General Paper Peer-Reviewed

Generic Mechanism for Reducing Repetitions in Encoder-decoder Models

Ying Zhang[†], Hidetaka Kamigaito[††], Tatsuya Aoki[†],
Hiroya Takamura[†††] and Manabu Okumura[†]

Encoder-decoder models have been commonly used; they have achieved state-of-the-art results for many natural language generation tasks. However, according to the reports of previous studies, encoder-decoder models suffer from generating redundant repetitions. Thus, we herein propose a repetition reduction module (RRM) for encoder-decoder models that estimates the semantic difference of a source sentence before and after it is fed into the model to capture the consistency between the two sides. As an autoencoder, the proposed mechanism supervises the training of encoder-decoder models to reduce the number of repeatedly generated tokens. The evaluation results of the publicly available machine translation and response generation datasets demonstrate the effectiveness of our proposal.

Key Words: *Encoder-decoder, Repetition Reduction, Autoencoder*

1 Introduction

Sequence-to-sequence (seq2seq) models (i.e., *encoder-decoder* models) are a dominant paradigm in various natural language generation tasks such as machine translation (Luong et al. 2015b; Tu et al. 2016), text summarization (Kiyono et al. 2018; Li et al. 2017), and response generation (Miller et al. 2017; Pasunuru and Bansal 2018).[1] However, Mi et al. (2016) reported that basic seq2seq models (Bahdanau et al. 2015; Luong et al. 2015b) sometimes suffer from the repetition problem, which is the generation of duplicate fragments. Several studies have been conducted to explain the issue of repetition. For instance, Holtzman et al. (2020) established different probability distributions between natural-language text and beam-search-decoded text. Counter-intuitively, the models would assign a higher probability to generic, repetitive, and awkward texts than to grammatical or natural texts. This causes repetition when a maximization-based decod-

[†] Tokyo Institute of Technology
[††] NARA Institute of Science and Technology (NAIST)
[†††] National Institute of Advanced Industrial Science and Technology (AIST)

[1] This paper was published in the Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021), when the third author was a PhD candidate at Tokyo Institute of Technology.

ing method is used. Fu et al. (2021) claimed that many words predicted the same next word with a high probability, which caused a tendency for the word and formed repetitions. Xu et al. (2022) observed that the model had a strong preference to repeat the previous sentence. This preference is stronger when providing natural text rather than random text. Tu et al. (2016) and Mi et al. (2016) claimed that the attention mechanism for the seq2seq model did not explicitly consider which source side tokens had already been covered in the past attentions. Thus, the *decoder* repeatedly attended to the translated source token during the decoding steps, which led to redundant generation. Following Tu et al. (2016) and Mi et al. (2016), this study regarded over-considered attention to source tokens as the main cause of the repetition problem when using the seq2seq model.

Several researchers have proposed variants of the seq2seq model to address redundant repetitions. The coverage mechanism (Tu et al. 2016; Mi et al. 2016) prevents the model from generating redundant outputs by considering the coverage of the attention distribution. These approaches can be easily incorporated into the seq2seq model with a single attention distribution between the *encoder* and the *decoder*. However, for seq2seq models with multiple attentions such as the transformer (Vaswani et al. 2017), calculating the coverage of attention is intractable because the *encoder* attempts to attend to multiple attentions on each layer in the *decoder*. Thus, incorporating the coverage mechanism into multi-attention-based seq2seq models is challenging.

Additionally, Suzuki and Nagata (2017) proposed word-frequency estimation (WFE), which predicts the upper-bound frequency for each output token from given input tokens to control redundancy in the output. Furthermore, Kiyono et al. (2018) proposed a source-side prediction module (SPM) that estimates the occurrences of input tokens from the hidden states of the *decoder* in the seq2seq model to reduce repetition. The SPM works as an autoencoder (Bengio et al. 2009) to encode and reconstruct the input word frequency to supervise model training. Although WFE and SPM do not depend on the structure of the seq2seq model, it is difficult to apply them to tasks other than text summarization because they assume that the input sentence contains more tokens than the output. In addition to the SPM, several studies have proven the effectiveness of using autoencoders to learn semantic representations and supervise model training in natural language generation tasks (Ma et al. 2018; Luo et al. 2018; Liu et al. 2019).

To address these problems, inspired by previous research (Tu et al. 2016; Kiyono et al. 2018; Ma et al. 2018), we herein propose a generic approach, RRM, to supervise the attention distribution for the seq2seq model. The RRM shares the model architecture with the seq2seq model and functions as an autoencoder to focus on the differences between the embedding spaces of the source and target sides. Based on the assumption of distributional semantics, RRM regards

