  $q = \text{reward.keys}()$ 
2. while  $i \geq 0$  &  $i \leq \text{epoch}$  do
3.  $\text{cur\_state\_index} = \text{random.rand.int}(0, \text{len}(\text{states}) - 1)$ 
4.  $\text{cur\_state} = \text{state}[\text{cur\_state\_index}]$ 
5. while  $\text{cur\_state} \neq \text{end}$  do
6.  $\text{Scores.fill}(0)$ 
7.  $\text{eos.fill}(0)$ 
8.  $\text{is\_eos} = \text{abs}(\text{mat} == \text{eos})$ 
9.  $\text{result} = \text{theta} * (1 - \text{f}_x) * \text{is\_eos} + \text{theta} * (1 - \text{is\_eos})$ 
10. return result

```

---

## 4 Experiment

### 4.1 Experiment process and environment

In our experiments, model training and testing work is performed on the following system and hardware:

- System: Ubuntu 16.04 Desktop
- Hardware: Xeon E5-2620 v3 @2.4 GHz 64 GB NVIDIA Tesla K40c
- Development Languages: Python, Lua
- Development Tool: vim, PyCharm
- Deep learning framework: Torch

### 4.2 Data set

The data sets used in this paper are mainly Chinese and English bilingual parallel corpora which is download from the website of NiuTrans translation. The corpus is divided into two documents in Chinese and English respectively, each of which records 1,000,000 pieces of data. Each line of the two files is corresponds one-to-one, and is a translation of each other. The CASIA corpus [Zhou, Li, Yin et al. (2010)] is also used in the experiments in this paper.

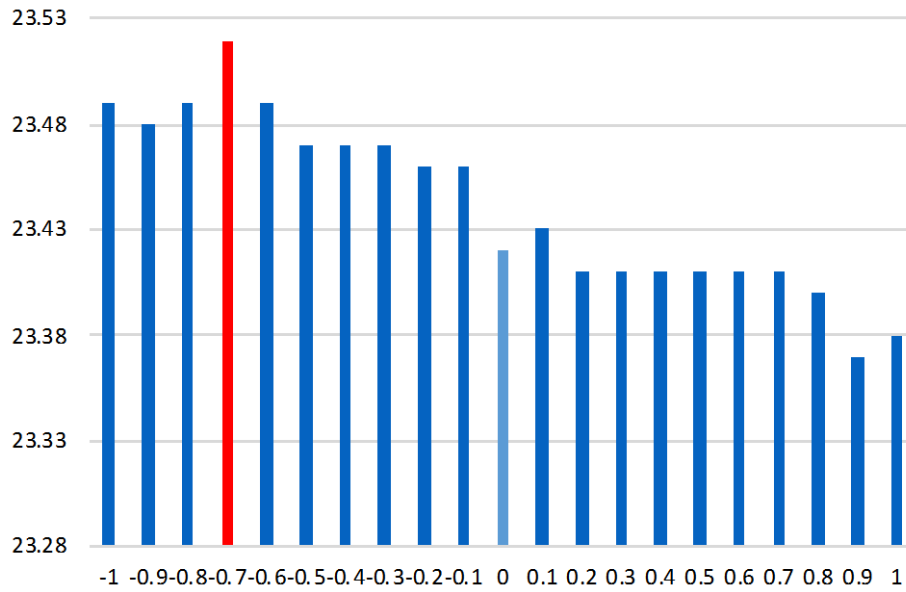
### 4.3 Experimental results on the first penalty term

We firstly use CASIA's corpus to train 24 epochs, then sets  $\delta$  to 20, and compares the BLEU scores of the 24th epochs, which the model with the  $\varepsilon$  in intervals  $[-2, 2]$  and  $[-1, 1]$ . The scores are shown in the Tab. 1 below:

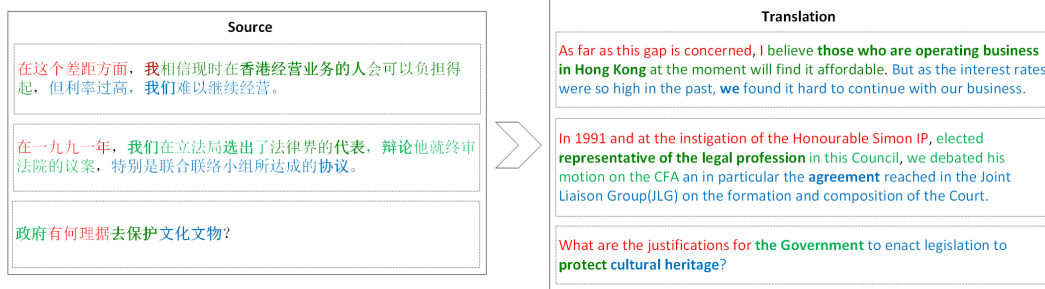
**Table 1:** BLEU scores for translation models corresponding to different  $\delta$  values

$\varepsilon$	Score
-1	BLEU=23.49, 60.3/33.4/19.5/12.1 (BP=0.893, ration=0.899)
-0.9	BLEU=23.48, 60.3/33.4/19.5/12.1 (BP=0.893, ration=0.899)
