Improving Context-Aware Neural Machine Translation Using Self-Attentive Sentence Embedding

Hyeongu Yun[1*], **Yongkeun Hwang**[1*], **and Kyomin Jung**[1 2]
[1]Seoul National University, Seoul, Korea
[2]Automation and Systems Research Institute, Seoul National University, Seoul, Korea
{youaredead, wangcho2k, kjung}@snu.ac.kr

Abstract

Fully Attentional Networks (FAN) like Transformer (Vaswani et al. 2017) has shown superior results in Neural Machine Translation (NMT) tasks and has become a solid baseline for translation tasks. More recent studies also have reported experimental results that additional contextual sentences improve translation qualities of NMT models (Voita et al. 2018; Müller et al. 2018; Zhang et al. 2018). However, those studies have exploited multiple context sentences as a single long concatenated sentence, that may cause the models to suffer from inefficient computational complexities and long-range dependencies. In this paper, we propose Hierarchical Context Encoder (HCE) that is able to exploit multiple context sentences separately using the hierarchical FAN structure. Our proposed encoder first abstracts sentence-level information from preceding sentences in a self-attentive way, and then hierarchically encodes context-level information. Through extensive experiments, we observe that our HCE records the best performance measured in BLEU score on English-German, English-Turkish, and English-Korean corpus. In addition, we observe that our HCE records the best performance in a crowd-sourced test set which is designed to evaluate how well an encoder can exploit contextual information. Finally, evaluation on English-Korean pronoun resolution test suite also shows that our HCE can properly exploit contextual information.

1 Introduction

Recently, interests on context-awareness in Neural Machine Translation (NMT) tasks have been increasing since additional contextual information is often crucial to produce adequate translations. For example, especially in spoken languages, duplicated information tends to be omitted frequently if the same information is mentioned in the preceding sentences. That omitted information often cause inaccurate, incomplete or ambiguous translations of spoken languages such as movie subtitles. However, current translation models including Fully Attentional Networks (FAN) (Vaswani et al. 2017) operate on a single sentence level do not take account of contextual sentences, hence they record lower performances in spoken languages compared to those in written and formal language documents.

A few studies have addressed this issue by introducing a secondary context encoder to represent contextual sentences then combining them with the source sentence prior to passing them onto the decoder (Voita et al. 2018; Zhang et al. 2018; Miculicich et al. 2018). They proposed context encoders that encode contextual information in the sentence level vectors and use that information in translating input words. These context encoders handle multiple sentences as long word vectors by concatenating them and do not involve the contextual level information.

Such approaches cause critical drawbacks in handling a larger span of contextual sentences. First, the computational complexity of context encoder scales quadratically both with the number of tokens in each contextual sentence and the number of contextual sentences. Second, (Tang et al. 2018; Tran, Bisazza, and Monz 2018) have empirically shown that FAN is limited at capturing long-range dependencies in translation tasks. Hence, concatenating multiple contextual sentences as a long single sentence is not only computationally expensive, but it also weakens the context-awareness of the model for large contexts.

In this work, we propose a Hierarchical Context Encoder (HCE) to resolve this issue by hierarchically encoding multiple sentences into a contextual level tensor. HCE first encodes each sentence to a tensor with the FAN encoder, then it converts the encoded tensors into a sentence embedding vector by the attentive weighted summation. Since each sentence embedding vector contains the contextual information of each contextual sentence, we are able to build a context-level tensor by listing all the sentence embedding vectors. Then the context-level tensor is fed into another FAN encoder in order to get a tensor with correlative information between contextual sentences, and the obtained tensor is finally combined with the source encoder to form the final encoder output. Our HCE processes each context sentence separately instead of a long concatenated sentence, hence it shows efficiency in computational complexity. The computational complexity of HCE increases linearly as the number of context sentences increase and HCE shows the fastest

[*]Equal contribution

running time among standard baseline models in our experiments.

We conduct a series of extensive experiments on NMT with various language pairs to empirically show that our HCE properly yields better translation with multiple context sentences. Our experiments include public OpenSubtitles corpus in English-German, English-Turkish and our web-crawled movie subtitles corpus in English-Korean. On all language pairs, we observed that the translation qualities of our model outperform all the other models measured in BLEU score.

Furthermore, we have constructed an English→Korean evaluation set by crowd-sourcing in order to analyze how well our HCE exploits contextual information. Our evaluation set consists of two parts, a part where contextual information is helpful for translation and another part where contextual information is unhelpful. We measure translation performances in each part and analyze the effects of contextual encoders including HCE by evaluating the performance gap of the two parts. The results from this evaluation set also show that our HCE performs the best among the baseline models. Lastly we create a test suite for pronoun resolution on English→Korean similar to (Voita et al. 2018; Müller et al. 2018). Evaluation results on the pronoun resolution test suite also reveal the effectiveness of our proposed model. We plan to release both the crowd-sourced evaluation set and the pronoun resolution test suite.