German-English translation			
	Short	*Medium*	*Long*
Source	die einzige wahre wahl war " wer " , nicht wann , und nicht was sie danach taten .	wir nehmen also etwas sehr kompliziertes , wandeln es in töne um , eine sequenz von tönen , und produzieren damit etwas sehr kompliziertes in den köpfen von anderen .	sie ist ein prozess , und manchmal funktioniert er und manchmal nicht , aber die idee , dass wir der wissenschaft nicht erlauben sollten , ihre arbeit zu tun , weil wir angst haben ist eine wirkliche sackgasse , und sie hält millionen von menschen vom aufblühen ab .
Reference	the only real choice was who , not when , and not what you did after .	but we 're taking something very complicated , turning it into sound , sequences of sounds , and producing something very complicated in your brain .	it 's a process , and sometimes it works and sometimes it doesn 't , but the idea that we should not allow science to do its job because we 're afraid , is really very deadening , and it 's preventing millions of people from prospering .
LocalJoint	the only real choice was who , not when , not when , and not what **they** did after that .	so we take something very complicated , we turn it into sound , we turn it into sound sequence , and we produce something very complicated in the head **of others** .	it 's a process , and sometimes it doesn 't work and sometimes it doesn 't work , but the idea that we shouldn 't allow science to do their work , because we 're afraid to have a real dead end , and it keeps millions of people from flourishing .
+RRM	the only real choice was **"** who , **"** not when , and not what **they** did after that .	so we take something very complicated , we turn it into sound , a sequence of sound , and we produce something very complicated in the head **of others** .	it 's a process , and sometimes it works and sometimes it doesn 't , but the idea that we shouldn 't allow science to do its job because we 're afraid is a truly dead end , and it keeps millions of people from flourishing .

Table 1 Sample translations for *short*, *medium*, and *long* data. Underline indicates repetitions with more than two words and **bold** indicates wrong translations. Scaling factor α for balancing losses of the proposed repetition reduction module (RRM) and a baseline model, LocalJoint, was fixed to 0.3, which yielded the least repeat on the validation dataset.

the representations of an input sentence on both sides as word vectors and attempts to minimize their differences during training. Hence, the seq2seq model explicitly considers the source-side context in the *decoder*.

Our experimental results on the IWSLT 2014 German-to-English translation task (Cettolo et al. 2014), WMT 2014 English-to-German translation task (Bojar et al. 2014), and PERSONA-CHAT response generation task (Zhang et al. 2018) demonstrate that the proposed method effectively alleviates the nonconsecutive repetition problem for the seq2seq model. Sample translations are listed in Table 1.

2 Background

Based on the coverage mechanism, we attributed the redundant repetitions in seq2seq models to the over-considered attention to the source tokens. To solve this problem, the RRM works as an autoencoder to supervise the seq2seq models during training. Here, we describe the seq2seq

model (Luong et al. 2015b), coverage mechanism (Tu et al. 2016), and autoencoder (Bengio et al. 2009) using their mathematical notations. Additionally, we list the types of repetitions introduced by Fu et al. (2021) and Xu et al. (2022).

2.1 Seq2seq Model

Given a source sentence $X = (x_1, \ldots, x_I)$, the seq2seq model generates target sentence $Y = (y_1, \ldots, y_J)$, where I and J are the numbers of source and target tokens, respectively. The seq2seq model consists of two main parts: *encoder* and *decoder*. The *encoder* computes the representation of source sentence X and the *decoder* generates target sentence Y. As an autoregressive model, the seq2seq model with parameter θ decomposes the conditional probability $p(Y|X;\theta)$ as follows:

$$p(Y|X;\theta) = \prod_{j=1}^{J} p(y_j|y_1, \ldots, y_{j-1}, X;\theta), \tag{1}$$

$$p(\cdot|y_1, \ldots, y_{j-1}, X;\theta) = o_j \tag{2}$$

$$= \mathrm{softmax}(W_o \tilde{z}_j + b_o), \tag{3}$$

where $\tilde{z}_j$ denotes the final hidden state of the *decoder* during the j-th decoding step. o_j denotes the probability distribution over target vocabulary V_t at the j-th decoding step. W_o is a weight matrix and b_o is a bias term.

Model parameter θ is optimized by minimizing the cross-entropy loss (Brier 1950) as follows:

$$\ell_{CrossEntropy} = -\log p(Y|X;\theta) = -\frac{1}{J}\sum_{j=1}^{J}\sum_{c}^{|V_t|} y_{j,c} \log o_{j,c}, \tag{4}$$

where $|V_t|$ denotes the length of V_t; $o_{j,c}$ denotes the probability at the j-th decoding step for vocab $c \in V_t$, and

$$y_{j,c} := \begin{cases} 1 & \text{if } y_j = c \\ 0 & \text{if } y_j \neq c. \end{cases} \tag{5}$$

Let FNN denote the feed-forward layer and Concat be the concatenation layer. When using global attention (Luong et al. 2015b), $\tilde{z}_j$ is computed as follows:

$$s_j = \sum_{i=1}^{I} a_{i,j} h_i = \sum_{i=1}^{I} \mathrm{softmax}(f(h_i, z_j)) h_i, \tag{6}$$

$$\tilde{z}_j = \mathrm{FNN}(\mathrm{Concat}(s_j, z_j)), \tag{7}$$

where h_i is the i-th hidden state of X and z_j is the hidden state at the j-th decoding step.

Both h_i and z_j were computed using long short-term memory networks (LSTM) (Hochreiter and Schmidhuber 1997). z_1 is the state of special token <sos>, which indicates the start of the sequence. $H = (h_1, \ldots, h_I)$ denotes the set of hidden states in X. Score function f evaluates the content matching between h_i and z_j.