-0.8	BLEU=23.49, 60.3/33.4/19.5/12.1 (BP=0.894, ration=0.899)
-0.7	BLEU=23.52, 60.3/33.4/19.5/12.1 (BP=0.894, ration=0.899)
-0.6	BLEU=23.49, 60.3/33.3/19.5/12.1 (BP=0.894, ration=0.899)
-0.5	BLEU=23.47, 60.3/33.3/19.5/12.1 (BP=0.894, ration=0.899)

-0.4	BLEU=23.47, 60.3/33.3/19.5/12.1 (BP=0.894, ration=0.899)
-0.3	BLEU=23.47, 60.3/33.3/19.5/12.1 (BP=0.894, ration=0.899)
-0.2	BLEU=23.46, 60.2/33.3/19.5/12.1 (BP=0.895, ration=0.900)
-0.1	BLEU=23.46, 60.2/33.3/19.5/12.1 (BP=0.895, ration=0.900)
0	BLEU=23.42, 60.2/33.3/19.5/12.1 (BP=0.895, ration=0.900)
0.1	BLEU=23.43, 60.2/33.2/19.5/12.1 (BP=0.895, ration=0.901)
0.2	BLEU=23.41, 60.1/33.2/19.4/12.1 (BP=0.895, ration=0.900)
0.3	BLEU=23.41, 60.1/33.2/19.4/12.1 (BP=0.895, ration=0.900)
0.4	BLEU=23.41, 60.0/33.2/19.4/12.1 (BP=0.897, ration=0.902)
0.5	BLEU=23.41, 60.0/33.1/19.4/12.0 (BP=0.898, ration=0.903)
0.6	BLEU=23.41, 60.0/33.1/19.4/12.0 (BP=0.898, ration=0.903)
0.7	BLEU=23.41, 59.9/33.1/19.4/12.0 (BP=0.898, ration=0.903)
0.8	BLEU=23.40, 59.9/33.1/19.4/12.0 (BP=0.899, ration=0.903)
0.9	BLEU=23.37, 59.8/33.0/19.3/12.0 (BP=0.900, ration=0.904)
1	BLEU=23.38, 59.8/33.0/19.3/12.0 (BP=0.900, ration=0.905)



**Figure 4:** BLEU scores for translation models corresponding to different  $\delta$  values. When  $\delta=-0.7$  partial translation is shown in Fig. 5



**Figure 5:** Partial translation results at  $\delta=-0.7$

When  $\varepsilon=-0.7$ , the BLEU is the highest, which is 0.1 higher than the model that  $\varepsilon=0$  (the original model). We use the same coefficient in the corpus of NiuTrans translation, and also use 24 as the training epoch of the model, and get the following data:

**Table 2:** Comparison of BLEU scores between the translation model with  $\delta$  of -0.7 and the base model

$\varepsilon$	Score
0	BLEU = 26.29, 67.7/42.7/28.8/20.7 (BP=0.726, ration=0.757)
-0.7	BLEU = 26.37, 67.6/42.6/28.8/20.7 (BP=0.728, ration=0.759)

The BLEU score of the model has also been improved, proving the effectiveness of the method.

