applied sciences

MDPI

Article

Named Entity Correction in Neural Machine Translation Using the Attention Alignment Map

Jangwon Lee [1,2], Jungi Lee [3], Minho Lee [3] and Gil-Jin Jang [2,4,*]

[1] SK Holdings C&C, Gyeonggi-do, Suwon City 13558, Korea; saraitne11@naver.com
[2] School of Electronic and Electrical Engineering, Kyungpook National University, Daegu 41566, Korea
[3] Department of Artificial Intelligence, Kyungpook National University, Daegu 41566, Korea; darbams77@naver.com (J.L.); mholee@gmail.com (M.L.)
[4] School of Electronics Engineering, Kyungpook National University, Daegu 41566, Korea
* Correspondence: gjang@knu.ac.kr; Tel.: +82-53-950-5517

Featured Application: machine translation; information retrieval; text-to-speech.

Abstract: Neural machine translation (NMT) methods based on various artificial neural network models have shown remarkable performance in diverse tasks and have become mainstream for machine translation currently. Despite the recent successes of NMT applications, a predefined vocabulary is still required, meaning that it cannot cope with out-of-vocabulary (OOV) or rarely occurring words. In this paper, we propose a postprocessing method for correcting machine translation outputs using a named entity recognition (NER) model to overcome the problem of OOV words in NMT tasks. We use attention alignment mapping (AAM) between the named entities of input and output sentences, and mistranslated named entities are corrected using word look-up tables. The proposed method corrects named entities only, so it does not require retraining of existing NMT models. We carried out translation experiments on a Chinese-to-Korean translation task for Korean historical documents, and the evaluation results demonstrated that the proposed method improved the bilingual evaluation understudy (BLEU) score by 3.70 from the baseline.

Keywords: neural networks; recurrent neural networks; natural language processing; neural machine translation; named entity recognition

Citation: Lee, J.; Lee, J.; Lee, M.; Jang, G.-J. Named Entity Correction in Neural Machine Translation Using the Attention Alignment Map. *Appl. Sci.* **2021**, *11*, 7026. https://doi.org/10.3390/app11157026

Academic Editor: Arturo Montejo-Ráez

Received: 1 July 2021
Accepted: 26 July 2021
Published: 29 July 2021

1. Introduction

Neural machine translation (NMT) models based on artificial neural networks have shown successful results in comparison to traditional machine translation methods [1–5]. Traditional methods usually consist of sequential steps, such as morphological, syntactic, and semantic analyses. On the contrary, NMT aims to construct a single neural network and jointly train the entire system. Therefore, NMT requires less prior knowledge than traditional methods if a sufficient amount of training data is provided. Early NMT models, called sequence-to-sequence [6–9], are based on encoder–decoder architectures implemented with recurrent neural networks (RNNs) [10], such as long short-term memory (LSTM) [11] and the gated recurrent unit (GRU) [12]. The attention mechanism is usually used in RNN-based machine translation systems with variable lengths. The network generates an output vector, as well as its importance, called attention, to allow the decoder focus on the important part of the output [13–15]. Recently, a new NMT model called the transformer [16] has been proposed based on an attention mechanism with feedforward networks and without RNNs. Using the transformer, the learning time is reduced greatly with the help of non-RNN-type networks.

One of the problems in machine translation is the lack of training data. This problem was reported by Seljan [17] and Dunđer [18,19] for the problem of the automatic translation of poetry with a low-resource language pair. It was reported that the fluency and adequacy

of the translation results were skewed to higher scores. Especially for old literature translation where the machine translation is of great importance, obtaining reliable training data is much more difficult. The types of errors in the machine translation were extensively analyzed by Brkić [20]. They were wrong word mapping, omitted or surplus words, morphological and lexical errors, and syntactic errors such as word order and punctuation errors. There have been several methods to successfully solve these problems using transfer learning [21], contrastive learning [22], and open vocabularies [23].

Another major problem in NMT are out-of-vocabulary (OOV) words [24,25]. This is often called the rare word problem as well [26,27]. The words in the training dataset are converted into indices to the word dictionary or a predefined set of vectors, and a sequence of the converted numbers or vectors is used as an input to NMT systems. When a new word that is not in the dictionary is observed, the behavior of the trained network is unpredictable because there are no training sentences with the OOV words. It is almost impossible to include all of them in the dictionary because of the complexity limit for efficient translation. One of the solutions to this problem is subword tokenization using byte pair encoding (BPE) [27]. In this work, the unknown words are broken into reasonable subunits. Another solution is the unsupervised learning of the OOVs [28]. However, most of the OOV words are for named entities: human names, city names, and newly coined academic terms, and subword tokenization [27] and unsupervised learning [28] are not able to handle the named entities because they do not contain any meaningful information in them. As a solution, conventional systems use special labels for such OOV words (often as "UNK") and include them in the training data [24–26], so that the NMT model would distinguish them from ordinary words. Table 1 shows examples of translation outputs with an "UNK" symbol. The first named entity in the first example, "李周鎭," is mistranslated into "이진," although the expected output is "이주진." The second named entity, "元景淳," is not translated, but replaced with an "UNK" symbol because the true translation "원경순" is an OOV or rarely occurring word for the trained NMT model. Moreover, there are many similar cases in the subsequent named entities.

There have been several attempts to build open-vocabulary NMT models to deal with OOV words. Ling et al. [29] used a sub-LSTM layer that takes a sequence of characters to produce a word embedding vector. In the decoding process, another LSTM cell also generates words character-by-character. Luong and Manning [25] proposed a hybrid word–character model. This model adopts a sub-LSTM layer to use the information at the character level when it finds *unknown* words both in the encoding and decoding steps. Although character-based models show a translation quality comparable to word-based models and achieve open-vocabulary NMT, they require a huge amount of training time when compared with word-based models. This is because, if words are split into characters, their sequence lengths are increased to the number of characters, so the model complexity grows significantly. There are other approaches to use character-based models such as using convolutional neural networks [30,31]. However, it is hard to directly apply fully character-based models to Korean, because a Korean character is made by combining consonants and a vowel. Luong et al. [26] augmented a parallel corpus to allow NMT models to learn the alignments of "UNK" symbols between the input and output sentences. However, this method is difficult to apply to language pairs with extremely different structures, such as English–Chinese, English–Korean, and Chinese-to-Korean. Luong [26] and Jean [24] effectively addressed "UNK" symbols in translated sentence. However, mistranslated words, which often appear for rare input words, still were not considered.

In this work, we propose a postprocessing method that corrects mistranslated named entities in the target language using a named entity recognition (NER) model and an attention alignment mapping (AAM) between an input and an output sentence by using the attention mechanism (to the best of our knowledge, first proposed by Bahdanau et al. [13]). The proposed method can be directly applied to pretrained NMT models that use an attention mechanism by appending the postprocessing step to its output, without retraining the existing NMT models or modifying the parallel corpus. Our experiments on the Chinese-

to-Korean translation task of historical documents, the *Annals of the Joseon Dynasty* (http://sillok.history.go.kr/main/main.do, last access date: 1 July 2021) demonstrate that the proposed method is effective. In a numerical evaluation, the proposed method shows that the bilingual evaluation understudy (BLEU) score [32] was improved up to 3.70 compared to the baseline when the proposed method was not applied. Our work is available in a Git repository https://bitbucket.org/saraitne76/chn_nmt/src/master/, last access date: 1 July 2021).

Table 1. Examples of Chinese-to-Korean translation results with OOV words. Input and Truth: raw sentence pairs from the Chinese-to-Korean parallel corpus. English translation: translation of the Korean sentence to an English expression. NMT output: Korean translation results of a typical NMT model, with OOV words represented by the "UNK" symbol. The underlined words are named entities. Among those words, red-colored ones are human names; blue-colored ones are place names; green-colored ones are book names.

Input	以李周鎭爲平安監司, 元景淳爲副校理, 尹敬周爲正言。
Truth	이주진을 평안 감사로, 원경순을 부교리로, 윤경주를 정언으로 삼았다.
English Translation	Lee Joo Jin is assigned as the Pyeongan inspector, Won Kyung Soon as the vice dictator, Yun Gyeong Joo as the dictator.
NMT output	*이진을 평안 감사로, UNK을 부교리로, 윤주를 정언으로 삼았다.*
Input	分遣暗行御史李允明, 金夢臣, 李宇謙等, 廉察諸道。
Truth	암행어사 이윤명 · 김몽신 · 이우겸 등을 나누어 파견하여 여러 도를 검찰하게 하였다.
English Translation	The secret royal inspectors Lee Yun Myeong, Kim Mong Shin, and Lee Woo Gyeom were dispatched to investigate various provinces.
NMT output	암행어사 UNK · UNK · UNK 등을 나누어 보내어 두루 제도를 살피게 하였다.
Input	江原道 楊口縣民家九十九戶, 一時燒燼。道臣以聞, 上命行恤典。
Truth	강원도 양구현의 민가 99호가 한꺼번에 불타 없어졌는데, 도신이 계문하니, 임금이 휼전을 시행하라고 명하였다.
English Translation	99 civil houses in Yanggu Gangwon province were burnt down all at once, Do Shin requested the king to distribute food tickets to civilians.
NMT output	강원도 UNK민가 99호가 한꺼번에 불에타버렸다. 도주가 아뢰니, 상이 휼전을 행하라고 명하였다.
Input	上詣永禧殿展謁, 仍詣儲慶宮, 毓祥宮, 延祜宮, 宣禧宮展拜。
Truth	임금이 영희전에 나아가 전알하고, 이어서 저경궁 · 육상궁 · 연호궁 · 선희궁에 나아가 전배하였다.
English Translation	The king went to Yeonghuijeon and perform a rites, and then to Jeogyeonggung, Sokseonggung, Yeonhogung, and Seonhuigung and performed rites.
NMT output	임금이 영모전에 나아가서 전알하고, 이어서 경복궁 · UNK · UNK · 경희전에 나아가 참배하였다.
Input	行召對, 講≪名臣奏議≫。
Truth	소대를 행하고 ≪명신주의≫를 강론하였다.
English Translation	Conducted a So Dae and lectured on ≪Myungshinism≫.
NMT output	소대를 행하고 ≪UNK≫를 강하였다.