Vaswani et al. (2017) proposed a seq2seq model with multihead self-attention mechanism to compute $\tilde{z}_j$, transformer, which outperformed the global attention-based seq2seq model and became the dominant paradigm in the natural language generation tasks. Let Attn denote a self-attention layer. We assume that both the *encoder* and *decoder* of the transformer consist of L layers with N attention heads. The *encoder* encodes X for representation $\widetilde{H}^L$ using

$$h_i^l = \text{MultiHeadAttn}(\tilde{h}_i^{l-1}, \widetilde{H}^{l-1}, \widetilde{H}^{l-1}) = \text{Concat}(\text{head}_1, \ldots, \text{head}_N) W_a^{l-1}, \tag{8}$$

$$\text{head}_n = \text{Attn}(\tilde{h}_i^{l-1}, \widetilde{H}^{l-1}, \widetilde{H}^{l-1}), \quad n \in \{1, \ldots, N\}, \tag{9}$$

$$\tilde{h}_i^l = \text{FNN}(h_i^l), \tag{10}$$

where W_a^{l-1} is a weight matrix and $\tilde{h}_i^0$ is the embedding of the i-th token in X. $\widetilde{H}^l = (\tilde{h}_1^l, \ldots, \tilde{h}_I^l)$ denotes the set of hidden states in the l-th layer for X.

The *decoder* computes the hidden state $\tilde{z}_j^l$ of the l-th layer using

$$z_j^l = \text{MultiHeadAttn}(\tilde{z}_j^{l-1}, \widetilde{Z}_{\leq j}^{l-1}, \widetilde{Z}_{\leq j}^{l-1}), \tag{11}$$

$$\hat{z}_j^l = \text{MultiHeadAttn}(z_j^l, \widetilde{H}^L, \widetilde{H}^L), \tag{12}$$

$$\tilde{z}_j^l = \text{FNN}(\hat{z}_j^l), \tag{13}$$

where $\tilde{z}_j^0$ is the embedding of token y_{j-1} and $\tilde{z}_1$ is the state for the special token <sos>. $\widetilde{Z}_{\leq j}^{l-1}$ denotes a set of hidden states $(\tilde{z}_1^{l-1}, \ldots, \tilde{z}_j^{l-1})$.

2.2 Coverage Mechanism

Tu et al. (2016) suggested that $a_{i,j}$ in Eq. (6) describes the probability of generating target word y_j at time step j from source word x_i. Based on $a_{i,j}$, they proposed coverage value $C_{i,j}$ to denote the translated ratio of x_i at time step j in a machine translation task. $C_{i,j}$ is calculated as follows:

$$C_{i,j} = C_{i,j-1} + \tfrac{1}{\Phi_i} a_{i,j} = \tfrac{1}{\Phi_i} \sum_{k=1}^{j} a_{i,k}, \tag{14}$$

where Φ_i is a weight to denote the number of target words expected to be translated from x_i. Tu et al. (2016) assumed that some source words might be unnecessarily translated multiple times (over-generation) while some source words might be mistakenly untranslated (under-generation)

if a model lacked coverage information. Based on this assumption, Tu et al. (2016) introduced the last coverage $C_{i,j-1}$ to the global attention to calculate $a_{i,j}$ using score function $f(h_i, z_j, C_{i,j-1})$ in Eq. (6). Although using this coverage mechanism clearly shows the used ratio of x_i at each decoding step and produces a more accurate alignment to guide target generation, using it for the multihead self-attention mechanism is intractable owing to the concatenation in Eq. (8), and stacked sublayers in both the *encoder* and *decoder*. Owing to this limitation, we propose a generic method for supervising the attention distribution in the seq2seq model.

2.3 Autoencoder

Given source X, the autoencoder encodes X in representation $h(X)$ and reconstructs X from $h(X)$. This reconstruction by the autoencoder with parameter θ can be written as $p(X|h(X);\theta)$. Let $X^{'}$ be the reconstructed X. The autoencoder utilizes loss function $\ell(X, X^{'})$ to measure the reconstruction errors. L_2 norm loss is a widely used loss function as $\ell(X, X^{'})$ and can be computed as follows:

$$\ell(X, X^{'}) = \|X - X'\|_2^2, \tag{15}$$

where X is a vector.

Owing to its *encoder-decoder* architecture, the seq2seq model has been utilized as an autoencoder in several natural language generation tasks (Ma et al. 2018; Kiyono et al. 2018; Liu et al. 2019). Ma et al. (2018) proposed an autoencoder using an LSTM-based seq2seq architecture to encode and reconstruct target summaries in the text summarization task. During training, the autoencoder was combined with an LSTM-based seq2seq text summarization model by sharing the decoder to supervise the training of the summarization model. Their experimental results demonstrated that using the autoencoder assisted the LSTM-based seq2seq summarization model in receiving additional gains for the ROUGE score. To supervise the training step, Kiyono et al. (2018) considered the source word frequency as input X and utilized Eq. (15) to measure the distance between the input and reconstructed word frequencies in the text summarization task. Their experimental results demonstrated that the supervised global attention-based seq2seq model generated fewer repetitions than the unsupervised model. However, their assumption required a source sequence longer than the target, which constrained the application of their proposal to text-generation tasks other than text summarization. Liu et al. (2019) also utilized an LSTM-based seq2seq model as the text autoencoder to extract representation from the target text and supervise the table-to-text generation task. Their experimental results demonstrated that the supervised table-to-text model achieved higher BLEU and ROUGE scores than the

unsupervised model. Instead of supervising the training, Luo et al. (2018) utilized two autoencoders to separately reconstruct the inputs and responses to learn semantic representations in the dialogue generation task. After mapping the semantic representations of the input sentence and response, their proposal generated a fluent and coherent response than the seq2seq model of Sutskever et al. (2014).