#### 4.4 Experimental results on the second penalty term

First, we count 1,000,000 pairs of sentences in the parallel corpus of NiuTrans. Second, we let the length of the target language sentences and the length of the corresponding original sentence do the ratio operation in each pair of sentences. Finally, for those ratios in the interval  $[0, 1]$ , use them to fit the distribution line chart. Results are shown in Fig. 7.

白日依山尽 ... .. Mountains cover the white sun	爱自己,是终身浪漫的开始 ... .. Love yourself, is the beginning of a lifetime of romance	这叫做“部落法则” ... .. This is called “Tribal Law”.	薪资福利丰厚,大量实习岗位 ... .. Rich salary and welfare, a large number of internship positions
平安喜乐,万事胜意 ... .. Peace and joy prevail over all things.	我们所给出的价格方案 ... .. The price plan we have given	一折起囤过年好货 ... .. 10% to hoard good Chinese New Year goods	如今音响外放最棒的笔记本 ... .. Nowadays, the best notebook is placed outside the stereo.
将表达式重组为一种语言 ... .. Reassemble expressions into a language	最近的降雨在80公里处 ... .. The most recent rainfall was 80 kilometers away.	没有你我不想永生不灭 ... .. I don't want live forever without you	十二星座新年运势如何锦上添花? ... .. How can the twelve constellations add to the icing on the cake in the New Year?
“碳循环”是最近很流行的减脂方法 ... .. “Carbon cycle” is a popular fat reduction method recently.	雅诗兰黛多效抗老三件套 ... .. Estee Lauder Multi-effect Anti-aging Three-piece Set	制作蒸包的独家秘方从不外传 ... .. The exclusive secret recipe for making steamed bags is never disclosed.	没选择的话,创造选择 ... .. If there is no choice, create a choice.

Figure 6: The corpus of NiuTrans

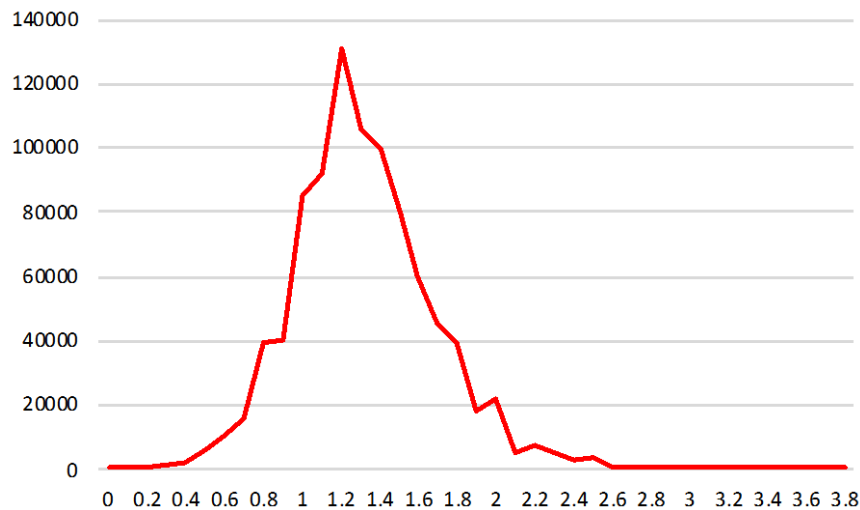
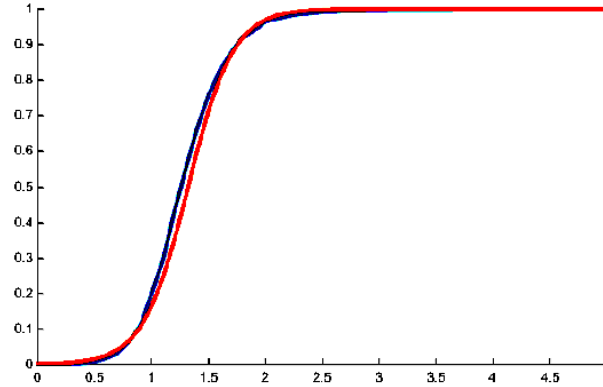


Figure 7: Distribution of corpus in the range [0.1, 3.9]

By narrowing the logarithmic interval, it is clear that the ratios are mainly taken in [0.2,

2.6]. Then, by plotting the cumulative distribution function of the data, we get the blue line in Fig. 8 below.