The remainder of the paper is organized as follows: Section 2 provides a review of the conventional machine translation and NER methods that are related to the proposed method. The named entity matching using the attention alignment map that forms the core of the current study is introduced in Section 3, along with the implementation details of the transformer and the proposed NER algorithm. Section 4 describes a series of experiments

that were carried out to evaluate the performance of the proposed NER method. In Section 5, the output of the NER results is further analyzed, and Section 6 concludes the paper.

2. Machine Translation

2.1. Neural Machine Translation

NMT maps a source sentence to a target sentence with neural networks. In a probabilistic representation, the NMT model is required to map a given source sentence $\mathbf{X} = [\mathbf{x}_1\ \mathbf{x}_2\ \cdots\ \mathbf{x}_n] \in \mathbb{B}^{v_s \times n}$ to a target sentence $\mathbf{Y} = [\mathbf{y}_1\ \mathbf{y}_2 \cdots\ \mathbf{y}_m] \in \mathbb{B}^{v_t \times m}$, where $\mathbb{B} = \{0, 1\}$, a binary domain space, v_s and v_t are the source (input) and target (output) vocabulary size, and n and m represent the sequence lengths of the input and output sentences, respectively. A vocabulary is usually defined by a set of tokens, which is a minimum processing unit for natural language processing (NLP) models. From a linguistic point of view, words or characters are the most popular mapping units for the tokens, depending on the grammar of the source and target languages. Each element of the encoding vector is assigned a positive integer index that uniquely identifies a single token in the corresponding vocabulary, so we can construct source vectors $\mathbf{x}_k \in \mathbb{B}^{v_s}$ by the following one-hot representation:

$$x_{k,i} = \begin{cases} 1 & \text{if } i = \text{index of } k\text{th token} \\ 0 & \text{otherwise} \end{cases}, \tag{1}$$

where $x_{k,i}$ is the ith element of $\mathbf{x}_k$. We can also construct a one-hot representation for the target vector $\mathbf{y}_k$ in a similar manner as well. The one-hot representation is extremely sparse, and the dimensions of input and target vector, n and m, may become too large to handle for the large vocabulary sizes. The embedding method, a general approach in natural language processing, is introduced to produce dense vector representations for the one-hot encoding vectors [33–36]. For given dimensions of the source and target, d_s and d_t with $d_s \ll v_s$ and $d_t \ll v_t$, linear embeddings from a higher dimensional binary space to a lower dimensions real domain are defined as follows:

$$\begin{aligned} \mathbf{E}^s \in \mathbb{R}^{d_s \times v_s}, \qquad & \mathbf{E}^t \in \mathbb{R}^{d_t \times v_t}, \\ \tilde{\mathbf{x}}_i = \mathbf{E}_s \mathbf{x}_i \in \mathbb{R}^{d_s}, \qquad & \tilde{\mathbf{y}}_j = \mathbf{E}_t \mathbf{y}_i \in \mathbb{R}^{d_t}, \end{aligned} \tag{2}$$

where $\mathbb{R}$ is the real number space and $\mathbf{E}^s$ and $\mathbf{E}^t$ are the source and target embedding matrices, respectively. Applying the linear embedding in (2), dense representation for the source and the target sentences are obtained by multiplying $\mathbf{E}_s$ and $\mathbf{E}_t$ to $\mathbf{X}$ and $\mathbf{Y}$,

$$\tilde{\mathbf{X}} = [\tilde{\mathbf{x}}_1\ \tilde{\mathbf{x}}_2\ \cdots \tilde{\mathbf{x}}_n] = \mathbf{E}_s \mathbf{X} \in \mathbb{R}^{d_s \times n} \tag{3}$$

$$\tilde{\mathbf{Y}} = [\tilde{\mathbf{y}}_1\ \tilde{\mathbf{y}}_2\ \cdots \tilde{\mathbf{y}}_n] = \mathbf{E}_t \mathbf{Y} \in \mathbb{R}^{d_t \times n}. \tag{4}$$

This linear transformation is one of the Word2Vec methods [33]. In our paper, we use this embedding for all the input and target one-hot vectors.

The target of machine translation is finding a mapping that maximizes the conditional probability $p(\mathbf{Y}|\mathbf{X})$. The direct approximation of $p(\mathbf{Y}|\mathbf{X})$ is intractable due to the high dimensionality, so most of recent NMT models are based on an encoder–decoder architecture [34]. The encoder reads an input sentence $\tilde{\mathbf{X}}$ in a dense representation and encodes it into an intermediate, contextual representation $\mathbf{C}$.

$$\mathbf{C} = \text{Encoder}(\tilde{\mathbf{X}}), \tag{5}$$

where "Encoder" is a neural network model for deriving contextual representation. After the encoding process, the decoder starts generating a translated sentence. At the first decoding step, it takes encoded contextual representation $\mathbf{C}$ and the "START" symbol, which means the start of the decoding process, and generates the first translated token. Second, the token generated previously is fed back into the decoder. It produces the next

token based on the tokens generated previously and contextual representation **C**. These decoding processes are conducted recursively until an "EOS" symbol is generated, which denotes the end of the sentence. The decoding process can be formulated by the following Markovian equation,

$$p(\mathbf{y}_j) \quad = \quad g(\{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{j-1}\}, \mathbf{C}), \tag{6}$$

where j is the symbol index to be generated, $\mathbf{y}_i$ is ith symbol, and $g(\cdot)$ is decoding step function, which generates a conditional probability if $\mathbf{y}_j$ given the previous outputs, $\{\mathbf{y}_1, \ldots, \mathbf{y}_{j-1}\}$, and the encoder output of the input sequence. Figures 1 and 2 illustrate the framework of the "sequence-to-sequence with attention mechanism" model (seq2seq) [13] and the framework of the "the transformer" model [16], which are NMT models used in the proposed methods.

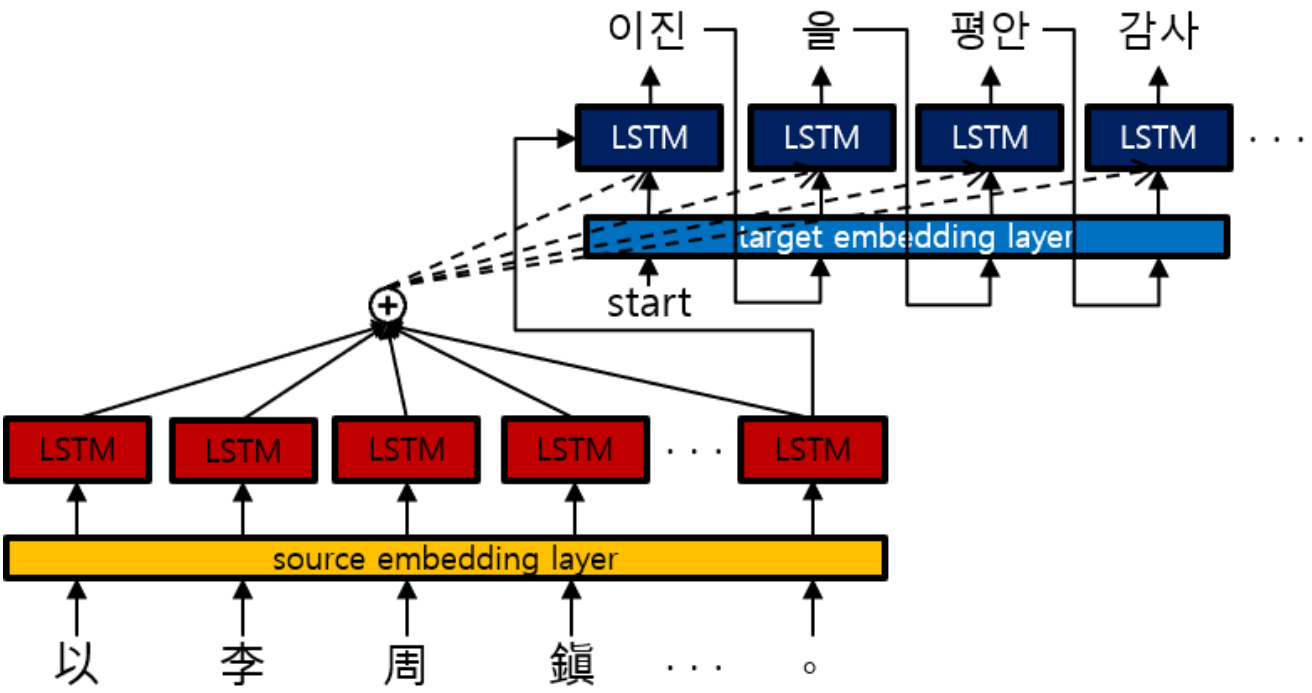


Figure 1. Framework of the seq2seq model.The red LSTM cell is the encoder and the blue is the decoder. The encoder encodes a source sentence into the context vector. The decoder is initialized with the context vector and generates a translated token, receiving a previous token and an attention vector. The translation results in this figure are given just to show that the input and output are Chinese text and Korean text, respectively. The Korean and the Chinese text do not have one-to-one correspondence.