These studies motivated us to deploy an autoencoder to learn the semantic representations of the source word frequency and supervise model training in natural language generation tasks.

2.4 Repetition Types

Fu et al. (2021) defined that the consecutive repetitions required at least two adjacent identical fragments, in which the first fragment was the counterpart and the rest fragments were repetitions, as shown in Figure 1. Based on this definition, we define nonconsecutive repetitions as those that require at least two detached identical fragments. Xu et al. (2022) mentioned three types of consecutive repetitions: word-, phrase-, and sentence-level, and used the rules below for calculation without considering the word matching between model predictions and gold references.

Given sequence Y, the probability of word-level consecutive repetitions is calculated according to $\frac{1}{J-1}\sum_{j=2}^{J}\mathbb{1}(y_j = y_{j-1})$, where $\mathbb{1}$ is an indicator function. Assuming a k-word phrase, the probability of phrase-level consecutive repetitions is computed according to $\frac{1}{J-2k+1}\sum_{j=2k}^{J}\mathbb{1}((y_{j-k+1},\ldots,y_j) = (y_{j-2k+1},\ldots,y_{j-k}))$. To distinguish sentence-level consecutive repetitions, sequence Y is split into $N+1$ subsentences by ".!?". Let $Y = (\mathrm{s}^0,\ldots,\mathrm{s}^N)$ denote the set of split subsentences, the probability of sentence-level consecutive repetitions is calculated according to

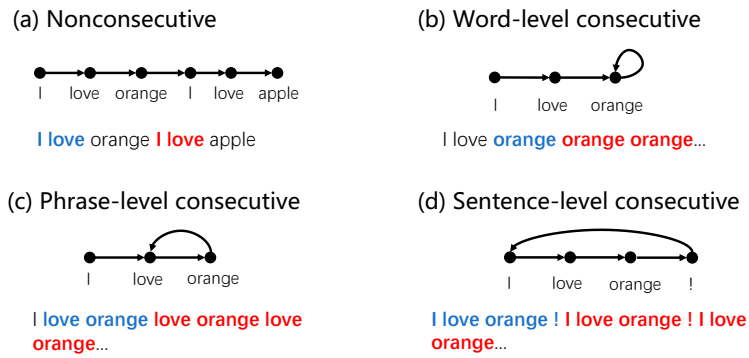

Figure 1 Examples of repetition types. Red and blue indicate the repetitions and their counterparts, respectively.

$\frac{1}{N}\sum_{i=1}^{N}\mathbb{1}(s^i = s^{i-1})$. We followed Xu et al. (2022) to compute word-, phrase-, and sentence-level consecutive repetitions. Considering the usages of "..." and "!!!" in the natural text, the research limited the length of the subsentence by at least one word (ignoring ".!?"). For example, the sequence "I like apple.i.e.e.e." would be split into subsentences ("I like apple," "i," "e," "e," "e," ""), and only "I like apple" was regarded as an valid subsentence. We renamed the sentence-level consecutive repetition as subsentence-level consecutive repetition to distinguish it from the repeat (sentence-level), which is defined in Section 4.3. Additionally, we evaluate the probability of total repetitions according to $1 - |\text{unique words}|/J$.

We report the average probability over the entire corpus and compare the probabilities of the predicted texts with gold references for evaluation.

3 RRM

3.1 Overview

An overview of the RRM is presented in Figure 2. Here, we employed the structure of the transformer for both the *encoder* and *decoder*. Let $\tilde{x}$ denote the source-side sentence representation of source sentence X; $\tilde{q}$ denote the reconstructed $\tilde{x}$. According to the coverage mechanism, supervising the ratio of source tokens used for the seq2seq model could produce an accurate alignment and guide target generation. Inspired by Luong et al. (2015a) and Kiyono et al. (2018), the RRM considers $\tilde{x}$ as the correct representation of X and attempts to reconstruct $\tilde{x}$ on the target side as an autoencoder. We assume that $\tilde{x}$ includes the word frequency of X and $\tilde{q}$ includes the

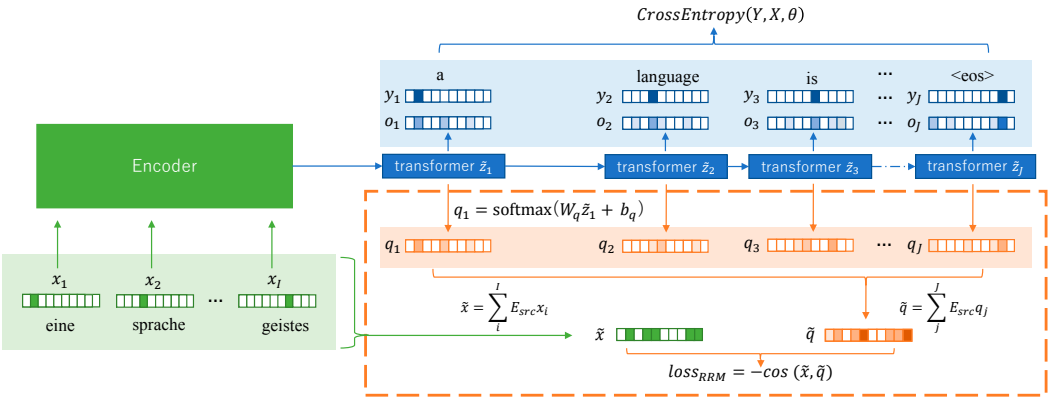

Figure 2 Overview of a transformer-based *encoder-decoder* model with the RRM. The part inside a dashed rectangular box represents the RRM.