**Figure 8:** The cumulative curve and the fitted curve of the ratio of sentence pairs in the corpus

From the blue curve in Fig. 8 above, we can see that the ratio of most sentence pairs in the corpus is between (0.5, 2.5). The trajectory of this line is close to sigmoid, so we used a linear regression algorithm to fit it.

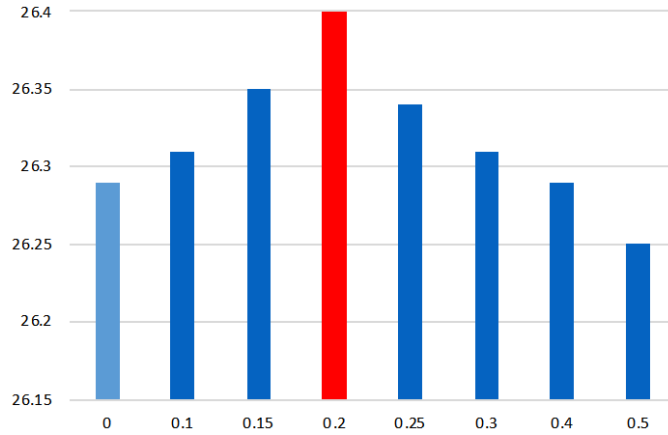
We first assume that  $l(x) = y, \frac{t}{s} = x$ . Secondly, because the shape of the curve is similar to sigmoid, so we bring  $x$  into  $y = \frac{1}{1+e^{ax+b}}$ . Then we define the error as  $\frac{1}{m} |y - y_{-}|^2$  and use the gradient descent method to find the minimum value. And then we got the formula (corresponding to the red curve in Fig. 8 above):

$$F_X(x) = \frac{1}{1+e^{ax+b}} \quad (10)$$

In the above formula,  $a=-5.0875, b=6.7187, x$  is the ratio of the length of the source language sentence to the length of the target language sentence. By substituting the fitted  $F_X(x)$  into the formula, we got a penalty term for the length ratio which is used to correct the evaluation function of the beam search. We take different values for  $\theta$ , use the corpus of NiuTrans, and use 24 epochs as our training epoch. We got the experimental data as follow:

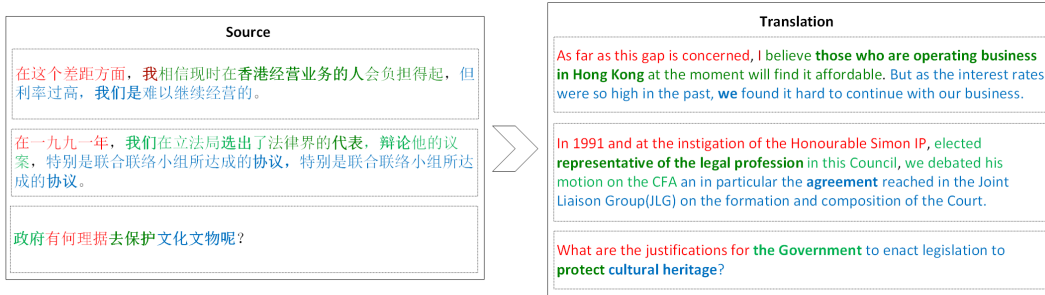
**Table 3:** BLEU scores for translation models corresponding to different  $\theta$  values

$\theta$	Score
0	BLEU = 26.29, 67.7/42.7/28.8/20.7 (BP=0.726, ration=0.757)
0.1	BLEU = 26.31, 67.7/42.7/28.8/20.7 (BP=0.726, ration=0.758)
0.15	BLEU = 26.35, 67.7/42.7/28.8/20.7 (BP=0.728, ration=0.760)
0.2	BLEU = 26.40, 67.6/42.6/28.8/20.7 (BP=0.730, ration=0.763)
0.25	BLEU = 26.34, 67.6/42.6/28.8/20.6 (BP=0.727, ration=0.759)
0.3	BLEU = 26.31, 67.6/42.6/28.8/20.6 (BP=0.727, ration=0.759)
0.4	BLEU = 26.29, 67.6/42.6/28.7/20.6 (BP=0.727, ration=0.759)
0.5	BLEU = 26.25, 67.6/42.6/28.7/20.6 (BP=0.727, ration=0.758)



**Figure 9:** BLEU scores for translation models corresponding to different  $\theta$  values

When  $\theta=0.2$ , part of the translation result is shown in Fig. 10.