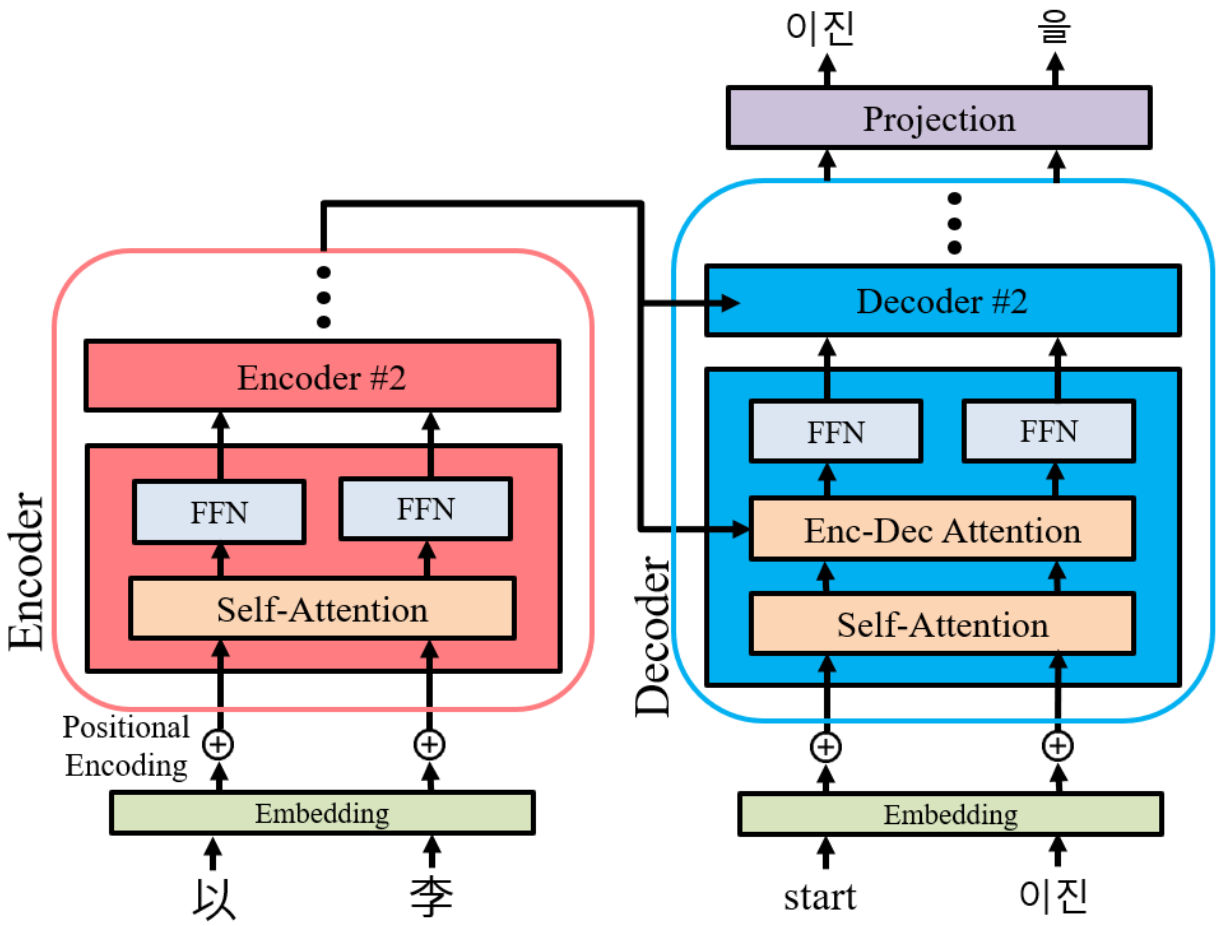


Figure 2. Framework of the transformer model. The red box is the encoder, and the blue one is the decoder. The outputs of the encoder are fed into the encoder–decoder attention layers. All residual connections and normalization layers are omitted.

2.2. Conventional Named Entity Recognition

There have been many studies for named entity recognition (NER) based on recurrent neural networks (RNNs) [37,38]. Similar to neural machine translation, the input is a sequence of tokens, $\tilde{\mathbf{X}} = [\tilde{\mathbf{x}}_1\ \tilde{\mathbf{x}}_2\ \cdots\ \tilde{\mathbf{x}}_n] \in \mathbb{R}^{v_s \times n}$, and the output is a sequence of binary labels indicating which tokens are named entities, so the length of the output is the same as that of the input sequence: $\mathbf{t} = [t_1\ t_2\ \cdots t_n] \in \mathbb{B}^n$. The example target encoding is shown in Table 1. Each word in the "Truth" and "NMT output" is underlined if it is a named entity. In those cases, the target labels are assigned one. The objective of named entity recognition is finding a sequence that maximizes the posterior probability of $\mathbf{t}$ given the input,

$$\mathbf{t}^* = \arg\max_{\mathbf{t}} p(\mathbf{t}|\mathbf{X}), \tag{7}$$

where $\mathbf{t}^*$ is an optimal NER result. Recently, a novel model for NER based only on attention mechanisms and feedforward networks achieved state-of-the-art performance on the CoNLL-2003 NER task [39].

3. Proposed Method

3.1. AAM in Sequence-to-Sequence Models

In this subsection, we describe the seq2seq model [13] used in our method and explain how to obtain an AAM from it. The seq2seq model consists of an LSTM-based [11] encoder–decoder and an attention mechanism. The encoder encodes a sequence of input tokens $\tilde{\mathbf{x}} = (\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n)$, represented as dense vectors, into a context vector c, which is a fixed-

length vector. We used a bidirectional LSTM (BiLSTM) [40–43] as the encoder to capture bidirectional context information of the input sentences.

$$\overrightarrow{h_i} = f(\tilde{x}_i, \overrightarrow{h_{i-1}}), \tag{8}$$

$$\overleftarrow{h_i} = f(\tilde{x}_i, \overleftarrow{h_{i+1}}), \tag{9}$$

where f are stacked unidirectional LSTM cells and $\overrightarrow{h_i} \in \mathbb{R}^d$ and $\overleftarrow{h_i} \in \mathbb{R}^d$ are the hidden states of the top forward LSTM cell and the top backward cell, respectively. Moreover, i indicates the encoding steps, and d is the number of hidden units of the top LSTM cell in the encoder.

$$c = [\overrightarrow{h_n}; \overleftarrow{h_1}] \tag{10}$$

Hidden states at the last encoding step for both directions $\overrightarrow{h_n}$ and $\overleftarrow{h_1}$ are concatenated to obtain $c \in \mathbb{R}^{2d}$. Stacked unidirectional LSTM cells are used for the decoder. Once the encoder produces context vector c, the bottom LSTM cell of the decoder is initialized with c.

$$s_{j=1} = c, \tag{11}$$

where $s_j \in \mathbb{R}^{2d}$ are the hidden states of the bottom LSTM cell in the decoder, and subscript j denotes the index of the decoding step. Next, the decoder starts the process of decoding:

$$p(y_j) = g(y_{j-1},\ s_j,\ o_j)\,. \tag{12}$$

Each decoding step computes the probability of the next token using three components. The first is the previously generated token y_{j-1}; the second is the current hidden state s_j; the third is an attention output vector o_j. An attention output allows the decoder to retrieve hidden states of encoder $\{h_1, \cdots, h_n\}$, where $h_i = [\overrightarrow{h_i}; \overleftarrow{h_i}]$.

$$o_j = \sum_{i=1}^{n} a_{ij} h_i\,, \tag{13}$$

$$a_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{n} exp(e_{kj})}\,, \tag{14}$$

$$e_{ij} = v_a^T \tanh(W^a s_{j-1} + U^a h_i + b^a)\,. \tag{15}$$

Attention scores e_{ij} indicate how related s_{j-1} is to h_i. Here, $W^a \in \mathbb{R}^{d_a \times 2d}$, $U^a \in \mathbb{R}^{d_a \times 2d}$, $v_a \in \mathbb{R}^{d_a}$, and $b^a \in \mathbb{R}^{d_a}$ are trainable parameters, where d_a is the hidden dimension of the attention mechanism. Attention weights are computed by the softmax function across the attention scores. The attention output o_j is a weighted sum of hidden states from the encoder $\{h_1, \cdots, h_n\}$. It tells the decoder where to focus on the input sentence when the decoder generates the next token.

In the seq2seq model, an attention alignment map (AAM) $\mathbf{A} \in \mathbb{R}^{n \times m}$, where n and m represent the sequence lengths of the input and output sentences, can be easily computed by stacking up the results of (14) while the model generates the translation. Figure 1 illustrates the framework of the seq2seq model used in this paper.

3.2. Transformer

In this subsection, we describe the transformer model [16] used in our method and explain how to obtain an AAM from it. The transformer also consists of an encoder and a decoder. Unlike the seq2seq model, it introduces a position encoding [16,44] to add

positional information into the model, because it does not have any recurrent units that would model positional information automatically.

$$x'_i = \tilde{x}_i + PE(i)\,, \tag{16}$$
$$y'_j = \tilde{y}_j + PE(j)\,. \tag{17}$$

Here, $PE(k) \in \mathbb{R}^{d_{model}}$ produces a position encoding vector that corresponds to position k, d_{model} is the dimensionality of the model, and i and j are the positional indices of the input and output sentence, respectively. The positional encoding vectors are added to a sequence of input tokens $\tilde{\mathbf{x}} = (\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n)$ represented as dense vectors as in (2). The sum of embedding vectors and positional encoding $\mathbf{x}' = (x'_1, x'_2, \ldots, x'_n) \in \mathbb{R}^{d_{model} \times n}$ are fed into the bottom encoding layer.

The encoder is a stack of encoding layers, where each encoding layer is composed of a self-attention layer and a feedforward layer. The self-attention layer of the bottom encoding layer takes $\mathbf{x}'$, and the others receive the outputs of the encoding layer right below them. Self-attention layers allow the model to refer to other tokens in the input sequence.

$$A_h^{enc} = \text{softmax}\left(\frac{Q_h K_h^T}{\sqrt{d_k}}\right) \tag{18}$$
$$\text{Attention}(Q_h, K_h, V_h) = A_h^{enc} V_h \tag{19}$$
$$\text{head}_h = \text{Attention}(Q_h, K_h, V_h) \tag{20}$$
$$\begin{aligned}&\text{MultiHead}(Q, K, V) = \\ &\qquad \text{Concat}(\text{head}_1, \cdots, \text{head}_{h_n}) W^O\,.\end{aligned} \tag{21}$$

The "multihead scaled dot-product attention" is computed by the above equations and was proposed by Vaswani et al. [16]. Here, Q_h, K_h, and V_h are linear transformations of its input. $Q_h = \mathbf{z_e^T} \cdot W_h^Q$, $K_h = \mathbf{z_e^T} \cdot W_h^K$, and $V_h = \mathbf{z_e^T} \cdot W_h^V$, where $\mathbf{z_e} \in \mathbb{R}^{d_{model} \times n}$ is the input of each attention layer in the encoder. A_h^{enc} denotes an AAM of the encoder self-attention layer on the hth head. Moreover, $W_h^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W_h^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_h^V \in \mathbb{R}^{d_{model} \times d_v}$, and $W^O \in \mathbb{R}^{h_n d_v \times d_{model}}$ are trainable parameters, and $d_k = d_v = d_{model}/h_n$, where h_n is the number of heads.