used $\tilde{x}$. By sharing the model parameters of the RRM with the seq2seq model, the source tokens used in the seq2seq model are supervised during training without considering the global attention mechanism, as in Tu et al. (2016). The seq2seq model with parameter θ then predicts target-side sequence Y and $\tilde{x}$. This prediction can be expressed as follows:

$$p(Y, \tilde{x}|X; \theta) = p(\tilde{x}|Y, X; \theta)p(Y|X; \theta). \tag{16}$$

Conditional probability $p(\tilde{x}|Y, X; \theta)$ prevents either over- or under-generation of Y by reconstructing the representation of the source context until the decoding step ends. $p(\tilde{x}|Y, X; \theta)$ can be simplified to $p(\tilde{x}|X; \theta)$ if $\tilde{q}$ does not depend on Y. As $p(Y|X; \theta)$ is predicted by the seq2seq model, as shown in Eq. (1), we provide details of $p(\tilde{x}|Y, X; \theta)$ in the section below.

3.2 Prediction of Source Side Context

The SPM proposed by Kiyono et al. (2018) considers the count-based discrete representations of X as the input and reconstructs the representations on the target side to supervise the seq2seq model to reduce the number of repetitions. However, this discrete vector failed to recognize a word by its tokenized subwords and may mislead the reconstruction; for instance, the word "All" and its tokenized subwords "Al l." Furthermore, the SPM requires a longer X than Y when calculating Eq. (15), which constrains its application. Instead of using count-based discrete representations, as in Kiyono et al. (2018), we incorporated continuous representations for both the source and target sides to capture deeper semantic relations (Mikolov et al. 2013). We assume $p(\tilde{x}|Y, X; \theta)$ is proportional to the similarity between the representations of source sentence X before and after being encoded and decoded as follows:

$$p(\tilde{x}|Y, X; \theta) \propto \exp(\alpha(\cos(\tilde{x}, \tilde{q}))), \tag{17}$$

where α denotes the scaling factor used in the experiments.

Next, we explain the representations of source sentence X in the source and target sides. Let V_s denote the source vocabulary. We define the indicator vector for the presence of source tokens as $x_i \in \{0, 1\}^{|V_s|}$, where x_i denotes the i-th token in X. The source-side representation $\tilde{x}$ of the source sentence is defined as follows:

$$\tilde{x} = \sum_{i}^{I} E_{src} x_i, \tag{18}$$

where $E_{src} \in R^{H \times |V_s|}$ is the word-embedding matrix for the source vocabulary and H is the embedding size.

To reconstruct $\tilde{x}$, we define the target-side representation $\tilde{q}$ of the source sentence as follows:

$$\tilde{q} = \sum_{j}^{J} E_{src} q_j, \tag{19}$$

where $q_j \in R^{|V_s|}$ represents the probability distribution over source vocabulary V_s at the j-th decoding step, which is calculated as follows:

$$q_j = \mathrm{softmax}(W_q \tilde{z}_j + b_q), \tag{20}$$

where W_q is a weight matrix and b_q is a bias term. The softmax layer was used only in the training step.

3.3 Objective Function

Considering the negative log-likelihood of Eq. (16) , we induce objective function G_t as follows:

$$G_t = \sum_{(X,Y)\in\mathcal{D}} \{-\log p(Y|X;\theta) - \alpha(\cos(\tilde{x}, \tilde{q}))\}, \tag{21}$$

where $\mathcal{D}$ denotes a parallel training corpus.

4 Experiments

4.1 Datasets

To investigate the performance of the RRM, we evaluated the model performance on two natural language generation tasks: machine translation and response generation. As a directed generation task, machine translation constrains the output as a transformation of the input. Conversely, the response generation without such constraints is an open-ended generation task (Holtzman et al. 2020).

We first used the IWSLT 2014 German-to-English translation dataset (Cettolo et al. 2014) to evaluate the proposed method. The dataset was split into 160k/7k/7k sentences for training, validation, and testing, respectively. As Cho et al. (2014) reported that seq2seq models tend to produce few unknown tokens and yield high BLEU scores for short sentences in neural machine translation tasks, we assumed that longer sentences could contain more repetitions because of the difficulty in aligning the attention between more tokens using the seq2seq model, and our proposal could perform better for longer sentences. Therefore, we divided the test data into three parts: *short*, *medium*, and *long*. In *short* with 4927 pairs, the source contained less than 25 byte pair encoding (BPE) (Sennrich et al. 2016) tokens. In *medium* with 1524 pairs, the

source contained 26–50 BPE tokens. In *long* with 299 pairs, the source contained more than 50 BPE tokens. Additionally, we evaluated our method on the WMT 2014 English-to-German translation dataset (Bojar et al. 2014). We used the script[2] from fairseq (Ott et al. 2019) to follow the setup of Gehring et al. (2017) to preprocess the dataset. The preprocessed dataset was split into 3.9M/3.9k sentences for training and validation, respectively. The newstest2014 dataset[3] with 3k sentences was used for the test. We split the newstest2014 set into 1661, 1173, and 169 pairs for the *short*, *medium*, and *long* parts, respectively.

