Case-Sensitive Neural Machine Translation

Xuewen Shi^{1,2}, Heyan Huang^{1,2}, Ping Jian^{1,2(✉)}, and Yi-Kun Tang^{1,2}

¹ School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China

{xwshi,hhy63,pjian,tangyk}@bit.edu.cn

² Beijing Engineering Research Center of High Volume Language Information Processing and Cloud Computing Applications, Beijing, China

Abstract. Even as an important lexical information for Latin languages, word case is often ignored in machine translation. According to observations, the translation performance drops significantly when we introduce case-sensitive evaluation metrics. In this paper, we introduce two types of case-sensitive neural machine translation (NMT) approaches to alleviate the above problems: i) adding case tokens into the decoding sequence, and ii) adopting case prediction to the conventional NMT. Our proposed approaches incorporate case information to the NMT decoder by jointly learning target word generation and word case prediction. We compare our approaches with multiple kinds of baselines including NMT with naive case-restoration methods and analyze the impacts of various setups on our approaches. Experimental results on three typical translation tasks (Zh-En, En-Fr, En-De) show that our proposed methods lead to the improvements up to 2.5, 1.0 and 0.5 in case-sensitive BLEU scores respectively. Further analyses also illustrate the inherent reasons why our approaches lead to different improvements on different translation tasks.

Keywords: Natural language processing · Neural machine translation · Case-sensitive

1 Introduction

In the real world, many of the natural language texts that are written in Latin language are case sensitive, such as English, French, German, etc. For many natural language processing (NLP) tasks, case information is an important feature for algorithms to distinguish sentence structures, identify the part-of-speech of a word, and recognize named entities. However, most existing machine translation approaches pay little attention to the capitalization correctness of the generated words, which does not meet the needs of practical requirements and may introduce noise to downstream NLP applications [9, 20].

In fact, there is a contradiction in the training corpus preprocessing process: using lowercased corpus can reduce the expansion of the vocabulary but neglecting some morphology information, while keeping the original morphological form

Chinese (pinyin)	English
zhè shì yī kē <i>píngguǒ</i> .	This is an apple .
zhè shì yī kuài <i>píngguǒ</i> shǒubiǎo.	This is an Apple Watch.

Fig. 1. Two example of Zh-En translation. The Chinese side is presented in pinyin. “*píngguǒ*” and “apple” are aligned words pair, which are same in the source side but written in different case in the target side in our examples. The contradiction is that using lowercased “apple” in the second example will lose the information of a proper noun, while using a individual word “Apple” will lose the semantic connection with the parallel pair (“*píngguǒ*” “apple”).

Table 1. Case insensitive/sensitive BLEU scores on Zh-En translation. Δ represents the reduced BLEU scores compared to the “insensitive”. *NRC* is a rule-based case restoring method and more experiment setup details are described in Sect. 5.2.

Models	Insensitive	Sensitive (Δ)
Regular case	43.23	41.45 (−1.78)
True-case	43.08	40.02 (−3.06)
True-case + <i>NRC</i>	43.08	41.22 (−1.86)
Lowercase	43.49	28.49 (−15.00)
Lowercase + <i>NRC</i>	43.49	30.02 (−12.49)

will increase the vocabulary and lose its connection with the lowercase form of the word. Figure 1 gives an example to illustrate this contradiction. Using true-cased corpus seems to balance the unnecessary increasing vocabulary and the missing morphology information of word case. However, re-storing cases from true-cased corpus is not as easy as the reverse process. Table 1 shows that using corpus in lowercase and regular case gets the highest case-insensitive and case-sensitive BLEU scores respectively, which reflects the difficulty of case restoration.

In this paper, we introduce case-sensitive neural machine translation (NMT) approaches to alleviate the above problems. In our approaches, we apply lowercased vocabulary to both the source input and target output side in the NMT model, and the model is trained to jointly learn to generate translation and distinguish the capitalization of the generated words. During the decoding step, the model predicts the case of the output word while generating the translation.