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2\,. \tag{22}$$

The outputs of the self-attention layers pass through the feedforward network. Each position is processed independently and identically. Here, $W_1 \in \mathbb{R}^{d_{model} \times d_{ff}}$, $W_2 \in \mathbb{R}^{d_{ff} \times d_{model}}$, $b_1 \in \mathbb{R}^{d_{ff}}$, and $b_2 \in \mathbb{R}^{d_{model}}$ are learnable parameters, where d_{ff} is the dimensionality of the inner linear transformation.

The final output of the encoder is considered as contextual representation $\mathbf{c} = (c_1, c_2, \ldots, c_n) \in \mathbb{R}^{d_{model} \times n}$ as in (5). It is fed into the encoder–decoder attention layers of the decoder. The decoder has a stack of decoding layers, where each decoding layer consists of a self-attention layer, encoder–decoder attention, and a feedforward network. By analogy to the encoder, the bottom decoding layer takes the sum of embedding vectors and positional encoding $\mathbf{y}' = (y'_1, y'_2, \ldots, y'_m) \in \mathbb{R}^{d_{model} \times m}$, as in (17), and the others receive the outputs of the decoding layer right below them. Self-attention layers in the decoder are similar to those in the encoder. However, the model can only retrieve the earlier positions at the current step. Hence, the model cannot attend to tokens not yet generated in the prediction phase. An encoder–decoder attention layer receives contextual representation

c and the output of self-attention layers located below in the decoder $\mathbf{z}_\mathbf{d}^T \in \mathbb{R}^{d_{model} \times m}$ as in Figure 2.

$$Q_h = \mathbf{z}_\mathbf{d}^T W_h^Q \tag{23}$$

$$K_h = \mathbf{c}^T W_h^K \tag{24}$$

$$V_h = \mathbf{c}^T W_h^V \tag{25}$$

$$A_h^{enc-dec} = \text{softmax}\left(\frac{Q_h K_h^T}{\sqrt{d_k}}\right) \tag{26}$$

$$\text{Attention}(Q_h, K_h, V_h) = A_h^{enc-dec} V_h\,. \tag{27}$$

This layer helps the decoder concentrate on the proper context in an input sequence when the decoder generates the next token. For every sublayer, residual connection [45] and layer normalization [46] are applied. Although we did not annotate layer indices for the trainable parameters, each layer does not share them. There are h_n encoder–decoder AAMs $A_h^{enc-dec}$ for each decoding layer. To obtain $\mathbf{A} \in \mathbb{R}^{n \times m}$, we reduced the mean across layers l and attention heads h.

$$\mathbf{A} = \underset{l}{\text{mean}}(\underset{h}{\text{mean}}(A_{hl}^{enc-dec})) \tag{28}$$

Figure 2 illustrates the framework of the transformer model in our study. The input Chinese characters, "以李", if directly translated, can be mapped to Korean "이을". However, due to the embedding in the decoder with the context information, the output of the transformer becomes "이진을".

3.3. NER Model

The detection of named entities of the input Chinese sentence is required to improve the quality of the translation. The NER model used in our study was based on stacked BiLSTM [40–42] and the conditional random field (CRF) [47,48]. We considered each Chinese character as a token and assigned a tag to each token. The tagging scheme was the IOB format [47]. As shown in Figure 3, to each of the input characters, it was given a label that was composed of one or two tags according to the membership of the input character to the named entities. The first tag is one of I, O, or B, for the inside, outside, or beginning of named entity words, respectively. The I-*tag* denotes the inside part of the named entity, but not the first character. The B-*tag* is the beginning character of the named entity. The O-*tag* means that a corresponding character is not inside a named entity. In our implementation, there were 4 types of named entities: *Person*, *Location*, *Book*, and *Era*. This type of information corresponds to B-*tag* and I-*tag*. Therefore, the NER model is asked to assign one of the nine tags to each token.

$$t_i \in \{\text{BP, BL, BB, BE, IP, IL, IB, IE, O}\}\,, \tag{29}$$

where BP, BL, BB, and BE are B-*tags* for *Person*, *Location*, *Book* and *Era*, respectively, and IP, IL, IB, and IE are I-*tags* for the same 4 named entity types. Table 2 shows an example of the input and output of the NER model. The NER model receives n Chinese tokens (characters) and predicts n named entity tags. A named entity "楊口縣" can be extracted by taking characters from the Chinese input from the index of B-Location to the index of the last I-Location. To separate the consecutive named entities, we used B-*tag* and I-*tag* together. If we only classify whether a character is within a named entity or not, it is impossible to separate "江原道楊口縣" into "江原道" and "楊口縣."

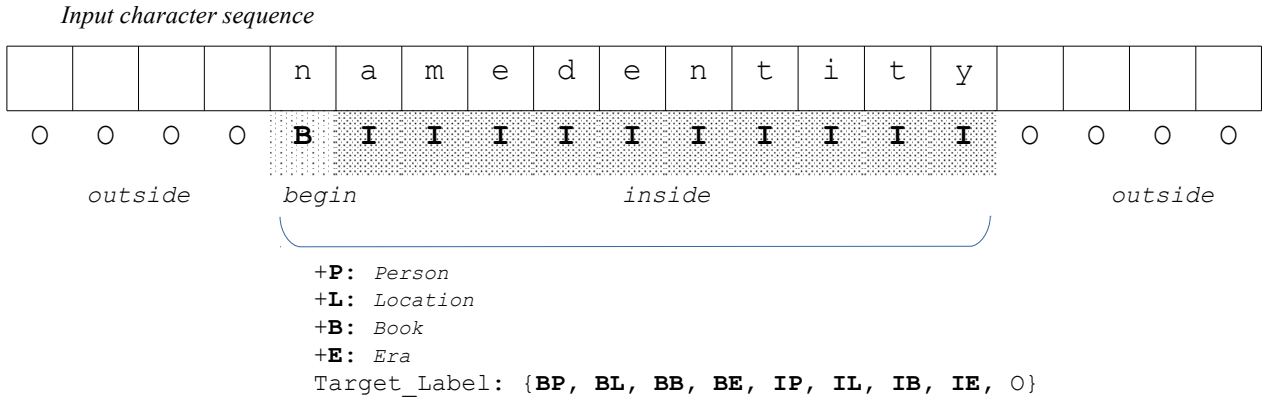


Figure 3. Target labeling of the NER model. All the individual input characters are assigned a target label of one or two tags. Characters not belonging to named entities are labeled by a single tag ''O'', meaning "outside" of the named entities. The first character of a named entity word is assigned the ''B''-tag, whose meaning is "beginning" of the named entity. All the other characters of the named entity word are assigned the ''I''-tag ("inside"). To each of the first tags of the the named entities, ''B'' and ''I'', an extra tag from {P, L, B, E} is concatenated according to the types of the named entities, {Place, Location, Book, Era}, respectively.

Table 2. Example of an input and an output of the NER model. Input: Chinese sentences that are fed into the NER model. The underlined words are named entities. Among those, human names are red; place names are blue. Output: named entity tags for each Chinese character that should be predicted by the model. BL and IL are the B-*tag* and I-*tag* for the *Location* named entity occurrence, and BP and IP are for the *Person* named entities, while O is for *Outside*.

Input	江原道 楊口縣民家九十九戶, 一時燒燼。
Output	BL IL IL BL IL IL O O O O O O O O O O O O
Input	道臣以聞, 上命行恤典。
Output	BP IP O O O O O O O O O

As in the NMT model, a Chinese sentence $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^{v_s \times n}$ represented as one-hot encoding vectors is converted into dense vector representations $\tilde{\mathbf{x}} = (\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n) \in \mathbb{R}^{d_s \times n}$ by using the embedding method [33–36] as in (2). Next, $\mathbf{x}$ are fed into the BiLSTM sequentially, and the BiLSTM captures bidirectional contextual information from input sequence $\mathbf{x}$, as in (8) and (9).

$$\begin{aligned} h_i &= [\overrightarrow{h_i}; \overleftarrow{h_i}] \\ p(t_i|x_i) &= \mathrm{CRF}(W^N h_i + b^N). \end{aligned} \tag{30}$$

Hidden state $h_i \in \mathbb{R}^{2d}$, which is the output of BiLSTM, is a concatenation of both directional LSTM hidden states $\overrightarrow{h_i} \in \mathbb{R}^d$ and $\overleftarrow{h_i} \in \mathbb{R}^d$. A linear transformation layer and a CRF [47,48] layer are applied to $\mathbf{h} = (h_1, h_2, \ldots, h_n)$, and the CRF layer predicts named entity tags for each input token x_i, where i is the time step of tokens and d is the number of hidden units of a top LSTM cell. Here, $W^N \in \mathbb{R}^{2d \times d_t}$ and $b^N \in \mathbb{R}^{d_t}$ are trainable parameters, where $d_t = 9$ is the number of tag classes.

Finally, we can extract a list of named entities from the combination between the input sentence and the predicted tags. Figure 4 illustrates the NER framework used in our study.