In summary, our contributions are as follows; 1) we propose a novel architecture for embedding multiple sentences into a tensor in order to exploit contextual information in machine translation tasks; 2) we empirically show the effectiveness our model by BLEU score, crowd-sourced helpful/unhelpful evaluation set and a pronoun resolution test suite; and 3) we publicly open our crowd-sourced evaluation set and the pronoun resolution test suite.

2 Related works

Context-aware machine translation models need to focus on additional contexts. In Statistical Machine Translation (SMT), context-awareness is modeled explicitly which is designed for the specific discourse phenomena (Sim Smith 2017). For example, anaphora resolution in translation typically involves identifying previously stated nouns, numbers, and genders in source documents and manipulating restoration in target sentences accordingly.

In NMT, either context of the source or the target language can be considered. Exploiting source-side of contexts requires an encoder to represent the multiple context sentence efficiently (Miculicich et al. 2018; Voita et al. 2018). On the other hand, the use of target-side contexts often involves multi-pass decoding which translates a part of documents or discourses in the sentence level at first, then refines translations using the previous translations as target contexts (Xiong et al. 2019; Voita, Sennrich, and Titov 2019). Our proposed model targets to exploit the source side of context-awareness in this paper.

The simplest approach to incorporate contexts in the source documents is concatenating all context sentences and passing them into a sentence-level model (Tiedemann and Scherrer 2017). In addition, having an extra encoder for contexts is then introduced recently. An extra encoder module for context sentences is a natural extension since the source and context sentences do not have the same significance in translation. In those studies, the context sentences are separately encoded then integrated into the source sentence representations using context-source attention and/or gating network on encoder (Voita et al. 2018), decoder (Jean et al. 2017) or both (Zhang et al. 2018).

Furthermore, structured modeling of context sentences is also suggested to capture complex dependencies between a source sentence and context sentences. For example, (Wang et al. 2017) uses Recurrent Neural Networks (RNN) encoders operating both on sentence and document level. (Miculicich et al. 2018) introduces a hierarchical attention network that encodes context sentences first then summarizes those contexts using a hierarchical structure. (Maruf and Haffari 2018) introduces a memory network augmented model that summarizes and stores context sentences. Our method is closely related to those approaches, as our proposed encoder also incorporates a hierarchically structured abstraction of encoded context sentences. (Maruf, Martins, and Haffari 2019) suggests a context attention module which attends to contexts in both word and sentence level. It uses an averaged word embedding as a sentence-level representation, whereas ours generate sentence-level tensor with FAN encoders resulting in richer sentence representation.

On the other hand, how the quality of translation can be benefited with contextual information is a viable research question (Jean et al. 2017; Bawden et al. 2018). Those researches mainly focus on the design of evaluation tasks that assess the performance of the translation model on handling discourse phenomena problems such as pronoun resolution (Voita et al. 2018; Müller et al. 2018). (Voita, Sennrich, and Titov 2019) also suggests that a carefully designed test suite to evaluate context-aware translation models is crucial since the standard metrics such as BLEU are insensitive on measuring consistency in translation with contexts.

3 Model description

In this section, we briefly review common parts of encoders in the context-aware NMT framework. We also review structures of the context-aware encoders which are our baseline models. Then we introduce a detailed structure of our Hierarchical Context Encoder (HCE). In addition, we analyze computational complexities in our proposed encoder and other baseline models.

3.1 Context-aware NMT encoders

NMT models without contexts take an input sentence x in a source language and return an output sentence y' in a target language. We denote a target sentence as y which is used as a golden truth sentence in supervised learning. Each of x, y, and y' is a tensor that is composed of word vectors, also learnable weights during training.

We especially focus on Fully Attentional Networks(FAN) based models like Transformer (Vaswani et al. 2017) which has recently been widely used in NMT because of its performance and efficiency. Transformer consists of an encoder

module and a decoder module, an encoder extract features in x using self-attention and a decoder generate an output y' from the extracted features using both self-attention with itself and attention with the encoder.

Through a single layer in Transformer encoder, an input tensor passes a self-attention layer using multi-head dot product attention and a position-wise feed-forward layer (Vaswani et al. 2017):

$$TransformerEncoder(x) = FFN(MultiHead(x, x)). \tag{1}$$

The position-wise feed forward layer, denoted as $FFN(x)$, is composed double linear transformation layer with a ReLU activation in between. The multi-head dot product attention $MultiHead$ and the dot product attention $DotProduct$ are given as follows;

$$\begin{aligned} &MultiHead(q, v) = \\ &\quad [DotProduct(q, v)_1, ..., DotProduct(q, v)_H]W, \end{aligned} \tag{2}$$

$$DotProduct(q, v) = softmax(\frac{qW^q W^k v^T}{\sqrt{D}})vW^v, \tag{3}$$

where all W denote learnable weights, D is a dimension of hidden space, and H is a number of heads. Both the self-attention layer and position-wise feed-forward layer are followed by skip connection and layer normalization. In addition, a stack with multiple $TransformerEncoder$ is generally used in order to capture more abundant representations.