Specifically, we proposed two kinds of methods to this extent: i) mixing case tokens into lowercased corpus to indicate the real case of the adjacent word; ii) expanding NMT model architecture with an additional network layer that performances case prediction. We evaluate on pairs of linguistically disparate corpora in three translation tasks: Chinese-English (Zh-En), English-German (En-De) and English-French (En-Fr), and observe that the proposed techniques improve translation quality on case-sensitive BLEU [16]. We also study the model performances on case-restoration tasks and experimental results show that our proposed methods lead to improvements on P , R and F_1 scores.

2 Related Work

Recently, neural machine translation (NMT) with encoder-decoder framework [6] has shown promising results on many language pairs [8, 21], and incorporating linguistic knowledge into neural machine translation has been extensively studied [7, 12, 17]. However, the procedure of NMT decoding rarely considers the case correctness of the generated words, and there are approaches performing case restoration on the machine generated texts [9, 20].

Recent efforts have demonstrated that incorporating linguistic information can be useful in NMT [7, 12, 15, 17, 22, 23]. Since the source sentence is definitive and easy to attach extra information, it is a straightforward way to improve the translation performance by using the source side features [12, 17]. For example, Sennrich and Haddow incorporate linguistic features to improve the NMT performance by appending feature vectors to word embeddings [17], and the source side hierarchical syntax structures are also used for achieving promising improvement [7, 12]. It is uncertain to leverage target syntactic information for NMT as target words in the real decoding process. Niehues and Cho apply multi-task learning where the encoder of the NMT model is trained to produce multiple tasks such as POS tagging and named-entity recognition into NMT models [15]. There are also works that directly model the syntax of the target sentence during decoding [22–24].

Word case information is a kind of lexical morphology which is definitive and easy to be obtained without any additional annotation and parsing of the training corpus. Recently, a joint decoder is proposed for predicting words as well as their cases synchronously [25], which shares a similar spirit with a part of our approaches (see Sect. 4.2). The main distinction of our approaches is that we propose two series of case-sensitive NMT and study various model setups.

3 Neural Machine Translation

Given a source sentence $\mathbf{x} = \{x_1, x_2, \dots, x_{T_x}\}$ and a target sentence $\mathbf{y} = \{y_1, y_2, \dots, y_{T_y}\}$, most of popular neural machine translation approaches [3, 8, 21] directly model the conditional probability:

$$p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}) = \prod_{t=1}^T p(y_t|\mathbf{y}_{<t}, \mathbf{x}; \boldsymbol{\theta}), \quad (1)$$

where $\mathbf{y}_{<t}$ is the partial translation before decoding step t and $\boldsymbol{\theta}$ is a set of parameters of the NMT model.

In this paper, the proposed approaches make scarcely assumptions about the specific NMT model, and it can be applied to any popular encoder-decoder based NMT model architecture [8, 21]. To simplify the experiment and highlight our contributions, we take the Transformer [21], one of the popular state-of-the-art NMT models, as the specific implementation of the baseline NMT model. Specifically, the encoder contains a stack of six identical layers. Each layer consists of

two sub-layers: i) a multi-head self-attention mechanism, and ii) a position-wise fully connected feed-forward network. A residual connection is applied around each of the two sub-layers, followed by layer normalization [2]. The decoder is also composed of a stack of six identical layers. Besides the two sub-layers stated above, a third sub-layer is inserted in each layer that performs multi-head attention over the output of the encoder. The implementations of our approaches are all based on the above model architecture. Following the base model setups of the Transformer [21], we use 8 attention heads, 512-dimensional output vectors for each layer, and 2048-dimensional inner-layer of the feed-forward network.

4 Approaches

4.1 Adding Case Token

The technique of adding artificial tokens is a straightforward and practical way to incorporate additional knowledge to NMT [4, 18], since it hardly modifies the model architecture or increases the model parameters.

In our approach, we add two artificial tokens “<ca>” and “<ab>” to indicate capital words and abbreviation words in a sequence, respectively. This special token can be insert to the left (*LCT*) or the right (*RCT*) side of the capital word. For the target sequence, *LCT* represents to predict the case of word previously and then generate general target language word and the case is opposite for applying *RCT*.