**Figure 10:** Partial translation results at  $\theta=0.2$

It can be seen from the table that when  $\theta=0.2$ , the BLEU is the highest, which is 0.11 higher than the model that  $\varepsilon=0$  (the original model). We use the same coefficients in the CASIA corpus, and also use 24 as the training epoch of the model, and get the following data:

**Table 4:** Comparison of BLEU scores between the translation model with  $\theta$  of -0.7 and the original model

$\theta$	Score
0	BLEU=23.42, 60.2/33.3/19.5/12.1 (BP=0.895, ration=0.900)
0.2	BLEU=23.51, 60.3/33.4/19.5/12.1 (BP=0.896, ration=0.902)

After the improvement, the model’s BLEU score is also higher than the original model’s score, which proves the effectiveness of the method.

### 5 Conclusion

Traditional beam search always selects the word with the highest probability at the current time step, which is not necessarily the best choice for the whole translation, and result in the problem of duplicate translation. In addition, traditional beam search also prefers short results over long ones, thus, may miss part of the source statement. In this

paper, we proposed an innovative beam search evaluation function with two penalties based on duplicate detection and length ratio, to solve duplicate and missing translation issues. Furthermore, we use a novel translation evaluation system based on reinforcement learning to select better candidate words for generating translations. Then we conducted extensive experiments to evaluate our methods. CASIA corpus and the 1,000,000 pairs of bilingual corpora of NiuTrans are used in our experiments. Experimental results show that our two methods can effectively improve the quality of the translation.

**Acknowledgement:** Authors are very grateful to the helpful suggestion from the reviewers.

**Funding Statement:** This work is supported by the National Natural Science Foundation of China (61872231, 61701297).

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

- Bahdanau, D.; Cho, K.; Bengio, Y.** (2014): Neural machine translation by jointly learning to align and translate. *International Conference on Learning Representations, ICLR, 3rd International Conference on Learning Representations, ICLR Conference Track Proceedings*, <http://arxiv.org/abs/1409.0473>.
- Chen, Y. T.; Xiong, J.; Xu, W. H.; Zuo, J. W.** (2019): A novel online incremental and decremental learning algorithm based on variable support vector machine. *Cluster Computing*, vol. 22, no. 3, pp. 7435-7445.
- Cho, K.; Merriënboer, B. V.; Gülçehre, Ç.; Bahdanau, D.; Bougares, F. et al.** (2014): Learning phrase representations using RNN encoder-decoder for statistical machine translation. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1724-1734.
- Chollet, F.** (2017): Xception: deep learning with depthwise separable convolutions. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1800-1807.
- Chorowski, J.; Bahdanau, D.; Serdyuk, D.; Cho, K.; Bengio, Y.** (2015): Attention-based models for speech recognition. *Computer Science*, vol. 10, no. 4, pp. 429-439.
- Chrisman, L.** (1991): Learning recursive distributed representations for holistic computation. *Connection Science*, vol. 3, no. 4, pp. 345-366.
- Graves, A.** (2013): Generating sequences with recurrent neural networks. <http://arxiv.org/abs/1308.0850>.
- Kaiser, L.; Gomez, A. N.; Chollet, F.** (2017): Depthwise separable convolutions for neural machine translation. <http://arxiv.org/abs/1706.03059>.
- Kalchbrenner, N.; Blunsom, P.** (2013): Recurrent continuous translation models. *EMNLP Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, vol. 3, pp. 1700-1709.



- Lample, G.; Ott, M.; Conneau, A.; Denoyer, L.; Ranzato, M.** (2018): Phrase-based & neural unsupervised machine translation. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 5039-5049.
- Liu, J.; Lin, L.; Ren, H.; Gu, M.; Wang, J. et al.** (2018): Building neural network language model with pos-based negative sampling and stochastic conjugate gradient descent. *Soft Computing*, vol. 22, no. 20, pp. 6705-6717.