With N many additional context sentences $[c_0, ..., c_{N-1}]$ are given, an encoder has to capture contextual information among them then combine the contextual information with source sentence representations. We list four previously suggested models as follows, which are also our baseline models in our experiments;

- **Transformer without contexts (TwoC)**: As a baseline, we have experimented with Transformer without contexts (TwoC) model which has the same structure as (Vaswani et al. 2017). TwoC completely ignores given additional context sentences and only incorporates with the input x and the target y. The computational complexity is $\mathcal{O}((L_s)^2)$, where L_s is a length of input x.
- **Transformer with contexts (TwC)**: The simplest approach is concatenating all context sentences and an input sentence and consider the concatenated sentence as a single input sentence;

  $$x' = Concat([x, c_0, ..., c_{N-1}]). \tag{4}$$

  Then, the output of TwoC encoder is the output of a stacked transformer encoder with x'. The computational complexity is $\mathcal{O}((L_s + NL_c)^2)$, where L_c is a fixed length of context sentences. The complexity becomes quadratically expensive as N grows.
- **Discourse Aware Transformer (DAT) (Voita et al. 2018)**: DAT handles context sentences with an extra context encoder which is also a stacked transformer encoder. We slightly modified DAT to make it available at handling multiple context sentences since (Voita et al. 2018) is originally designed for handling a single context sentence.

  The context encoder has the same structure and even shares its weights with the source encoder through $N_{Layer} - 1$ layers. In the last layer, the context encoder has another transformer encoder module without sharing its weights. The last layer of the source encoder takes an intermediate output tensor h' which is resulted from $N_{Layer} - 1$ stacked transformer encoder, processes both self-attention and context-source attention with t using $MultiHead$;

  $$\begin{aligned} t = &Concat([StackedTransformerEncoder(c_0), ..., \\ &StackedTransformerEncoder(c_N)]), \end{aligned} \tag{5}$$

  $$h_{context} = MultiHead(h', t), \tag{6}$$

  and

  $$h_{source} = MultiHead(h', h'). \tag{7}$$

  the final output tensor of encoder h is given with the gated sum as follows;

  $$h = \sigma(W^h[h_{source}, h_{context}] + b^h), \tag{8}$$

  where W^h is a learnable weights and b^h is a learnable bias term.

  The computational complexity of DAT is $\mathcal{O}(L_s^2 + NL_c^2)$, which is comparable to our model. However, in order to process context-source attention with multiple context sentences, it concatenates all tensors from each context encoders to a long tensor where long-range dependencies of FAN may be limited.
- **Document-level Context Transformer (DCT) (Zhang et al. 2018)**: The encoder of DCT is similar to the DAT, except for the integration of the context and source encoder. Instead of context-source attention and gated sum at the output of both encoders, each layer of the source encoder takes encoded contextual information t and compute context-source attention followed by point-wise feed-forward layer;

  $$h_{context} = MultiHead(h', t), \tag{9}$$

  and

  $$h = FFN(h_{context}). \tag{10}$$

  Since the extensive use of the context-source attention in the encoder, the computational complexity of DCT is $\mathcal{O}(NL_cL_s + L_s^2 + NL_c^2)$. This can grow prohibitively, especially on handling long context sentences or when the number of context sentences is large.
- **Hierarchical Attention Networks (HAN) (Miculicich et al. 2018)**: HAN has a hierarchical structure with two stage at every HAN layer. At the first level of the hierarchy, a single HAN layer encodes each context sentence c_i to an intermediate tensor $e_i \in \mathbb{R}^{L_c \times D}$ with context-source attention;

  $$e_i = MultiHead(h', c_i), \tag{11}$$

  where h' denotes an output from a previous layer or an input x. Each e_i is a tensor with a length of L_c and let e_i^j be the j-th vector of e_i.

At the second level of hierarchy, e_i^j in all context sentences are concatenated through i dimension, resulting tensors $s^j \in \mathbb{R}^{N \times D}$;

$$s^j = Concat([e_0^j, ..., e_N^j]), \tag{12}$$

where N is a number of context sentences. Then, an intermediate output tensor t which contains contextual information queried by each word from the input sentence can be given as follows;

$$t = MultiHead(h', s^j). \tag{13}$$

All $MultiHead$ layers are followed by position-wise feed forward layers and normalization layers. Finally, the output tensor h of HAN encoder is computed with a gated-sum module introduced by (Tu et al. 2017). The aforementioned structure of a single layer in HAN is stacked N_{Layer} times.

The computational complexity of HAN encoder is $\mathcal{O}(NL_cL_s + L_s^2 + NL_c^2)$ which is also comparable to our proposed model. Nonetheless, HAN encoder requires context-source attention two times at every layers. Also, since the second context-source attention is performed on $s_i = Concat([e_0^j, ..., e_N^j])$, HAN does not take account of internal correlations among $[e_i^0, ..., e_i^{L_c}]$.