For the corpus segmented by subword units [11, 19], we insert the *LCT* to the left side of the first subword unit of a capital word and insert *RCT* to the right side of the last subword unit of a capital word. For instance, Fig. 2 shows the modified sentences by adding *LCT* and *RCT* given the original sentence and the sentence encoded by subword units.

<i>Original</i>	Executive Committee of FIFA also announced some reform measures .
<i>LCT</i>	<ca> executive <ca> committee of <ab> fifa also announced some reform measures .
<i>RCT</i>	executive <ca> committee <ca> of fifa <ab> also announced some reform measures .
<i>Subword</i>	_Executive _Committee _of _FIFA _also _announc ed _some _reform _measures _.
<i>Subword+LCT</i>	_ <ca> _executive _ <ca> _committee _of _ <ab> _fifa _also _an-nounc ed _some _reform _measures _.
<i>Subword+RCT</i>	_executive _ <ca> _committee _ <ca> _of _fifa _ <ab> _also _an-nounc ed _some _reform _measures _.

Fig. 2. Examples of modifying original sentence (or sentence encoded by subwords) by *LCT* and *RCT*. “<ca>” and “<ab>” are two additional artificial tokens for indicating capital words and abbreviation words.

4.2 NMT Jointly Learning Case Prediction

In this approach, we add an additional case prediction output to the decoder of the encoder-decoder based NMT model on each decoding step. Given a source sentence: $\mathbf{x} = \{x_1, x_2, \dots, x_{T_x}\}$, its target translation: $\mathbf{y} = \{y_1, y_2, \dots, y_{T_y}\}$, and the case category sequence of the target language: $\mathbf{c} = \{c_1, c_2, \dots, c_{T_c}\}$, the goal of the extension is to enable NMT model to compute the joint probability $P(\mathbf{y}, \mathbf{c}|\mathbf{x})$. The overall joint model can be computed as:

$$P(\mathbf{y}, \mathbf{c}|\mathbf{x}) = \prod_{t=1}^{T_y} p(y_t|\mathbf{y}_{<t}, \mathbf{c}_{<t}, \mathbf{x})p(c_t|\mathbf{y}_{<t}, \mathbf{c}_{<t}, \mathbf{x}). \quad (2)$$

Fig. 3. The graphical illustrations of the proposed approaches. The hollow circle with black dashed lines represents the next word/case to be generated. (a) represents adding case token without modifying the decoder (see Sect. 4.1 for more details); (b), (c) and (d) are three kinds of implementations for joint predicting word and its cases (see Sect. 4.2 for more details).

Intuitively, there are three assumptions about joint predicting c_t at time step t : i) predicting c_t before generating the word y_t (CP_{pre}), ii) predicting c_t after the word y_t generated (CP_{pos}), and iii) predicting the probability of c_t and y_t synchronously (CP_{syn}).

CP_{pos} : At the time step t , the model first predict y_t and then predict the case c_t for the known word y_t , which is consistent with most of the case restoration process (as shown in Fig. 3(b)). Under this assumption, the conditional probabilities in Eq. (2) can be computed as:

$$p(y_t|\mathbf{y}_{<t}, \mathbf{c}_{<t}, \mathbf{x}) = g_y(y_{t-1}, z_t, s_t, h_t) \quad (3)$$

and

$$p(c_t|\mathbf{y}_{\leq t}, \mathbf{c}_{<t}, \mathbf{x}) = g_c(y_t, c_{t-1}, z_t, s_t, h_t), \quad (4)$$

respectively, where s_t and z_t are self-attention based context vectors of previous generated $\mathbf{y}_{<t}$ and $\mathbf{c}_{<t}$. h_t is the output of the encoder. $g_y(\cdot)$ is the output layer

of the Transformer [21] decoder, and $g_c(\cdot)$ is the additional output layer that performs case prediction. z_t and $g_c(\cdot)$ compose an additional 1-layer Transformer-based decoder with one attention head, 32-dimensional output vectors and inner-layer of the feed-forward network, which works parallel with the original NMT decoder.