- Liu, J.; Yang, Y.; Lv, S.; Wang, J.; Chen, H.** (2019): Attention-based BiGRU-CNN for Chinese question classification. *Journal of Ambient Intelligence and Humanized Computing*, <https://doi.org/10.1007/s12652-019-01344-9>.
- Luong, M. T.; Pham, H.; Manning, C. D.** (2015): Effective approaches to attention-based neural machine translation. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1412-1421. <https://doi.org/10.18653/v1/D15-1166>.
- Mikolov, T.; Corrado, G.; Chen, K.; Dean, J.** (2013): Efficient estimation of word representations in vector space, arxiv.CORR, <http://arxiv.org/abs/1301.3781>.
- Moryossef, A.; Aharoni, R.; Goldberg, Y.** (2019): Filling gender & number gaps in neural machine translation with black-box context Injection. *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pp. 49-54.
- NiuTrans, NiuTrans.** (2011): NiuTrans machine translation system. <https://niutrans.vip/documents/help>.
- Schwenk, E. S.; Udani, A. D.; Gupta, R. K.; Mariano, E. R.** (2017): How academic physicians can benefit from social media. *Revista Española de Anestesiología y Reanimación*, vol. 65, no. 2, pp. 103-107.
- Sennrich, R.; Haddow, B.; Birch, A.** (2015): Neural machine translation of rare words with subword units. *54th Annual Meeting of the Association for Computational Linguistics, ACL Long Papers*, vol. 3, pp. 1715-1725.
- Sutskever, I.; Vinyals, O.; Le, Q. V.** (2014): Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems*, vol. 4, pp. 3104-3112.
- Thắng, L.; Sutskever, I.; Le, Q.; Vinyals, O.; Zaremba, W.** (2014): Addressing the rare word problem in neural machine translation. *ACL-IJCNLP 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Proceedings of the Conference*, vol. 1, pp. 11-19.
- Williams, I.; Aleksic, P. S.** (2017): Rescoring-aware beam search for reduced search errors in contextual automatic speech recognition. *Interspeech*, pp. 508-512.
- Wiseman, S.; Rush, A. M.** (2016): Sequence-to-sequence learning as beam-search optimization. *Conference on Empirical Methods in Natural Language Processing, The Association for Computational Linguistics*, pp. 1296-1306.
- Wu, Y.; Schuster, M.; Chen, Z.; Le, Q. V.; Norouzi, M. et al.** (2016): Google's neural machine translation system: bridging the gap between human and machine translation. *Transactions of the Association for Computational Linguistics*, pp. 339-351.

**Zeng, D. J.; Dai, Y.; Li, F.; Wang, J.; Sangaiah, A. K.** (2019): Aspect based sentiment analysis by a linguistically regularized CNN with gated mechanism. *Journal of Intelligent & Fuzzy Systems*, vol. 36, no. 5, pp. 3971-3980.

**Zeng, D.; Dai, Y.; Li, F.; Sherratt, R. S.; Wang, J.** (2018): Adversarial learning for distant supervised relation extraction. *Computers, Materials & Continua*, vol. 55, no. 1, pp. 121-136.

**Zhang, J.; Jin, X.; Sun, J.; Jin, W.; Kumar, S. A.** (2018): Spatial and semantic convolutional features for robust visual object tracking. *Multimedia Tools and Applications*, <https://doi.org/10.1007/s11042-018-6562-8>.

**Zhang, J.; Wang, W.; Lu, C.; Wang, J.; Sangaiah, A. K.** (2019): Lightweight deep network for traffic sign classification. *Annals of Telecommunications*, <https://doi.org/10.1007/s12243-019-00731-9>.

**Zhang, J.; Xie, Z.; Sun, J.; Zou, X.; Wang, J.** (2020): A cascaded R-CNN with multiscale attention and imbalanced samples for traffic sign detection. *Institute of Electrical and Electronics Engineers Access*, vol. 8, pp. 29742-29754.

**Zhou, K.; Li, A.; Yin, Z.; Zong, C.** (2010): CASIA-CASSIL: a Chinese telephone conversation corpus in real scenarios with multi-leveled annotation. *Proceedings of the International Conference on Language Resources and Evaluation, LREC*, pp. 17-23.