3.2 Hierarchical context encoder

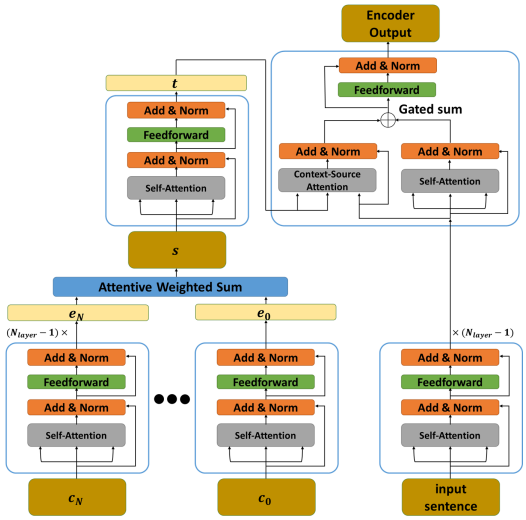

Figure 1: The structure of our proposed Hierarchical Context Encoder. Each context sentences c_i is encoded through transformer encoders to the tensor e_i and the attentive weighted sum module vectorizes each e_i to the vector s_i. Upper transformer encoder encodes the input tensor s composed by concatenation $s = [s_0, ..., s_N]$ and outputs our final context representation tensor t. Then the context representation is combined to the source encoder by gated sum.

We propose a novel context encoder that hierarchically encodes multiple sentences into a tensor. Our proposed encoder, Hierarchical Context Encoder (HCE), is designed to capture correlations between sentences in contexts as well as correlations between words in each sentence.

Each context sentence c_i after word embedding layer is given as a tensor of order 2; $c_i \in \mathbb{R}^{L_c \times D'}$ where L_c is a maximum length of each context sentence and D' is a dimension of word embeddeding vectors. In the lower part of hierarchy, HCE encodes each of c_i to sentence-level tensor e_i using the stacked transformer encoder as (Vaswani et al. 2017);

$$e_i = StackedTransformerEncoder(c_i). \tag{14}$$

Each encoded sentence-level tensor e_i is also a tensor of order 2, $e_i \in \mathbb{R}^{L_c \times D}$ where D is a hidden dimension.

We then compress each encoded sentence-level tensor into a sentence-level vector by a self-attentive weighted sum module which is similar to that of (Lin et al. 2017). Our self-attentive weighted sum module takes e_i as an input tensor and computes a vector s_i as follows;

$$s_i = \sum_j \alpha_j e_{ij}, \tag{15}$$

$$\alpha = FFN(MultiHead(e_i, e_i)). \tag{16}$$

The output of the attentive weighted sum module s_i is a vector representing the information of each i-th context sentence. Then we concatenate $[s_0, ..., s_N]$ to a context embedding tensor s. The context embedding tensor $s \in \mathbb{R}^{N \times D}$ is fed into another FAN encoder layer which is the upper part of the hierarchy to encode the whole contextual information into a single tensor t;

$$t = TransformerEncoder(s). \tag{17}$$

Finally, the contextual information tensor t is combined to source encoder by gated sum as Equation 6, 7, and 8, which is the same process introduced by (Voita et al. 2018). Full structure of HCE is depicted in Figure 1.

The main difference between HCE and other baseline models especially HAN is that HCE encodes each context sentence as the way of sentence embedding with self-attention independent to the source word, while HAN uses context-source attention. To explain more in detail, two main differences between the hierarchical FAN structures of HAN and HCE are as follows: 1) at the bottom part of the hierarchy, HCE encodes each context sentence to a tensor with self-attention while HAN encodes each context sentence with context-source attention using query words from input sentences; and 2) at the upper part of the hierarchy, HCE first uses the self-attentive weighted sum to encode a tensor into a vector which contains the whole information from each context sentence, then encodes the whole contexts with self-attention again. On the other hand, HAN uses context-source attention again. To summarize, HCE only models the context-source relations at the upper part of the hierarchy resulting in a simpler and clearer model structure.

The computational complexity of HCE is $\mathcal{O}(L_s^2 + NL_c^2)$. HCE extracts more compact context-level representation from each sentence-level representation by self-attentive weighted sum over each e_i, hence it complements DAT (Voita et al. 2018) and DCT (Zhang et al. 2018) whereas they

take the whole contexts as a single sentence by concatenation. Besides, the encoding procedure of context sentences is not dependent on the input sentence x unlike HAN. This allows HCE to cache context-level representations t of frequently appeared context sentences, which is important in implementing a real-time application.

4 Data

We experimented with our model and baseline models on English-German TED corpus, English-German OpenSubtitles corpus, English-Turkish OpenSubtitles corpus, and our web-crawled English-Korean subtitle corpus.

4.1 English-German IWSLT 2017 corpus

We use the English-German corpus from the IWSLT 2017 evaluation campaign (Cettolo et al. 2017), which is publicly available on WIT^3 website[1]. The corpus consist of transcriptions and their translations of TED talks. We combine `dev2010` and `tst2010` into a development(*dev*) set and `tst2015` as a *test* set. We extract context-aware dataset where each set consists of a *source*, a *target* sentence and multiple *context* sentences. Since the corpus is aligned as sentence level, we assume that every 2 preceding sentences are *context* sentences. We also include context sentences only within the same talk of the source sentence, as the data is separated as talks. The resulting dataset consists of $211k$, $2.4k$, $1.1k$ examples of *train*, *dev*, *test* sets respectively. Also, we put a special *beginning of context* token at the beginning of each context sentences to differentiate from source sentences. Finally, we have used a byte-pair encoded vocabulary with about 16,000 tokens.