CP_{pre} : For the case of CP_{pre} , the decoder first estimates the categories of the probable word for narrowing the selection of the vocabulary and then further confirms the output words (as shown in Fig. 3(c)). Under this assumption, the conditional probabilities in Eq. (2) can be computed as:

$$p(c_t | \mathbf{y}_{<t}, \mathbf{c}_{<t}, \mathbf{x}) = g_c(c_{t-1}, z_t, s_t, h_t) \quad (5)$$

and

$$p(y_t | \mathbf{y}_{<t}, \mathbf{c}_{\leq t}, \mathbf{x}) = g_y(y_{t-1}, c_t, z_t, s_t, h_t). \quad (6)$$

CP_{syn} : Under this assumption, the two generation processes are simultaneous and independent to each other (as shown in Fig. 3(d)), then, the decoder predicting the probability of c_t and y_t synchronously. The conditional probabilities in Eq. (2) can be computed as:

$$p(c_t | \mathbf{y}_{<t}, \mathbf{c}_{<t}, \mathbf{x}) = g_c(c_{t-1}, z_t, s_t, h_t) \quad (7)$$

and

$$p(y_t | \mathbf{y}_{<t}, \mathbf{c}_{<t}, \mathbf{x}) = g_y(y_{t-1}, z_t, s_t, h_t). \quad (8)$$

4.3 Adaptive Scaling Algorithm

In training corpus, capitalized words account for a small percentage of the vocabulary, which leads to class inequality problem for training case classification model. Reported results indicate that simply applying standard classification paradigm to class inequality tasks will result in deficient performance [1, 13]. To alleviate this problem, we apply Adaptive Scaling (AS) [13] to the case prediction training.

We refer words with uppercase letters as the positive instances and regard lowercased words as the negative instances. Formally, given P positive training instances $\mathcal{P}$ and N negative instances $\mathcal{N}$, $TP(\theta)$ and $TN(\theta)$ are the number of correctly predicted positive instances and the number of correctly predicted negative instances on training data with respect to θ -parameterized model. Then, taking the loss of CP_{pre} as the example, the loss function is modified as:

$$\mathcal{L}_{AS}(\theta) = - \sum_{(c_j, y_j) \in \mathcal{P}} \log p(c_j | y_j; \theta) - \sum_{(c_j, y_j) \in \mathcal{N}} w(\theta) \cdot \log p(c_j | y_j; \theta), \quad (9)$$

where

$$w(\theta) = \frac{TP(\theta)}{P + N - TN(\theta)}. \quad (10)$$

Batch-wise Adaptive Scaling Algorithm. In practice, most NMT models are trained with batch-wise gradient based algorithm, so we apply the batch-wise

version of the adaptive scaling algorithm [13] in our work. Let $\mathcal{P}^B$ represents P^B positive instances and $\mathcal{N}^B$ denotes N^B negative instances in a batch, TP^B and TN^B is estimated as:

$$TP^B(\theta) = \sum_{(c_i, y_i) \in \mathcal{P}^B} p(c_i | y_i; \theta) \quad (11)$$

and

$$TN^B(\theta) = \sum_{(c_i, y_i) \in \mathcal{N}^B} p(c_i | y_i; \theta). \quad (12)$$

Then $w(\theta)^B$ is estimated as:

$$w(\theta) = \frac{TP^B(\theta)}{P^B + N^B - TN^B(\theta)}. \quad (13)$$

5 Experiment

5.1 Datasets and Setups

To verify the effectiveness of the proposed methods, we evaluate the proposed approaches on three typical translation tasks: Chinese-English (Zh-En), English-French (En-Fr) and English-German (En-De). The above three language pair translation tasks represent three typical application scenarios: i) the source language does not share any word capitalization information with the target language (Zh-En); ii) the word capitalizing rules of the source language and the target language are approximate (En-Fr); iii) the words capitalizing rules for the source language and the target language are not the same (En-De). Those typical translation tasks will be helpful to study the effects of word cases on NMT performances.

Chinese-English. For Chinese-English translation, our training data are extracted from three LDC corpora¹. The training set contains about 1.3M parallel sentence pairs. For preprocessing, the Chinese part for both training sets and testing sets is segmented by the LTP Chinese word segmentor [5]. With the encoding of unigram language model [11], we get a Chinese vocabulary of about 39K tokens, and an English vocabulary of about 40K words. We use NIST02 as our validation set and use NIST2003–NIST2005 datasets as our test sets.