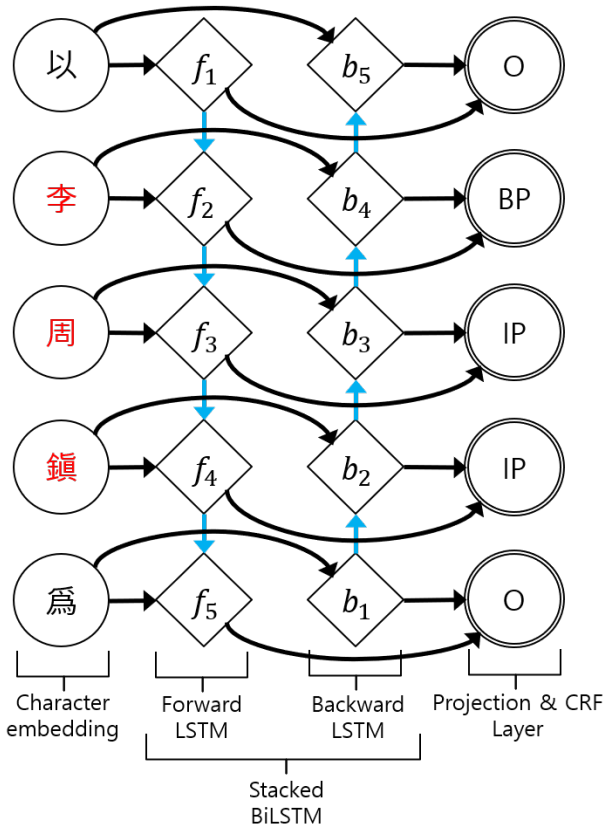


Figure 4. Framework of the NER model. The model predicts tags for each Chinese character. f_i and b_i represent the forward and backward LSTM cells, and i indicates the time steps. BP and IP are the B-*tag* and I-*tag* for the *Person* named entity occurrence, and O is for *Outside*.

3.4. Named Entity Correction with AAM

In Table 1, we can see that mistranslated words in the output of the NMT model correspond to named entities in the input sentences. The reason for this is that these named entities are OOV words or rarely occur in the training corpus. The NMT system cannot model these named entities well. In this section, we describe the proposed method that corrects mistranslated words in the output sentences through an example.

First, the NMT model translates a given Chinese sentence to a Korean sentence. In Table 3, it cannot accurately predict named entities that are names of persons.

Table 3. Neural machine translation. English translation is to explain the meaning of the text.

Input	以李周鎭爲平安監司, 元景淳爲副校理, 尹敬周爲正言。
Output	이진을 평안 감사로, UNK을 부교리로, 윤주를 정언으로 삼았다.
English	Lee Joo Jin is assigned as the Pyeongan inspector, Won Kyung Soon as the vice dictator, Yun Gyeong Joo as the dictator.

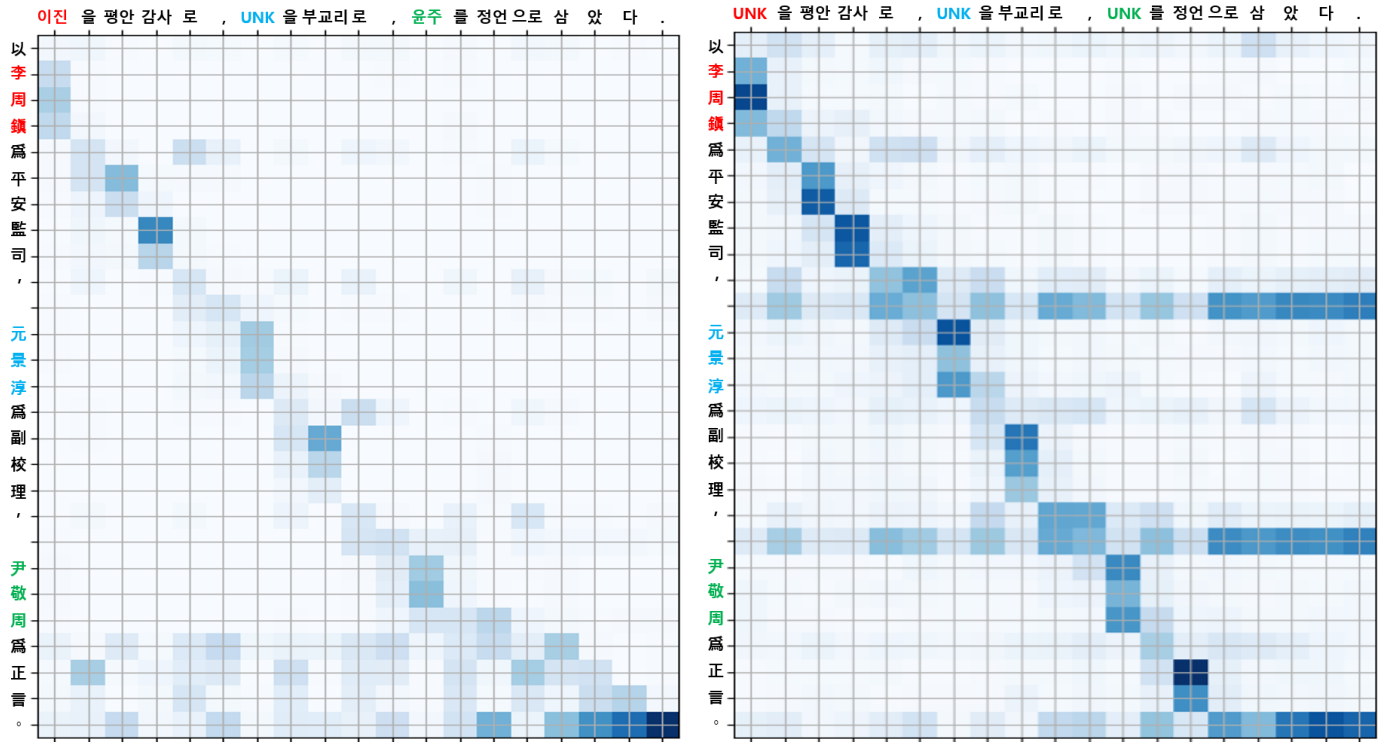
Second, the NER model finds named entities in the given Chinese sentence. In Table 4, red-colored words denote the named entities found by the NER model.

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix} \tag{31}$$

Third, we computed AAM $\mathbf{A} \in \mathbb{R}^{n \times m}$ from the NMT model, using (14) and (28). Here, n and m are the sequence length of the input and output sentence, respectively. Figure 5 shows examples of the attention alignment map. Each element a_{ij} of $\mathbf{A}$ is the amount of related information between input token x_i and output token y_j.

Table 4. Named entity recognition. The detected named entities of human names are underlined colored red.

Input	以李周鎭爲平安監司, 元景淳爲副校理, 尹敬周爲正言。
Output	以李周鎭爲平安監司, 元景淳爲副校理, 尹敬周爲正言。



(**a**) Attention alignment map from the seq2seq model.

(**b**) Attention alignment map from the transformer model.

Figure 5. Attention alignment maps. Labels of columns and rows correspond to the tokens in the input sentences (Chinese) and the output sentences (Korean), respectively. The postprocessing has not yet been applied to the output. Colored tokens on the input side are the named entities predicted by the NER model. Colored tokens on the output side are aligned with equally colored named entities on the input side by AAM.

Fourth, we took the row vectors of AAM corresponding to indices of the Chinese named entities. Figure 6 illustrates a part of the AAM. In this example, the indices of the

Chinese named entities "李周鎭" are 2, 3, 4, so we took row vectors $\mathbf{a}_2$, $\mathbf{a}_3$ and $\mathbf{a}_4$, where $\mathbf{a}_2 = (a_{21}, a_{22}, \ldots, a_{2m})$.

$$\hat{j} = \arg\max_{j} \sum_{i \in \{2,3,4\}} \mathbf{a}_i \tag{32}$$

Fifth, summation across the columns of $\mathbf{a}_2$, $\mathbf{a}_3$ and $\mathbf{a}_4$ was implemented to obtain the vector form. The index of the Korean token aligned with the Chinese named entity was found by the arg max function, where $\hat{j}$ is the index of Korean token "이진" aligned with Chinese named entity "李周鎭." The NER matching results are shown in Table 5.

Repeating the above process, we can align all Chinese named entities found by the NER model with the Korean tokens in the sentence translated by the NMT model.

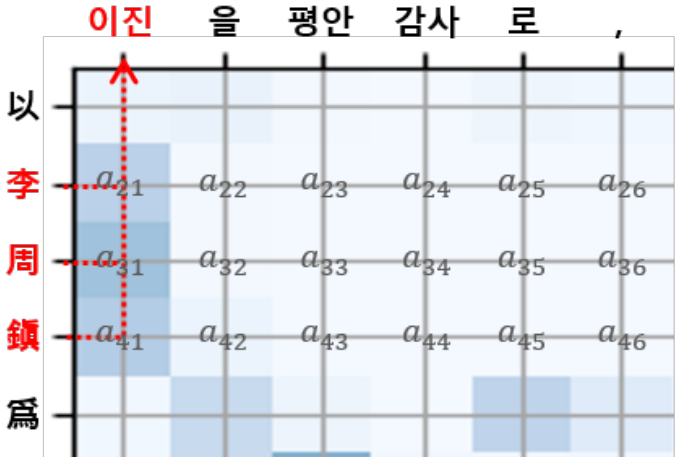


Figure 6. Korean tokens aligned with the Chinese named entities.

Table 5. Korean tokens aligned with the Chinese named entities. The underlined words are named entities. Among those words, red-colored ones are human names; blue-colored ones are place names; green-colored ones are book names.

Input	以李周鎭(1)爲平安監司, 元景淳(2)爲副校理, 尹敬周(3)爲正言。
Output	이진(1)을 평안 감사로, UNK(2)을 부교리로, 윤주(3)를 정언으로 삼았다.