4.2 OpenSubtitles corpus

We also choose the OpenSubtitles corpus for English-German and English-Turkish tasks. We use the 2018 version (Lison, Tiedemann, and Kouylekov 2018) of the data, each consist of $24.4M$, $47.4M$ parallel sentences respectively. Following the approach in (Voita, Sennrich, and Titov 2019), we first cleaned the data by picking only pairs with a time overlap of subtitle frames at least 0.9. After cleaning, we take $7.5M$ and $20.2M$ sentences for English-German and English-Turkish corpus.

We then take the *context* sentences by using the timestamp of each subtitle. The timestamps contain start time and end time in ms for each subtitle. We focus on the start times to compile a set of data including a source sentence and preceding contextual sentences. We assume that if the start time of a preceding sentence is within 3000 ms from the start time of a sentence then that preceding sentence contains the contextual information. We set the maximum number of preceding contextual sentences up to 2.

4.3 English-Korean subtitle corpus

Finally, for English-Korean experiments, we construct a web-crawled subtitle corpus with 5,917 files. These files are English-Korean bilingual subtitle files of movies, TV series, and documentary films from various online sources. We set randomly selected $5.3k$ files for *train*, 500 files for *dev*, and 50 files for *test* set. The *train* set includes $3.0M$ sentences, the *dev* set includes $28.8k$ sentences, and the *test* set includes $31.1k$ sentences. Our web-crawled English-Korean bilingual subtitle files include time stamps for each subtitles. Thus we pre-process those files as similar as processing in Section 4.2. The resulting data have $1.6M$ sets of serial sentences in *train set*, $155.6k$ sets in *dev set*, and $18.1k$ sets in *test set*. We also have used a byte-pair encoded vocabulary with about 16,500 tokens for English-Korean experiments. We display some raw samples from our test files in Table 1.

5 Experiments

We evaluate our HCE by BLEU score, model complexity, BLEU on helpful/unhelpful set, and accuracy on the pronoun resolution set. All experimental results show the effectiveness of HCE compared to baseline models.

5.1 Hyperparameters and Training details

Through our experiments, we use 512 hidden dimensions for all layers including words embedding layers, FAN layers, and the encoded context layer. We set $N_{Layer} = 6$ for all models and share the weights of the source encoder to context encoder for the DAT, HAN, and HCE models. For all attention mechanisms, we set the number of heads as 8. The dropout rate of each FAN layers is set to 0.1.

For each language pair, we tokenize each text by the wordpiece model (Schuster and Nakajima 2012; Wu et al. 2016) with a vocabulary of about 16,000 tokens. Also, we put a special *beginning of context* token `<BOS>` at the beginning of each context sentences to differentiate from source sentences.

We implement all the evaluated models using the `tensor2tensor` framework (Vaswani et al. 2018). We train all models with ADAM (Kingma and Ba 2014) optimizer with learning rate 1e-3 and adopt early stopping with *dev* loss. Unlike (Miculicich et al. 2018; Zhang et al. 2018; Maruf, Martins, and Haffari 2019), we do not use the iterative training which trains the model on a sentence-level task first, then fine-tunes the model with contextual information. All the models we have evaluated are trained from scratch with random initialization.

For scoring BLEU, we use the `t2t-bleu` script[2] which outputs the identical results as Moses script (Koehn et al. 2007).

5.2 Overall BLEU evaluation

We measure performances of HCE and other five baseline models in English-German (IWSLT'17 and OpenSubtitles), English-Turkish (OpenSubtitles), and English-Korean(our Web-crawled corpus). Overall BLEU scores on all eight datasets are displayed in Table 2. Our model yields the best performances on all eight datasets. Especially on our Web-crawled English-Korean, HCE shows superior performance compared to other models. These results indicate that our

[1] https://wit3.fbk.eu/mt.php?release=2017-01-trnted

[2] https://github.com/tensorflow/tensor2tensor

Start Time	End Time	English	Korean
		. . .	
337733	339967	Daniel likes hanging out with his cousins.	다니엘은 사촌들과 노는걸 좋아했거든요
340035	341168	He's been going back and forth until Leith and I	양육권을 제대로 가질 수 있을때까지
341236	342303	can settle custody.	왔다 갔다 했어요
344373	345940	Listen, don't worry.	너무 걱정 마세요
		. . .	