English-German and English-French. For English-German and English-French translation, we conduct our experiments on the publicly available corpora of WMT’14 dataset. This data set contains 4.5M sentence pairs and 18M sentence pairs for En-De translation task and the significantly larger En-Fr dataset consisting of 18M sentences pairs. We encode the corpora with unigram language

¹ LDC2005T10, LDC2003E14, LDC2004T08 and LDC2002E18. LDC2003E14 is a document alignment comparable and parallel sentence pairs are extracted by Champollion tools [14].

model [11], and both source and target vocabulary contain about 37K tokens and 30K tokens for En-De and En-Fr translation tasks, respectively. We report results on newstest2014, and the newstest2013 is used as validation.

For all translation tasks, we tokenize all corpora with the Moses tokenizer² before applying subword units, and sentences longer than 200 words are discarded.

Evaluation. Following [21], we report the result of a single model obtained by averaging the 5 checkpoints around the best model selected on the development set. We apply beam search during decoding with the beam size of 6. The translation results in this paper are measured in both case-insensitive and case-sensitive BLEU scores [16] evaluated by the multi-bleu.perl script (See footnote 2). We also analyze the model performances on word case restoration tasks and evaluate the results on P , R and F_1 scores.

5.2 Baselines

Regular Case (RC) and Lowercase (LC). RC and LC represent using original corpus in regular case and lowercasing all training corpus, respectively.

Table 2. “Case-insensitive/case-sensitive” BLEU scores on Zh-En, En-Fr and En-De translation tasks. For the column of “Models”, **bold** and *italic* represents target words cases and performed methods, respectively.

#	Models	Zh-En			En-Fr	En-De
		NIST2003	NIST2004	NIST2005	newstest2014	newstest2014
1	RC	44.42/42.74	42.86/41.37	41.51/39.72	39.61/38.70	26.88/26.40
2	TC + <i>NRC</i>	42.42/40.63	43.61/42.00	41.97/40.58	39.63/38.65	26.87/26.32
3	LC + <i>NRC</i>	43.11/37.03	44.02/38.92	42.12/36.67	39.94/35.43	26.91/23.49
4	LC + <i>JPM</i> [25]	43.48/41.72	43.97/42.73	42.81/41.12	40.01/39.11	26.77/26.37
5	LC + <i>LCT</i>	42.70/40.83	44.10/42.45	41.66/40.31	39.71/38.89	26.48/25.66
6	LC + <i>RCT</i>	41.71/39.72	43.18/41.53	41.36/39.92	39.84/39.05	26.38/25.57
7	LC + <i>CP_{pre}</i>	44.53 /42.88	44.84/43.37	42.95/41.72	40.85 /39.70	26.94/26.58
8	LC + <i>CP_{pos}</i>	43.64/42.10	44.36/43.39	42.84/41.55	40.72/39.69	26.72/26.45
9	LC + <i>CP_{syn}</i>	43.62/42.74	44.53/42.98	43.01/41.92	40.02/38.78	26.78/26.54
10	LC + <i>CP_{pre}</i> + <i>AS</i>	44.52/ 43.04	45.01 / 43.97	42.91/42.05	40.70/ 39.75	27.02 / 26.87
11	LC + <i>CP_{pos}</i> + <i>AS</i>	43.76/42.87	44.75/43.81	42.92/41.86	40.59/39.60	26.89/26.56
12	LC + <i>CP_{syn}</i> + <i>AS</i>	43.57/42.82	44.66/43.56	43.01 / 42.07	40.61/39.67	26.93/26.64

Truecase (TC). We truecase the target language part of the corpora using Moses [10] script truecase.perl (See footnote 2). It tries to keep words in their natural case, and only changes the words at the beginning of their sentence to their most frequent form.

² The related Moses [10] scripts are available at: <https://github.com/moses-smt/mosesdecoder/blob/master/scripts/>.

Naive Re-case (*NRC*). For comparative research, we also apply rule-based methods that restore model outputs into regular case. We first build a capitalized words dictionary based on the target side of the translation corpus. The dictionary counts the words that usually appear in capitalized form (the frequency of occurrence in capitalized form is greater than 50%).