We used the PERSONA-CHAT (Zhang et al. 2018) dataset for response generation task. This is the official dataset of the conversational intelligence challenge 2 (ConvAI2)[4] used for testing chatbots. It contains 164k/15k/15k utterances (corresponding to 10k/1k/1k dialogs) for training, validation, and testing, respectively. It also contained the corresponding persona information for each dialog. An example is presented in Table 2.

4.2 Compared Methods

To investigate the effectiveness of the proposed module, we compared the experimental results between models with and without the RRM on top of the baseline models.

The vanilla transformer architecture (Vaswani et al. 2017) with different hyperparameter settings was utilized as the baseline for machine translation tasks. We used the model of Fonollosa et al. (2019) as the baseline for the German-to-English task. In contrast to the transformer structure, the neural network in Fonollosa et al. (2019) discards the independent *encoder* to directly calculate the cross attention between the source and target tokens to predict the target sentence, which was similar to the autoregressive language model, generative pre-trained transformer (GPT) (Radford et al. 2018). Hereafter, we denote this model as LocalJoint.

We considered the most effective model (Wolf et al. 2019) in ConvAI2 as the baseline for the response generation task. Wolf et al. (2019) adopted a GPT-based *encoder* and a 12-layer transformer *decoder*, and concatenated the persona information, up to two turns of historical utterances, and the query (the utterance) together as an input sequence. We denote this model as TransferTransfo. We also utilized a pre-trained transformer model, BART (Lewis et al. 2020), as the baseline for the response generation task. Previously concatenated sequences were used as the input sequences for the BART.

[2] https://github.com/facebookresearch/fairseq/blob/main/examples/translation/prepare-wmt14en2de.sh

[3] https://nlp.stanford.edu/projects/nmt/

[4] http://convai.io/

Response generation		
	Persona	i have two children that are in their kindergarten . i am 34 years old .
	Information	i am an environmentalist . i have been working as a real estate agent for the past 12 years . i went to college at university of mississippi .
1st Turn	Query	hi , want to be my friend ?
	Reference	hello , sure why not ! how old are you ? i just turned 34
	TransferTransfo	sure , **i 'm a real estate agent**.
	+RRM	sure , what do you do for a living ?
2nd Turn	Query	whoa . i am only 12 . what is it like to be old ?
	Reference	my 2 kiddos are in kindergarten , they want to graduate at mississippi university like me
	TransferTransfo	i 'm 35 , so i 'm not sure . i 'm a real estate agent
	+RRM	i 'm a real estate agent , what about you?
3rd Turn	Query	that is cool i guess . are they twins
	Reference	no . only 12 ? so no job yet . i have done real estate 12 years now
	TransferTransfo	yes, **they are twins** , they are twins
	+RRM	they 're , they 're in their early 20s .

Table 2 Sample responses generated by various models. Underline indicates repetitions that contain more than two words and **bold** indicates their counterparts. A turn is a pair of a query (an utterance) and its response (either a reference or a generation). In this example, a sequence of the three turns consist of a part of a dialog. When a model tries to output a response to the third query, the first and second turns are the historical utterances, and it receives the concatenation of the persona information, the historical utterances and the third query as an input. α was fixed to 0.3, which yielded the least repeat (sentence-level) under 1-gram on the validation dataset.

4.3 Evaluation Metrics

We used tokenized BLEU (Papineni et al. 2002), Meteor (Denkowski and Lavie 2014), and repeat (Kiyono et al. 2018) for the machine translation tasks. The repetition is defined as follows: following the definitions in Kiyono et al. (2018) and Fu et al. (2021), we believe that a model causes repetition if it outputs the same token more than once. For each pair of generated translations and their corresponding references in the dataset, because we considered that some tokens might occur more than once in the reference, the repeat was computed by subtracting the frequency of tokens in the reference from the frequency of tokens that occur more than once in the generated translation (Kiyono et al. 2018). Additionally, we calculate the probability of each type of repetition, as described in Section 2.4.

For the response generation task, we used official evaluation metrics, F1 and Perplexity. The

official method offered by ParlAI (Miller et al. 2017) ignores words {a, an, the} and punctuations when computing F1. To compute perplexity, we followed Wolf et al. (2019) to indirectly predict word probability based on the ratio of probabilities of subwords because the official method requires a BPE vocabulary of 19304 tokens. Unlike the machine translation task, the response generation task has no fixed answers. Therefore, in this task, we ignored the reference sequence when computing the repeats. For each generated sequence, the repeat was computed by subtracting one from the frequency of tokens that occurred more than once in the generated sequence. We calculated repeat scores under an n-gram setting at sentence- and dialog-level while ignoring the words {a, an, the} and punctuations. We calculated the repeats only for each generated response at the sentence-level. We calculated the repeat with the concatenation of a sequence of the generated responses in a dialog at the dialog-level. We also calculated the probability of word-, phrase-level consecutive, and total repetitions with each generated response while ignoring the words {a, an, the} and punctuations.[5]

We used a paired t-test to evaluate whether the differences in the repeat score and probability of each type of repetition were significant. †, ‡, and § indicate that the difference between the baseline model and the baseline+RRM is significant, with p-values of < 0.01, < 0.05, and < 0.1, respectively.