Table 1: Bilingual subtitle samples from our English-Korean test files

Corpus	IWSLT'17		OpenSubtitles		OpenSubtitles		Web-crawled	
Language pair	En→De	De→En	En→De	De→En	En→Tr	Tr→En	En→Ko	Ko→En
Transformer without contexts	28.25	32.18	27.95	33.93	24.89	36.27	8.58	23.67
Transformer with contexts	28.65	32.68	28.07	34.04	23.96	35.81	9.46	24.23
DCT (Zhang et al. 2018)	26.76	30.33	26.3	32.05	21.91	34.3	6.5	20.72
DAT (Voita et al. 2018)	28.82	32.59	28.09	33.99	24.30	35.23	8.56	23.91
HAN (Miculicich et al. 2018)	28.85	32.72	28.00	34.42	24.86	36.55	8.76	24.41
HCE (ours)	**28.89**	**33.01**	**28.40**	**34.59**	**25.11**	**36.84**	**11.30**	**26.70**

Table 2: BLEU score. Our proposed Hierarchical Context Encoder have shown the best results in all language pairs.

model exploits given contextual sentences effectively and translate better than all five baseline models in English-German, English-Turkish and English-Korean translation tasks.

5.3 Model complexity analysis

Model	training speed (steps/sec)	inference time (tokens/sec)	# of Params
TwC	4.07	62.10	61.0M
DCT	2.42	45.32	98.7M
DAT	4.59	65.07	69.9M
HAN	4.47	64.05	66.2M
HCE	**4.67**	**65.12**	66.7M

Table 3: Training speed, inference time and number of parameters.

We also observe that our HCE is the most efficient in training speed and inference time among our baselines. In Table 3, HCE records the fastest training speed and inference time indicating that HCE has the most computationally efficient structure. These results also show that the performance gain of HCE is not only from the complexity of the model but the structural strength because the number of parameters is comparable to others.

5.4 BLEU evaluation on helpful/unhelpful context

In order to verify that our model actually uses the contextual information to improve translation quality, we conduct an additional experiment with a part of data where contextual sentences are helpful for translating and the other part of data where they are not. We randomly choose 10,000 sets of serial sentences from our *test set* of En→Ko data and split them up into two parts by crowd-sourcing with Amazon Mechanical Turk (Buhrmester, Kwang, and Gosling 2011). The first part consists of 4,331 sets of which context sentences are helpful for translating (*e.g.* context sentences include critical information, exact referred object by pronouns, or residual parts of an incomplete source sentence). The remaining part consists of 5,669 sets of which context sentences are unrelated to translate the source sentences.

We examine BLEU scores of two parts separately to observe how well each model uses helpful contexts. The results are displayed in Table 4. We observe a large gap between BLEU score on *helpful set* and that on *unhelpful set* with all four baseline models, showing that *helpful set* is harder to translate because abstracting and exploiting contextual information is likely to be mandatory to translate *helpful set*. On the other hand, HCE closes the gap between BLEU scores on each set, indicating that HCE understands the contextual information and is able to perform on *helpful set* as well as on *unhelpful set*.

5.5 En→Ko pronoun resolution test suite

Finally, we evaluate the accuracy of all models that use contexts on our En→Ko pronoun resolution test suite. we create a test suite for English→Korean pronoun resolution to examine how well a model understands contextual information. Our test suite is composed of 150 sets, each of which includes 1) a source sentence with a pronoun, 2) preceding contextual sentences with the exact word referred to by the pronoun, 3) a target sentence with the corresponding pronoun, 4) a *correct* target sentence where the pronoun is replaced with the exact word, and 5) a *wrong* target sentence where the pronoun is replaced with an unrelated word. We follow a scoring method in (Müller et al. 2018) for evaluation; if a model's negative log-likelihood of *correct* sentence is lower than that of *wrong* sentence, then we consider the

Model	Total set	helpful set	unhelpful set	BLEU gap
Transformer without contexts	7.46	6.69	8.04	+1.35
Transformer with contexts	8.29	7.45	8.92	+1.47
DAT (Voita et al. 2018)	8.22	7.48	8.77	+1.29
HAN (Miculicich et al. 2018)	8.34	7.44	9.01	+1.57
HCE (ours)	10.27	10.08	10.40	**+0.32**

Table 4: BLEU score evaluations with helpful contexts set and unhelpful contexts set from En→Ko test data. All four baseline models have shown large gap between BLEU score on *helpful* contexts set and BLEU score on *unhelpful* contexts set. On the other hand, Our proposed Hierarchical Context Encoder has almost closed the gap between BLEU scores on two sets.

Label	English	Korean
context 1	When did the **tower** collapse?	
context 0	Oh, last winter.	
source / target	Brother Remigius says we haven't the funds to repair **it**.	레미져스 수사님 말론 **그걸** 고칠 돈이 없다는군.
correct		레미져스 수사님 말론 **탑**을 고칠 돈이 없다는군.
wrong		레미져스 수사님 말론 **지붕**을 고칠 돈이 없다는군.

Table 5: A sample set of English→Korean pronoun resolution test suite

Model	accuracy
Transformer with contexts	0.25
DAT (Voita et al. 2018)	0.44
HAN (Miculicich et al. 2018)	0.47
HCE (ours)	**0.48**

Table 6: Accuracy on our En→Ko pronoun resolution test suite.

model is able to detect wrong pronoun translation.