Joint Prediction Model (*JPM*) [25]. This work shares similar motivation with our approach. It proposes an NMT model that jointly predicts English words and their cases by employing two outputs layers on one decoder.

5.3 Main Results

Table 2 shows the experimental results on the three translation tasks. The results of baseline methods are listed in rows of #1–#4 and from #5 to #12 are results of our proposed methods. From Table 2 we can observe that for each experimental setup, model gains higher case-insensitive BLEU score than the case-sensitive version. The reduction in case-sensitive BLEU is more pronounced in Zh-En translation, since the source language does not provide any relevant morphological information. For En-Fr translation, since the target language shares similar capitalization rules with the source input, the case-sensitive performance reduces less. The phenomenon is not very prominent in En-De translation, probably because the writing rules of German are different from the other two languages (En and Fr).

The results show that our proposed *CP* methods obtain better performances than multiple baseline setups on the three translation tasks, but the translation quality of *LCT* and *RCT* decrease in some cases. The reason for the negative results on “Adding case token” approaches may be: i) the additional case tokens

Table 3. Case restoration results on Zh-En, En-Fr and En-De translation tasks. For the column of “Models”, **bold** and *italic* represents target words cases and performed methods, respectively.

#	Methods	Zh-En			En-Fr			En-De		
		<i>P</i>	<i>R</i>	<i>F</i> ₁	<i>P</i>	<i>R</i>	<i>F</i> ₁	<i>P</i>	<i>R</i>	<i>F</i> ₁
1	RC	74.19	63.15	68.23	75.62	81.23	78.32	58.39	58.12	58.25
2	TC + <i>NRC</i>	73.98	63.58	68.39	75.41	81.33	78.25	57.86	58.24	58.05
3	LC + <i>NRC</i>	74.70	17.07	27.79	75.56	24.15	36.60	57.25	51.07	53.99
4	LC + <i>JPM</i> [25]	73.10	64.87	68.74	74.91	82.68	78.60	57.37	58.46	57.91
5	LC + <i>LCT</i>	71.22	65.29	68.13	73.46	83.67	78.23	56.05	58.53	57.26
6	LC + <i>RCT</i>	69.53	65.59	67.50	72.84	83.01	78.03	56.22	58.35	57.26
7	LC + <i>CP</i> _{pre}	73.67	65.43	69.31	74.89	83.20	78.82	57.74	58.54	58.13
8	LC + <i>CP</i> _{pos}	73.26	64.87	68.81	75.07	82.12	78.43	57.31	58.42	57.85
9	LC + <i>CP</i> _{syn}	73.71	64.24	68.65	74.65	83.09	78.64	57.40	58.65	58.01
10	LC + <i>CP</i> _{pre} + <i>AS</i>	73.73	66.39	69.87	75.42	83.44	79.22	58.09	58.23	58.15
11	LC + <i>CP</i> _{pos} + <i>AS</i>	73.65	66.17	69.71	75.08	83.30	78.97	57.93	58.76	58.34
12	LC + <i>CP</i> _{syn} + <i>AS</i>	73.68	66.20	69.74	75.13	83.25	78.98	58.20	58.29	58.24

increase the average length of the generated sequences by more than 5 words; ii) decoding with case tokens inside the sequence may dilute the impacts of previous generated words. On the contrast, *CP* methods use relatively independent decoders and the case information performs as additional feature inputs for NMT decoding.

One of the interesting findings in the overall results is that *LCT* and *CP_{pre}* usually gains better BLEU scores than *RCT* and *CP_{pos}*. In our approaches, *LCT* and *CP_{pre}* predict case label before generate target word while *RCT* and *CP_{pos}* follow the reverse order. We suspect that the possible reason for experimental results is that the generated label can reduce the search space of the target words. *CP_{pos}* and *CP_{syn}* also achieve improvements on baseline methods. We also study the impact of applying adaptive scaling algorithm [13] and the results are listed in rows from #10 to #12 in Table 2. The experimental results show that the proposed methods with scaling algorithm (*AS*) performs better under case-sensitive measurements, which indicates that applying *AS* can enhance the prediction of the word cases.