4.4 Hyperparameters

The hyperparameters used in each model are listed in Table 3. We followed the experimental settings of Fonollosa et al. (2019) and utilized their public code[6] to reproduce and train the LocalJoint and LocalJoint+RRM models. We used transformer architectures "transformer_iwslt_de_en" and "transformer_wmt_en_de" published by fairseq[7] for the German-to-English and English-to-German translation tasks, respectively. Following previous studies,[8,9] in the decoding steps, we used beam search (Wu et al. 2016) with beam sizes of 5 and 4 for the German-to-English and English-to-German translation tasks, respectively. To tune scaling factor α for LocalJoint+RRM and transformer+RRM, we set α to {1, 0.3, 0.2, 0.05, 0.01} and selected α with repeats as the evaluation metric for the validation dataset.

For the response generation task, we used the experimental settings of Wolf et al. (2019) and

[5] The probability of subsentence-level consecutive repetitions was not calculated because we ignored punctuations.

[6] https://github.com/jarfo/joint

[7] https://github.com/facebookresearch/fairseq/blob/main/fairseq/models/transformer/transformer_legacy.py

[8] https://github.com/facebookresearch/fairseq/tree/main/examples/translation

[9] https://github.com/facebookresearch/fairseq/issues/346

Task	German-to-English		English-to-German	Response generation	
Model	Transformer	LocalJoint	Transformer	BART	TransferTransfo
Optimizer	adam				
Adam β_1	0.9				
Adam β_1	0.98			0.999	
Adam eps	$1e-9$			$1e-6$	
Weight decay	0.0001			0	
Learning rate	0.0005	0.001	0.0007	$6.25e-5$	
Learning rate schedule	Inverse square root			Linear decay	
Batch size	8192 tokens	4000 tokens	32768 tokens	32 sequences	
Warm up steps	2k	4k		0	
Training steps	42.5k	85k	95k	1 epoch	
Attention layer	6	14	6		12
Attention head	4		8	12	12
Dropout	0.3		0.1		
Hidden size	512	256	512	768	
Feed-forward expansion size	1024		2048	3072	
Vocab size	31K		43k	50k	40k
Tokenizer	Byte pair encoder				
Max src/tgt sentence length	1024				512

Table 3 List of hyperparameters used for each model.

utilized their public code[10] to reproduce and train the TransferTransfo and TransferTransfo+RRM models. We used the HuggingFace[11] (Wolf et al. 2020) to reproduce BART. In the decoding step, we utilized the top 20 samplings (Fan et al. 2018) before selecting four beams via a beam search. To tune scaling factor α for TransferTransfo+RRM and BART+RRM, we set α to {1, 0.3, 0.2, 0.05, 0.01}. Because the RRM was designed to reduce repetitions at the sentence-level, we selected α with repeat (sentence-level) under 1-gram as the evaluation metric for the validation dataset.

The results of all baseline and baseline+RRM models were averaged over three runs using random seeds.

4.5 German-to-English Results

The experimental results for the German-to-English translation task are presented in Table 4. The results indicate that combining RRM ($\alpha = 0.3$) with LocalJoint improves the repeat, BLEU, and Meteor scores. Utilizing the RRM ($\alpha = 0.01$) for the transformer also mitigated the repeat

[10] https://github.com/huggingface/transfer-learning-conv-ai
[11] https://github.com/huggingface/transformers

Data	Model	Repeat	BLEU	Meteor
All	LocalJoint (Fonollosa et al. 2019)	—	35.70	—
	LocalJoint	1.244	35.61	35.76
	+RRM	**1.229**	**35.71**	**35.77**
	Transformer	1.570	33.92	34.92
	+RRM	1.509†	34.05	34.84
Short	LocalJoint	**0.552**	37.41	**36.83**
	+RRM	0.554	**37.47**	36.81
	Transformer	0.701	35.81	36.03
	+RRM	0.655†	35.95	35.94
Medium	LocalJoint	2.484	34.28	34.91
	+RRM	**2.467**	**34.38**	**35.00**
	Transformer	3.112	32.79	34.19
	+RRM	3.036‡	32.74	34.10
Long	LocalJoint	6.371	33.11	**34.12**
	+RRM	**6.036**	**33.36**	33.99
	Transformer	8.032	30.17	32.81
	+RRM	7.792	30.85	32.75

Table 4 Experimental results on the IWSLT 2014 De-En test dataset. α was fixed to 0.3 and 0.01 for LocalJoint and transformer, respectively, which yielded the least repeat on the validation dataset. **Bold** indicates the least repeat, highest BLEU and Meteor scores with respect to different length settings.

score and increased the BLEU score; however, it decreased the Meteor score. We then compared the experimental results for *short*, *medium*, and *long* to investigate the effectiveness of the RRM at different source sentence lengths. Similar to the results from Cho et al. (2014), compared with the *short* sentences, all models tended to have lower BLEU scores and more repetitions for *long* sentences. Conversely, the RRM performed relatively well for longer sentences. This reduces the number of repetitions and improves more BLEU on top of the baselines for longer sentences. Although the RRM showed limited effect in reducing the number of repetitions for *short* sentences, it assisted LocalJoint in reducing the repeat score by 0.335 points and improved the BLEU score by 0.25 points for *long* sentences. Additionally, the RRM assisted the transformer in reducing the repeat score by 0.24 points and improving the BLEU score by 0.68 points for *long* sentences. These results indicate the effectiveness of the RRM for long sentences.