A sample from our test suite is displayed in Table 5, the pronoun and corresponding words are emphasized in bold. In the sample, the *source* sentence has a pronoun "**it**" referring the word "**tower**" in the *context 1* sentence. The *target* sentence also has the corresponding boldfaced pronoun in Korean, "그걸 (it)". We replace the pronoun in *target* sentence to the exact referring Korean word "탑 (tower)" in the *correct* sentence, and we replace it to an unprecedented yet similar Korean word, "지붕 (roof)" in the *wrong* sentence.

The results are displayed in Table 6. While TwC scores the lowest accuracy with 0.25, DAT and HAN record accuracy with 0.44 and 0.47 respectively. HCE records the highest accuracy of 0.48 in this test. These results support the hypothesis that it is harder to capture contextual information on a single long concatenated sentence than on structured multiple context sentences. Also, the result that HCE and HAN both perform better than DAT reveals the strength of hierarchical structure for multiple contexts which is able to capture the contextual information effectively.

5.6 Qualitative Analysis

Table 7 shows three examples how contextual encoders attend and comprehend the context sentences while translating a particular pronoun. The words in brackets next to the input sentences are the words in context sentences referred by each boldfaced pronoun. The intensity of color (orange)

Model	Input sentence & Visualization
	I want to know what you told **him** that night. (*My father*)
DAT	c_0 <BOC> My father met with you right before he died . <EOS> c_1 <BOC> This is business . <EOS>
HAN	c_0 <BOC> My father met with you right before he died . <EOS> c_1 <BOC> This is business . <EOS>
HCE	c_0 <BOC> My father met with you right before he died . <EOS> c_1 <BOC> This is business . <EOS>
	Do you have any idea what **his** family has done? (*Dan*)
DAT	c_0 <BOC> Helping us ? <EOS> c_1 <BOC> Chuck , Dan has been helping us , unlike you . <EOS>
HAN	c_0 <BOC> Helping us ? <EOS> c_1 <BOC> Chuck , Dan has been helping us , unlike you . <EOS>
HCE	c_0 <BOC> Helping us ? <EOS> c_1 <BOC> Chuck , Dan has been helping us , unlike you . <EOS>
	She can be the one to tell me or not tell me. (*Lilly*)
DAT	c_0 <BOC> Uh , since this is about Lily . <EOS> c_1 <BOC> But my goal remains the same . <EOS>
HAN	c_0 <BOC> Uh , since this is about Lily . <EOS> c_1 <BOC> But my goal remains the same . <EOS>
HCE	c_0 <BOC> Uh , since this is about Lily . <EOS> c_1 <BOC> But my goal remains the same . <EOS>

Table 7: Three visualization examples of attention weights for given pronoun boldfaced words which are referring to the words in brackets. We refer each of them as (a) the uppermost example, (b) the middle example, and (c) the bottom example.

is proportional to the attention weight for each word. Also, the intensity of color (blue) is proportional to the attention weight for each context sentence in HCE and HAN.

In general, the third-person pronouns in English are often translated into Korean pronouns that do not contain attributes like gender, or phrases indicating the referenced person or object. For example, the word "his" in the middle example (b) has translated as "쟤네 (their)" which is a correct Korean possessive pronoun for referring "Dan" in c_1 sentence. In the bottom example (c), the word "She" has translated as "본인 (oneself)" which can be used for both male and female. Likewise, the word "him" in the uppermost example (a) has translated as "아버지 (father)" which is the exact referred word. Considering such phenomena, we regard that correctly referencing the proper nouns is crucial in translating pronouns into Korean.

From this point of view, Table 7 explains the strength of HCE in the En→Ko translation. As presented in Table 7, we observed that HCE gives more attention to the context sentences which contain the exact referred words. Hence, the upper hierarchy of HCE pays its attention to the more important sentence as we have intended. We also observed that both our HCE and HAN tend to attend to nouns such as names of people (e.g. Dan, Chuck) or names of specific locations (e.g. the church, Paris). Nevertheless, HCE more accurately attends to the exact referred words comparing to HAN. In the first example, HCE gives large portion of its attention to "My father" while HAN choose "business" as the most important word. The second example also shows the ability of HCE to exploit context information properly. HCE understands that the word "Dan" is more important than "Chuck", while HAN gives most of its attention to the word "Chuck" except for the <EOS> token. Although HCE computes context representations independent of the input query, these visualization examples show that HCE can correctly attend to the exact words referred by the pronouns.

6 Conclusion

In this work, we have introduced Hierarchical Context Encoder (HCE) structure which is able to encode multiple contextual sentences with hierarchical FAN structure. We have shown that our model outperforms all baseline models in English-German, English-Turkish and English-Korean translation tasks and also that our model is the most efficient in computational complexity. We also have shown that our model closes the gap of translation quality between the sentences with helpful contexts and the sentences with unrelated contexts, indicating that our model is better at exploiting the helpful contextual information for translating than baseline models. Analysis on pronoun resolution test suite support the effectiveness of our HCE.

Acknowledgements

This work was supported by Samsung Electronics Co., Ltd.