5.4 Case Restoration

In this section, we analyze the impact of different methods on case restoration tasks. We conduct the experiments on three testsets: NIST2003–NIST2005 (Zh-En), newstest2014 and newstest2015 (En-Fr and En-de). Experimental results are evaluated on *P*, *R* and *F₁*³ scores as shown in Table 3. From Table 3, we can see that our proposed *CP* methods gains higher *F₁* scores for most model setups. Comparing with adding case token, *CP* approaches separate case prediction from word prediction and introduce an additional network block to handle this task. Since the separated case prediction decoder learns more about lexical information, it leads to improvements on case-restoration tasks. As shown in rows #10–#12, applying adaptive scaling algorithm [13] is also effective for case prediction tasks, which can adaptively scale the influence of negative instances in loss function.

Comparing the three translation tasks, *P*, *R* and *F₁* performances on En-Fr are significantly better than other languages. As mentioned above, French shares similar capitalization rules with English, so the decoder can capture more lexical information from the source side. The *NRC* method works much better on En-De translation tasks than the others, since the capitalization rules for German words are relatively fixed.

5.5 Decoding Efficiency

We analyze the decoding time of our method compared with the baseline approach on NIST2002 Zh-En valid set with one NVIDIA GeForce GTX 1080Ti GPU and a batch size of 32 sentences. Table 4 shows that the proposed methods

³ We regard a word sequence as a collection without considering the word orders in the generated sequence for calculating *P*, *R* and *F₁* scores.

has lower decoding efficiency than the baseline method. For NMT with CP methods, the additional decoder creates extra decoding overhead. Especially, CP_{pre} and CP_{pos} require additional autoregressive steps to predict word case, which reduces decoding efficiency. For the approach of adding case token, although it does not increase the parameters number of NMT model, the additional tokens increase the length of generated target sequence.

Table 4. Comparison of model parameters and decoding efficiency. “#parameters” represents the free parameters number of the NMT model. “Speed” means the decoding speed (sentences per second) which are evaluated on NIST2002 Zh-En valid set.

Methods	#parameters	Speed (sents/s)
Transformer [21]	65×10^6	5.88
+ case token	65×10^6	5.06
+ CP_{syn}	66×10^6	3.64
+ $CP_{pre}/ + CP_{pos}$	66×10^6	2.85

6 Conclusion

Word case information, as one of linguistics features, is definitive and easily obtainable. Incorporating case information into machine translation also meets the needs of practical applications. In this paper, we propose two types of approaches to perform case-sensitive neural machine translation: i) directly adding a case token into the word sequence to indicate the case information of the nearby word; ii) applying additional decoder to the conventional NMT model performing case prediction along with generating translation. We test our approaches on multiple setups and three typical translation tasks (Zh-En, En-Fr, En-De). Experimental results show that our approaches outperform baseline approaches on case-sensitive BLEU score. Specifically, adding case token is easy to apply to any NMT models without modifying the network architecture but lose some accuracy, while applying case prediction decoder will offer more reliable results but increasing model parameters. In the future, we will apply our approaches to other natural language generation relatively tasks such as dialogue generation, text generation and automatic speech recognition.

Acknowledgments. We thank all anonymous reviewers for their valuable comments. This work is supported by the National Key Research and Development Program of China (Grant No. 2017YFB1002103) and the National Natural Science Foundation of China (No. 61732005).

References

1. Anand, R., Mehrotra, K.G., Mohan, C.K., Ranka, S.: An improved algorithm for neural network classification of imbalanced training sets. *IEEE Trans. Neural Netw.* 4(6), 962–969 (1993)