We assumed that Korean token $y_{\hat{j}}$ was mistranslated. Finally, the aligned Korean tokens were replaced with a direct translation of the corresponding Chinese named entities from the look-up table. If the look-up table does not have the named entity, an identity copy of the Chinese named entity is an appropriate alternative. Figure 7 shows correction of the named entities in the translation results using look-up table. The corrections are: "이진(Lee Jin)" ⇒ "이주진(Lee Joo Jin)", "UNK" ⇒ "원경순(Won Kyung Soon)", and "윤주(Yoon Joo)" ⇒ "윤경주(Yoon Kyung Joo)". The subscripted, parenthesized numbers are found by the proposed method.

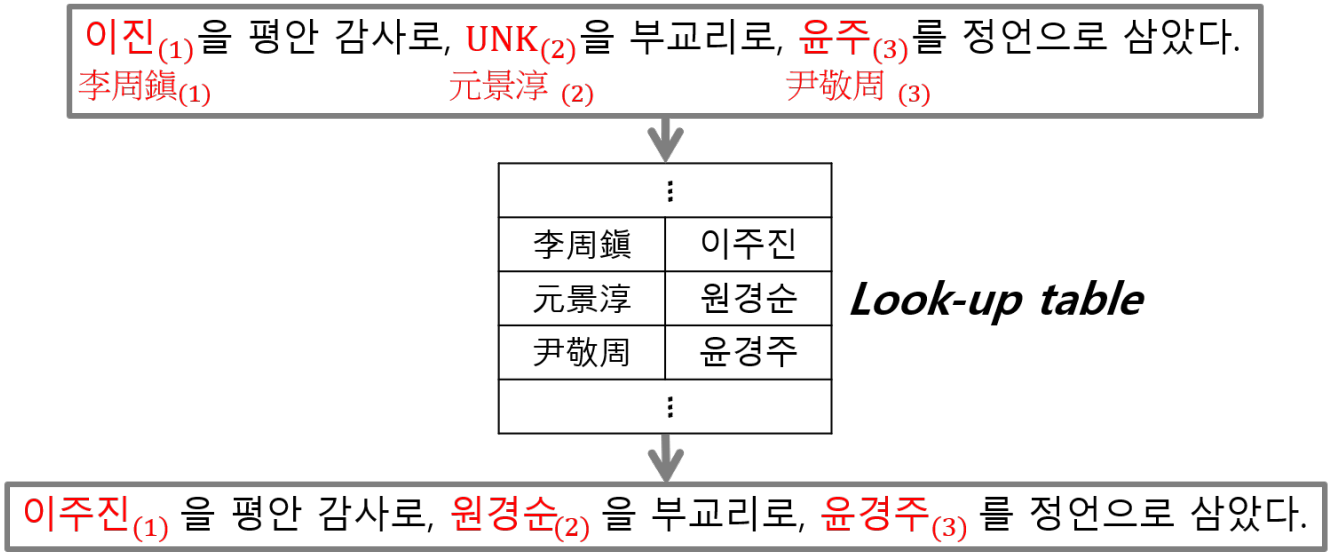


Figure 7. Named entity correction using the look-up table. The named entity, "이진(Lee Jin)" is corrected to "이주진(Lee Joo Jin)", "UNK" is to "원경순(Won Kyung Soon)", and "윤주(Yoon Joo)" to "윤경주(Yoon Kyung Joo)". All the named entities in this example are person names.

4. Experiments

We evaluated our approach on the Chinese-to-Korean translation task. The *Annals of the Joseon Dynasty* were used for our experiments as a parallel corpus. We compared the results for two cases: when the postprocessing was applied and when it was not applied.

4.1. Dataset: The Annals of the Joseon Dynasty

The *Annals* were written by the Joseon Dynasty of Korea in 1413–1865 and are listed in UNESCO's Memory of the World Registry. The *Annals* have been digitalized by the government of Korea since 2006 and are available on the website (http://sillok.history.go.kr/main/main.do, last access date: 1 July 2021) with the Korean translations and the original texts in Chinese. We used this parallel corpus to train our NMT models. To simulate real-world situations, we split the records according to the time they were written. Records from 1413 to 1623 were the training corpus, and records from 1623 to 1865 were the evaluation corpus. The training and evaluation corpus contained 230 K and 148 K parallel articles, respectively. We only used articles with Chinese and Korean tokens less than 200 in length, because the articles have an extremely variable length of letters. Figure 8 shows histograms for the sequence length of the Korean–Chinese parallel corpus. The Chinese–Korean pair sequences with the top 5% length were ignored in histogram Figure 8. For all Chinese sentences, the mean sequence length was 112.87 and the median was 54. For all Korean sentences, the mean was 124.56 and the median was 56. In Chinese (input), no tokenization was used. We simply split each Chinese sentence into a sequence of characters, because each Chinese character has its own meaning. In Korean (output), meanwhile, we used an explicit segmentation method [49] to split each Korean sentence into a sequence of tokens. Thus, the number of articles for training was 168 K and for evaluation was 113 K.

For the NER model, we also used the same corpus: the *Annals of the Joseon Dynasty*. The annotation of the Chinese named entities for this corpus is publicly available (https://www.data.go.kr/dataset/3071310/fileData.do, last access date: 1 July 2021). Additionally, Table 6 shows an analysis of the Chinese NER corpus. Approximately 7.5% of the characters belong to named entities, and the most frequently named entity type is *Person*.

Table 6. Analysis of the Chinese NER corpus.

# total characters			66M
# characters within the named entities			5M
# types of named entities			140K
Ratio of the named entity types			
Person	Location	Book	Era
73.3%	24.0%	2.4%	0.3%

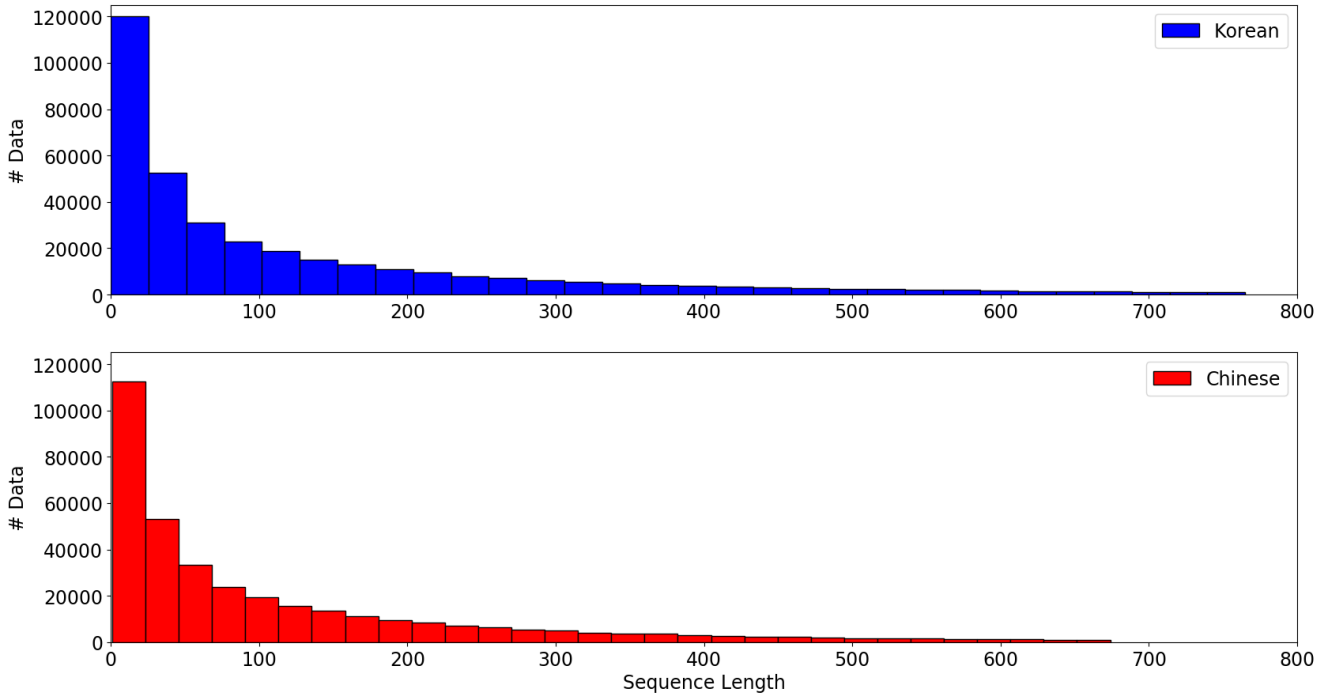


Figure 8. Histograms for the sequence length of the Korean–Chinese parallel corpus.

4.2. Models

For the seq2seq model, *Description* in Table 7 describes the model architecture used in our experiments. For the seq2seq model, *Description* means (embedding size, hidden units of encoder cells, # stack of encoder cells, hidden units of decoder cells, # stack of decoder cells). Embedding matrices for the source and target tokens were both pretrained by the word2vec algorithm [50], using only the training parallel corpus. The encoder is a stacked BiLSTM, and the decoder is a stacked unidirectional LSTM. During the learning process, the dropout approach [51,52] was applied to the output and states of the LSTM cells. Once training of the model was complete, beam-search decoding was used with a beam width of four to generate a translation that maximized the sum of the conditional probabilities of the sequence.

For the transformer model, *Description* in Table 7 represents (hidden size, # hidden layers, # heads, FFN filter size). To avoid overfitting of the model, the dropout [52] method was used among the layers in the training process. As the seq2seq model, beam-search decoding was implemented with a beam width of four. The NER model used in this study was the BiLSTM-CRF model. Specifically, the following model was used in our experiments. The embedding size was 500, and the embedding matrix was pretrained by the word2vec algorithm [50] using only the training dataset. Each cell had five-hundred twelve hidden units, and two cells are stacked. Here, we also used the dropout approach [51,52] in the learning phase.

Table 7. Performance improvements in the BLEU score. *Description*: model details. *Vocab*: the number of vocabularies. *Params*: the number of model parameters. *Original*: BLEU score without the proposed method. *Modified*: BLEU score for the results corrected by the proposed method. The best scores for both "Original" and "Modified" are achieved by using the second configuration of Seq2seq model, and those numbers are emphasized in bold face.