References

Bawden, R.; Sennrich, R.; Birch, A.; and Haddow, B. 2018. Evaluating Discourse Phenomena in Neural Machine Translation. In *NAACL*, 1304–1313.

Buhrmester, M.; Kwang, T.; and Gosling, S. D. 2011. Amazon's mechanical turk: A new source of inexpensive, yet high-quality, data? *Perspectives on psychological science* 6(1):3–5.

Cettolo, M.; Federico, M.; Bentivogli, L.; Niehues, J.; Stüker, S.; Sudoh, K.; Yoshino, K.; and Federmann, C. 2017. Overview of the IWSLT 2017 Evaluation Campaign. In *Proceedings of the 14th International Workshop on Spoken Language Translation (IWSLT 2017)*, 1–14.

Jean, S.; Lauly, S.; Firat, O.; and Cho, K. 2017. Does neural machine translation benefit from larger context? *arXiv preprint arXiv:1704.05135*.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Koehn, P.; Hoang, H.; Birch, A.; Callison-Burch, C.; Federico, M.; Bertoldi, N.; Cowan, B.; Shen, W.; Moran, C.; Zens, R.; et al. 2007. 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*, 177–180.

Lin, Z.; Feng, M.; Santos, C. N. d.; Yu, M.; Xiang, B.; Zhou, B.; and Bengio, Y. 2017. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*.

Lison, P.; Tiedemann, J.; and Kouylekov, M. 2018. Opensubtitles2018: Statistical rescoring of sentence alignments in large, noisy parallel corpora. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*.

Maruf, S., and Haffari, G. 2018. Document Context Neural Machine Translation with Memory Networks. In *ACL*, 1275–1284.

Maruf, S.; Martins, A. F. T.; and Haffari, G. 2019. Selective Attention for Context-aware Neural Machine Translation. In *NAACL*.

Miculicich, L.; Ram, D.; Pappas, N.; and Henderson, J. 2018. Document-Level Neural Machine Translation with Hierarchical Attention Networks. In *EMNLP*, number i, 2947–2954.

Müller, M.; Rios, A.; Voita, E.; and Sennrich, R. 2018. A large-scale test set for the evaluation of context-aware pronoun translation in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, 61–72.

Schuster, M., and Nakajima, K. 2012. Japanese and Korean voice search. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, volume 1, 5149–5152.

Sim Smith, K. 2017. On Integrating Discourse in Machine Translation. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, number Section 2, 110–121.

Tang, G.; Müller, M.; Rios, A.; and Sennrich, R. 2018. Why Self-Attention? A Targeted Evaluation of Neural Machine Translation Architectures. In *EMNLP*.

Tiedemann, J., and Scherrer, Y. 2017. Neural Machine Translation with Extended Context. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, 82–92.

Tran, K.; Bisazza, A.; and Monz, C. 2018. The importance of being recurrent for modeling hierarchical structure. *arXiv preprint arXiv:1803.03585*.

Tu, Z.; Liu, Y.; Lu, Z.; Liu, X.; and Li, H. 2017. Context gates for neural machine translation. *Transactions of the Association for Computational Linguistics* 5:87–99.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Vaswani, A.; Bengio, S.; Brevdo, E.; Chollet, F.; Gomez, A. N.; Gouws, S.; Jones, L.; Kaiser, L.; Kalchbrenner, N.; Parmar, N.; Sepassi, R.; Shazeer, N.; and Uszkoreit, J. 2018. Tensor2tensor for neural machine translation. *arXiv preprint* arXiv:1803.07416.

Voita, E.; Serdyukov, P.; Sennrich, R.; and Titov, I. 2018. Context-aware neural machine translation learns anaphora resolution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1264–1274.

Voita, E.; Sennrich, R.; and Titov, I. 2019. When a Good Translation is Wrong in Context: Context-Aware Machine Translation Improves on Deixis, Ellipsis, and Lexical Cohesion. In *ACL*.

Wang, L.; Tu, Z.; Way, A.; and Liu, Q. 2017. Exploiting Cross-Sentence Context for Neural Machine Translation. In *EMNLP*.

Wu, Y.; Schuster, M.; Chen, Z.; Le, Q. V.; Norouzi, M.; Macherey, W.; Krikun, M.; Cao, Y.; Gao, Q.; Macherey, K.; Klingner, J.; Shah, A.; Johnson, M.; Liu, X.; Kaiser, Ł.; Gouws, S.; Kato, Y.; Kudo, T.; Kazawa, H.; Stevens, K.; Kurian, G.; Patil, N.; Wang, W.; Young, C.; Smith, J.; Riesa, J.; Rudnick, A.; Vinyals, O.; Corrado, G.; Hughes, M.; and Dean, J. 2016. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. Technical report.

Xiong, H.; He, Z.; Wu, H.; and Wang, H. 2019. Modeling Coherence for Discourse Neural Machine Translation. In *AAAI*, number 10.

Zhang, J.; Luan, H.; Sun, M.; Zhai, F.; Xu, J.; Zhang, M.; and Liu, Y. 2018. Improving the transformer translation model with document-level context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 533–542.