2. Ba, L.J., Kiros, R., Hinton, G.E.: Layer normalization. CoRR abs/1607.06450 (2016)
3. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015, Conference Track Proceedings (2015)
4. Britz, D., Le, Q., Pryzant, R.: Effective domain mixing for neural machine translation. In: Proceedings of the Second Conference on Machine Translation, pp. 118–126 (2017)
5. Che, W., Li, Z., Liu, T.: LTP: a Chinese language technology platform. In: COLING 2010, 23rd International Conference on Computational Linguistics, Demonstrations Volume, Beijing, China, 23–27 August 2010, pp. 13–16 (2010)
6. Cho, K., et al.: 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 2014, Doha, Qatar, 25–29 October 2014, A Meeting of SIGDAT, a Special Interest Group of the ACL, pp. 1724–1734 (2014)
7. Eriguchi, A., Hashimoto, K., Tsuruoka, Y.: Tree-to-sequence attentional neural machine translation. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, Berlin, Germany, 7–12 August 2016, Long Papers, vol. 1, pp. 823–833 (2016)
8. Gehring, J., Auli, M., Grangier, D., Yarats, D., Dauphin, Y.N.: Convolutional sequence to sequence learning. In: Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6–11 August 2017, pp. 1243–1252 (2017)
9. Han, B., Cook, P., Baldwin, T.: Lexical normalization for social media text. *ACM Trans. Intell. Syst. Technol.* **4**(1), 5:1–5:27 (2013)
10. Koehn, P., et al.: Moses: open source toolkit for statistical machine translation. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions, pp. 177–180 (2007)
11. Kudo, T.: Subword regularization: improving neural network translation models with multiple subword candidates. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, 15–20 July 2018, Long Papers, vol. 1, pp. 66–75 (2018)
12. Li, J., Xiong, D., Tu, Z., Zhu, M., Zhang, M., Zhou, G.: Modeling source syntax for neural machine translation. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, 30 July–4 August, Long Papers, vol. 1, pp. 688–697 (2017)
13. Lin, H., Lu, Y., Han, X., Sun, L.: Adaptive scaling for sparse detection in information extraction. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, 15–20 July 2018, Long Papers, vol. 1, pp. 1033–1043 (2018)
14. Ma, X.: Champollion: a robust parallel text sentence aligner. In: Proceedings of the Fifth International Conference on Language Resources and Evaluation, LREC 2006, Genoa, Italy, 22–28 May 2006, pp. 489–492 (2006)
15. Niehues, J., Cho, E.: Exploiting linguistic resources for neural machine translation using multi-task learning. In: Proceedings of the Second Conference on Machine Translation, WMT 2017, Copenhagen, Denmark, 7–8 September 2017, pp. 80–89 (2017)

16. Papineni, K., Roukos, S., Ward, T., Zhu, W.: Bleu: a method for automatic evaluation of machine translation. In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, PA, USA, 6–12 July 2002, pp. 311–318 (2002)
17. Sennrich, R., Haddow, B.: Linguistic input features improve neural machine translation. In: *Proceedings of the First Conference on Machine Translation, WMT 2016, collocated with ACL 2016*, 11–12 August, Berlin, Germany, pp. 83–91 (2016)
18. Sennrich, R., Haddow, B., Birch, A.: Controlling politeness in neural machine translation via side constraints. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 35–40 (2016)
19. Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Long Papers*, vol. 1, pp. 1715–1725 (2016)
20. Susanto, R.H., Chieu, H.L., Lu, W.: Learning to capitalize with character-level recurrent neural networks: an empirical study. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016*, Austin, Texas, USA, 1–4 November 2016, pp. 2090–2095 (2016)
21. Vaswani, A., et al.: Attention is all you need. In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, Long Beach, CA, USA, 4–9 December 2017, pp. 6000–6010 (2017)
22. Wang, X., Pham, H., Yin, P., Neubig, G.: A tree-based decoder for neural machine translation. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, 31 October–4 November 2018, pp. 4772–4777 (2018)
23. Wu, S., Zhang, D., Yang, N., Li, M., Zhou, M.: Sequence-to-dependency neural machine translation. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017*, Vancouver, Canada, 30 July–4 August, Long Papers, vol. 1, pp. 698–707 (2017)
24. Yang, X., Liu, Y., Xie, D., Wang, X., Balasubramanian, N.: Latent part-of-speech sequences for neural machine translation. CoRR abs/1908.11782 (2019)
25. Zhang, N., Li, X., Jin, X., Chen, W.: Joint prediction model of English words and their cases in neural machine translation. *J. Chin. Inf. Process.* **33**(3), 52–58 (2019)