Model	Description	Vocab	Params	Original	Modified	Δ
Seq2seq	(500, 512, 3, 1024, 2)	40K	58M	35.75	39.29	+3.54
Seq2seq	(500, 512, 3, 1024, 2)	42K	59M	**35.83**	**39.53**	**+3.70**
Seq2seq	(500, 512, 3, 1024, 2)	50K	63M	35.66	39.13	+3.47
Seq2seq	(500, 512, 3, 1024, 2)	87K	81M	35.29	37.89	+2.60
Seq2seq-Reduced	(300, 256, 3, 512, 2)	42K	22M	33.95	37.26	+3.31
Seq2seq-Reduced	(300, 256, 3, 512, 2)	87K	54M	33.73	36.59	+2.86
Transformer-Big	(512, 6, 8, 2048)	42K	65M	33.90	37.07	+3.17
Transformer-Big	(512, 6, 8, 2048)	87K	88M	32.66	35.62	+2.96
Transformer	(256, 3, 4, 1024)	42K	16M	34.68	37.79	+3.11
Transformer	(256, 3, 4, 1024)	87K	27M	34.95	37.98	+3.03
Transformer-Reduced	(128, 2, 2, 256)	42K	6M	30.61	33.52	+2.91

4.3. Experimental Results

To evaluate our NER models, we introduced two types of F1-score: entity form and surface form [53]. First, the entity form is a conventional measurement calculated from the entity level. Second, the surface form evaluates the ability of NER models to find rare entity words. In Table 8, the lexicon used in *Dictionary search* was extracted only from the training corpus. The NER model used in the experiment was a two-stack LSTM model. Table 7 shows how the performance improved in the proposed method depending on the type of NMT model (seq2seq or transformer), the number of trainable parameters of the model, and the output (Korean) vocabulary size. Our experiments showed that the proposed method was effective regardless of these types, and the BLEU scores improved from 2.60 to 3.70. In Table 9, experimental results show that the proposed approach successfully corrected mistranslated named entities in the output of the NMT model.

Table 8. NER accuracy in 2 types of F1-score. *Entity Form* and *Surface Form* mean how many entities the model finds and how many types of entities the model finds, respectively.

Model	Entity Form	Surface Form
Dictionary search	4.4%	35.0%
1-layer LSTM Stack	91.1%	88.0%
2-layers LSTM Stack	91.9%	88.5%
3-layers LSTM Stack	91.8%	88.2%

Table 9. Named entity correction using the proposed method. *Baseline*: outputs of the seq2seq model. *Proposed*: results of our approach. Named entities are underlined. Human names are in red color; place names in blue; book names in green.

Truth	이주진을 평안 감사로, 원경순을 부교리로, 윤경주를 정언으로 삼았다.
English Translation	Lee Joo Jin is assigned as the Pyeongan inspector, Won Kyung Soon as the vice dictator, Yun Gyeong Joo as the dictator.
Baseline	이진을 평안 감사로, UNK을 부교리로, 윤주를 정언으로 삼았다.
Proposed	이주진을 평안 감사로, 원경순을 부교리로, 윤경주를 정언으로 삼았다.
Truth	암행어사 이윤명 · 김몽신 · 이우겸 등을 나누어 파견하여 여러 도를 검찰하게 하였다.
English Translation	The secret royal inspectors Lee Yun Myeong, Kim Mong Shin, and Lee Woo Gyeom were dispatched to investigate various provinces.
Baseline	암행어사 UNK · UNK · UNK 등을 나누어 보내어 두루 제도를 살피게 하였다.
Proposed	암행어사 이윤명 · 김몽신 · 이우겸 등을 나누어 보내어 두루 제도를 살피게 하였다.
Truth	강원도 양구현의 민가 99호가 한꺼번에 불타 없어졌는데, 도신이 계문하니, 임금이 휼전을 시행하라고 명하였다.
English Translation	99 civil houses in Yanggu Gangwon province were burnt down all at once, Do Shin requested the king to distribute food tickets to civilians.
Baseline	강원도 UNK민가 99호가 한꺼번에 불에타버렸다. 도주가 아뢰니, 상이 휼전을 행하라고 명하였다.
Proposed	강원도 양구현민가 99호가 한꺼번에 불에타버렸다. 도신가 아뢰니, 상이 휼전을 행하라고 명하였다.
Truth	임금이 영희전에 나아가 전알하고, 이어서 저경궁 · 육상궁 · 연호궁 · 선희궁에 나아가 전배하였다.
English Translation	The king went to Yeonghuijeon and perform a rites, and then to Jeogyeonggung, Sokseonggung, Yeonhogung, and Seonhuigung and performed rites.
Baseline	임금이 영모전에 나아가서 전알하고, 이어서 경복궁 · UNK · UNK · 경희전에 나아가 참배하였다.
Proposed	임금이 영희전에 나아가서 전알하고, 이어서 저경궁 · 육상궁 · 연호궁 · 선희궁에 나아가 참배하였다.
Truth	소대를 행하고 ≪명신주의≫를 강론하였다.
English Translation	Conducted a So Dae and lectured on ≪Myungshinism≫.
Baseline	소대를 행하고 ≪UNK≫를 강하였다.
Proposed	소대를 행하고 ≪명신주의≫를 강하였다.

5. Discussion

We found that the proposed method had several strengths and weaknesses. As for the strengths, the proposed method does not require retraining of the existing NMT models, and it can be directly applied to the NMT models without modifying the model architecture. It is suitable for any language pair. Moreover, it has a low computational complexity because of the small-sized vocabulary. As for the weaknesses, the proposed method does not work when predictions of the NER model are wrong. Additionally, tokens that should not be changed may be corrected if the alignment is not proper. The proposed method needs a look-up table to work properly. Table 10 shows examples of these weaknesses. In the above example, the NER model cannot find a named entity in the source sentence, so the *UNK* for "거려청" was not corrected. The *UNK* for "≪심경≫" was also not corrected, although the NER model recognized a token "≪必經≫" as a named entity, because our look-up table did not have "≪必經≫." In the final example, token "하" in *Baseline* was changed because the attention alignment map was not accurate.

Table 10. Weaknesses of the proposed method. *Source*: input sentence. *Truth*: ground truth. *Baseline*: outputs of the NMT models. *Proposed*: results of our approach. *NER output*: outputs of the NER model. Underlined words: named entities. Green-colored tokens: named entities for the names of books. Blue-colored: named entities for the names of place names.

Source	上御居廬廳, 召對, 命儒臣, 讀≪必經≫。
Truth	임금이 거려청에 나아가 소대하였다. 임금이 유신들에게 명하여 ≪심경≫을 읽게하였다.
English Translation	The king went to Georyeocheong and conducted a So Dae. The king ordered the subjects to read ≪Shim Gyung≫.
Baseline	상이 UNK에 나아가 소대하였다. 유신에게 명하여 ≪UNK≫을 읽게하였다.
Proposed	상이 UNK에 나아가 소대하였다. 유신에게 명하여 ≪UNK≫을 읽게하였다.
NER output	[≪必經≫, Book]
Source	進講于熙政堂。
Truth	희정당에서 진강하였다.
English Translation	He lectured at Huijeongdang.
Baseline	UNK에서 진강하였다.
Proposed	UNK에서 진강희정당였다.
NER output	[熙政堂, Location]

6. Conclusions

Even the NMT models that show state-of-the-art performance on multiple machine translation tasks are still limited when dealing with OOV and rarely occurring words. We found that the problem is particularly relevant for the translation of historical documents with multiple named entities. In this paper, we proposed a postprocessing approach to address this limitation. The proposed method corrects the machine translation output using the NER model and the attention map. The NER model finds named entities in the source sentence, and the attention map aligns the located named entities with the tokens in the translated sentence. Next, we assumed that the tokens aligned with the source named entities were mistranslated, and we replaced them using the look-up table or an identity copy. Experiments with various target vocabulary sizes in Section 4 demonstrated that our method is effective in the task of translation of historical documents from Chinese to Korean. Using the proposed NER method, the machine translation performance was improved up to 3.70 in terms of the BLEU score (35.83 to 39.53) in seq2seq translation models and up to 3.17 (33.90 to 37.07) in transformer models. Moreover, there was no BLEU score degradation due to the proposed method. The proposed method can be applied to an existing NMT model that uses the attention mechanism without retraining the model, if an NER model exists for the source language. Our method can be successfully applied not only to Chinese-to-Korean translation, but also to other language pairs. In our future work, we plan to explore this direction.

Author Contributions: Conceptualization, M.L., J.L. (Jungi Lee) and G.-J.J.; methodology, J.L. (Jangwon Lee); software, J.L. (Jangwon Lee); validation, J.L. (Jangwon Lee) and G.-J.J.; formal analysis, J.L. (Jungi Lee) and M.L.; investigation, J.L. (Jungi Lee); resources, G.-J.J. and J.L. (Jungi Lee); data curation, J.L. (Jangwon Lee); writing—original draft preparation, J.L. (Jangwon Lee) and G.-J.J.; writing—review and editing, M.L. and J.L. (Jungi Lee); visualization, J.L. (Jangwon Lee); supervision, G.-J.J.; project administration, G.-J.J.; funding acquisition, G.-J.J. All authors read and agree to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. NRF-2017M3C1B6071399), and by the Technology Innovation Program (20016180, Forecast of overseas inflow of new infectious diseases and development of intelligent blocking technology) funded By the Ministry of Trade, Industry & Energy (MOTIE, Korea).

Institutional Review Board Statement: Not applicable because this study is involved with neither humans nor animals.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

NMT	Neural machine translation
NER	Named entity recognition
OOV	Out of vocabulary
RNN	Recurrent neural network
LSTM	Long short-term memory
BLSTM	Bi-directional long short-term memory
GRU	Gated recurrent unit
AAM	Attention alignment map

References

1. Koehn, P.; Och, F.J.; Marcu, D. Statistical Phrase-based Translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology—Volume 1*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2003; pp. 48–54. [CrossRef]
2. Cho, K.; van Merrienboer, B.; Gülçehre, Ç.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv* **2014**, arXiv:1406.1078.
3. Chiang, D. A Hierarchical Phrase-based Model for Statistical Machine Translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2005; pp. 263–270. [CrossRef]
4. Chen, K.; Zhao, T.; Yang, M.; Liu, L.; Tamura, A.; Wang, R.; Utiyama, M.; Sumita, E. A Neural Approach to Source Dependence Based Context Model for Statistical Machine Translation. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2018**, *26*, 266–280. [CrossRef]
5. Wang, X.; Tu, Z.; Zhang, M. Incorporating Statistical Machine Translation Word Knowledge Into Neural Machine Translation. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2018**, *26*, 2255–2266. [CrossRef]
6. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems 27*; Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2014; pp. 3104–3112.
7. Kalchbrenner, N.; Blunsom, P. Recurrent Continuous Translation Models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*; Association for Computational Linguistics: Seattle, WA, USA, 2013; pp. 1700–1709.
8. Cho, K.; van Merriënboer, B.; Bahdanau, D.; Bengio, Y. On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. In *Proceedings of the SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*; Association for Computational Linguistics: Doha, Qatar, 2014; pp. 103–111. [CrossRef]
9. Li, S.; Zhao, J.; Shi, G.; Tan, Y.; Xu, H.; Chen, G.; Lan, H.; Lin, Z. Chinese Grammatical Error Correction Based on Convolutional Sequence to Sequence Model. *IEEE Access* **2019**, *7*, 72905–72913. [CrossRef]
10. Zhang, X.; Yin, F.; Zhang, Y.; Liu, C.; Bengio, Y. Drawing and Recognizing Chinese Characters with Recurrent Neural Network. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 849–862. [CrossRef] [PubMed]
11. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
12. Chung, J.; Gülçehre, Ç.; Cho, K.; Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv* **2014**, arXiv:1412.3555.
13. Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. In Proceedings of the 3rd International Conference on Learning Representations, (ICLR 2015), San Diego, CA, USA, 7–9 May 2015.
14. Luong, M.; Pham, H.; Manning, C.D. Effective Approaches to Attention-based Neural Machine Translation. *arXiv* **2015**, arXiv:1508.04025.
15. Xu, Y.; Liu, W.; Chen, G.; Ren, B.; Zhang, S.; Gao, S.; Guo, J. Enhancing Machine Reading Comprehension With Position Information. *IEEE Access* **2019**, *7*, 141602–141611. [CrossRef]
16. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.u.; Polosukhin, I. Attention is All you Need. In *Advances in Neural Information Processing Systems 30*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2017; pp. 5998–6008.
17. Seljan, S.; Dunđer, I.; Pavlovski, M. Human Quality Evaluation of Machine-Translated Poetry. In Proceedings of the 43rd International Convention on Information, Communication and Electronic Technology (MIPRO), Opatija, Croatia, 18 May 2020.

18. Dunđer, I.; Seljan, S.; Pavlovski, M. Automatic machine translation of poetry and a low-resource language pair. In Proceedings of the 43rd International Convention on Information, Communication and Electronic Technology (MIPRO), Opatija, Croatia, 28 September–2 October 2020. [CrossRef]
19. Dunđer, I. Machine Translation System for the Industry Domain and Croatian Language. *J. Inf. Organ. Sci.* **2020**, *44*, 33–50. [CrossRef]
20. Brkić, M.; Seljan, S.; Vičić, T. Automatic and Human Evaluation on English-Croatian Legislative Test Set. *Lect. Notes Comput. Sci. LNCS* **2013**, *7817*, 311–317. [CrossRef]
21. Zoph, B.; Yuret, D.; May, J.; Knight, K. Transfer Learning for Low-Resource Neural Machine Translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2016; pp. 1568–1575. [CrossRef]
22. Yang, Z.; Cheng, Y.; Liu, Y.; Sun, M. Reducing Word Omission Errors in Neural Machine Translation: A Contrastive Learning Approach. In *Proceedings of the 57th Annual Meeting on Association for Computational Linguistics*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 6191–6196. [CrossRef]
23. Tan, Z.; Wang, S.; Yang, Z.; Chen, G.; Huang, X.; Sun, M.; Liu, Y. Neural machine translation: A review of methods, resources, and tools. *AI Open* **2020**, *1*, 5–21. [CrossRef]
24. Jean, S.; Cho, K.; Memisevic, R.; Bengio, Y. On Using Very Large Target Vocabulary for Neural Machine Translation. *arXiv* **2014**, arXiv:1412.2007.
25. Luong, M.; Manning, C.D. Achieving Open Vocabulary Neural Machine Translation with Hybrid Word-Character Models. *arXiv* **2016**, arXiv:1604.00788.
26. Luong, T.; Sutskever, I.; Le, Q.; Vinyals, O.; Zaremba, W. Addressing the Rare Word Problem in Neural Machine Translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2015; pp. 11–19. [CrossRef]
27. Sennrich, R.; Haddow, B.; Birch, A. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting on Association for Computational Linguistics*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2016; pp. 1715–1725. [CrossRef]
28. Haddad, H.; Fadaei, H.; Faili, H. Handling OOV Words in NMT Using Unsupervised Bilingual Embedding. In Proceedings of the 2018 9th International Symposium on Telecommunications (IST), Tehran, Iran, 17–19 December 2018; pp. 569–574. [CrossRef]
29. Ling, W.; Trancoso, I.; Dyer, C.; Black, A.W. Character-based Neural Machine Translation. *arXiv* **2015**, arXiv:1511.04586.
30. Costa-jussà, M.R.; Fonollosa, J.A.R. Character-based Neural Machine Translation. *arXiv* **2016**, arXiv:1603.00810.
31. Lee, J.; Cho, K.; Hofmann, T. Fully Character-Level Neural Machine Translation without Explicit Segmentation. *Trans. Assoc. Comput. Linguist.* **2017**, *5*, 365–378. [CrossRef]
32. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.J. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2002; pp. 311–318. [CrossRef]
33. Mikolov, T. Statistical Language Models Based on Neural Networks. Ph.D. Thesis, Brno University of Technology, Brno-střed, Czech Republic, 2012.
34. Collobert, R.; Weston, J. A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. In *Proceedings of the 25th International Conference on Machine Learning*; ACM: New York, NY, USA, 2008; pp. 160–167. [CrossRef]
35. Socher, R.; Lin, C.C.Y.; Ng, A.Y.; Manning, C.D. Parsing Natural Scenes and Natural Language with Recursive Neural Networks. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*; Omnipress: Washington, WA, USA, 2011; pp. 129–136.
36. Glorot, X.; Bordes, A.; Bengio, Y. Domain Adaptation for Large-scale Sentiment Classification: A Deep Learning Approach. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*; Omnipress: Washington, WA, USA, 2011; pp. 513–520.
37. Nut Limsopatham, N.C. Bidirectional LSTM for Named Entity Recognition in Twitter Messages. In Proceedings of the 2nd Workshop on Noisy User-Generated Text, Osaka, Japan, 11 December 2016; pp. 145–152. [CrossRef]
38. Aguilar, G.; Maharjan, S.; López Monroy, A.P.; Solorio, T. A Multi-task Approach for Named Entity Recognition in Social Media Data. In *Proceedings of the 3rd Workshop on Noisy User-Generated Text*; Association for Computational Linguistics: Copenhagen, Denmark, 2017; pp. 148–153. [CrossRef]
39. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2018**, arXiv:1810.04805.
40. Schuster, M.; Paliwal, K. Bidirectional Recurrent Neural Networks. *Trans. Sig. Proc.* **1997**, *45*, 2673–2681. [CrossRef]
41. Zhou, P.; Shi, W.; Tian, J.; Qi, Z.; Li, B.; Hao, H.; Xu, B. *Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification*; ACL: Berlin, Germany, 2016.
42. Graves, A.; Schmidhuber, J. Framewise phoneme classification with bidirectional LSTM networks. In Proceedings of the 2005 IEEE International Joint Conference on Neural Networks, Montreal, QC, Canada, 31 July–4 August 2005; Volume 4, pp. 2047–2052
43. Liao, Y.; Xiong, P.; Min, W.; Min, W.; Lu, J. Dynamic Sign Language Recognition Based on Video Sequence with BLSTM-3D Residual Networks. *IEEE Access* **2019**, *7*, 38044–38054. [CrossRef]

44. 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—Volume 70, Sydney, Australia, 6–11 August 2017; pp. 1243–1252.
45. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 21–26 July 2016.
46. Lei Ba, J.; Kiros, J.R.; Hinton, G.E. Layer Normalization. *arXiv* **2016**, arXiv:1607.06450.
47. Lample, G.; Ballesteros, M.; Subramanian, S.; Kawakami, K.; Dyer, C. Neural Architectures for Named Entity Recognition. *arXiv* **2016**, arXiv:1603.01360.
48. Lafferty, J.D.; McCallum, A.; Pereira, F.C.N. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning*; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 2001; pp. 282–289.
49. Park, E.L.; Cho, S. KoNLPy: Korean natural language processing in Python. In Proceedings of the 26th Annual Conference on Human and Cognitive Language Technology, Chuncheon, Korea, 10 October 2014.
50. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26*; Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2013; pp. 3111–3119.
51. Gal, Y.; Ghahramani, Z. A Theoretically Grounded Application of Dropout in Recurrent Neural Networks. In *Advances in Neural Information Processing Systems 29*; Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2016; pp. 1019–1027.
52. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
53. Derczynski, L.; Nichols, E.; van Erp, M.; Limsopatham, N. Results of the WNUT2017 Shared Task on Novel and Emerging Entity Recognition. In *Proceedings of the 3rd Workshop on Noisy User-Generated Text*; Association for Computational Linguistics: Copenhagen, Denmark, 2017; pp. 140–147. [CrossRef]