The statistics for different repetitions on the IWSLT 2014 De-En test dataset are summarized in Table 5. By comparing the probability of total and consecutive repetitions, we suggest that

Data	Model	Word	Phrase #2	Phrase #3	Phrase #4	Subsentence	Total
All	Source	0.10	0.06	0.01	0.03	0.01	8.78
	Reference	0.07	0.05	0.04	0.02	0.00	9.78
	LocalJoint	0.02	**0.07**	0.02	**0.03**	**0.01**	**9.84**
	+RRM	0.03§	0.08	0.02	0.04	**0.01**	9.85
	Transformer	**0.05**	0.09	0.03	0.05	**0.01**	10.59
	+RRM	0.03†	0.09	**0.04**	0.06	0.02	10.44†
Short	Source	0.09	0.04	0.01	0.03	0.01	5.52
	Reference	0.06	0.07	0.03	0.00	0.01	6.67
	LocalJoint	0.02	0.06	0.02	**0.03**	**0.01**	6.49
	+RRM	**0.04**§	0.06	0.02	0.04	**0.01**	6.49
	Transformer	**0.04**	0.08	**0.03**	0.05	**0.01**	6.98
	+RRM	0.03‡	**0.07**	0.02	0.05	**0.01**	**6.83**†
Medium	Source	0.12	0.10	0.00	0.03	0.03	16.03
	Reference	0.10	0.09	0.01	0.04	0.00	17.43
	LocalJoint	0.02	0.08	0.04	0.01	0.02	17.18
	+RRM	0.03	**0.09**	**0.03**	**0.03**	0.02	**17.24**
	Transformer	**0.05**	0.10	**0.03**	**0.03**	**0.01**	18.56
	+RRM	0.04	0.10	0.07	**0.03**	0.03	18.42
Long	Source	0.10	0.10	0.04	0.03	0.00	26.58
	Reference	0.14	0.10	0.04	0.03	0.00	27.19
	LocalJoint	0.03	0.26	**0.05**	0.09	0.08	27.59
	+RRM	0.01§	**0.24**	0.08§	**0.06**	**0.03**	**27.47**
	Transformer	**0.13**	0.32	0.09	0.21	0.12	29.40
	+RRM	0.08§	0.36	0.11	0.23	0.16	29.22

Table 5 Probability (%) of word-, phrase (number of words)-, subsentence-level consecutive, and total repetitions on the IWSLT 2014 De-En test dataset. α was fixed to 0.3 and 0.01 for LocalJoint and transformer, respectively, which yielded the least repeat on the validation dataset. **Bold** indicates the numbers closest to the reference scores with respect to different length settings.

in natural text (referring to source and reference texts), the most common type of repetition is nonconsecutive. The RRM exhibited no effect on the consecutive fragments, which could be because of the limited number of consecutive repetitions generated by the seq2seq model or in the natural text. For the total repetitions, the LocalJoint and transformer with RRM obtained probabilities closer to the reference for *medium* and *long* sentences than those without RRM. These results also indicate the effectiveness of the RRM for longer sentences. Additionally, the table illustrates a small gap ($< 1\%$) in the probability of total repetition between the reference and predicted texts over the entire corpus. However, when considering word matching, the

Source	ich hielt meinen üblichen vortrag , und danach sah sie mich an und sagte : >> mhmm . mhmm . mhmm . <<
Reference	i gave her my whole rap , and when i finished she looked at me and she said , " mmm mmm mmm . "
LocalJoint	i gave my usual talk , and then she looked at me and she said , " mhmm . mhmm . mhmm . "
+RRM ($\alpha = 0.3$)	i gave my usual talk , and then she looked at me and she said , " mhmm . mhmm . mhmm . "
Transformer	i gave my usual talk , and after that , she looked at me and she said , " hmm . hmm . "
+RRM ($\alpha = 0.01$)	i gave my usual talk , and then she looked at me afterwards , and she said , " hmm . hmm . "

Table 6 Sample translations on the IWSLT 2014 De-En test dataset. Words "mhmm" and "hmm" are misspelled "mmm" and could be calculated in the repeat score.

neural models achieved a high repeat score (> 6 for *long* sentences), as summarized in Table 4. We established that the neural models could correctly predict word frequency in most cases but misspelled the word. Thus, these models achieved a closer probability of repetitions to the reference; however, they had a high repeat score. An example of predicted sentences containing misspelled words for "mmm" is presented in Table 6.

The top and bottom 20 words based on the degree of repeat reduction using the LocalJoint + RRM ($\alpha = 0.3$) are listed in Table 7. These results indicate that the LocalJoint+RRM reduced repetitions for high frequency words, whereas it exhibited no effect of reducing repetitions for "." and "'s."

4.6 English-to-German Results

The experimental results for the English-to-German translation task are presented in Table 8. Similar to the results in Table 4, using the RRM ($\alpha = 0.3$) for the transformer reduced the repeat score by 0.115 points and increased the BLEU score by 0.45 points over the entire corpus. However, the transformer+RRM achieved a lower Meteor score than the transformer only. Compared with the *short* and *medium* sentences, using RRM could reduce more repeat scores by 0.308 points and improve more BLEU scores by 0.67 points for *long* sentences.

Figure 3 shows a positive slope between the repeat score and the length of the source sentence. In the WMT 2014 German-to-English test dataset, the slope of the transformer is steeper when the source sentence is longer, indicating a higher probability of generating repetitions for longer sentences. Furthermore, on the two machine translation datasets, using RRM to supervise the baseline relieved the steep slope.

The statistics for different repetitions on the WMT 2014 En-De test dataset are summarized in Table 9. The table shows a low probability ($< 0.1\%$) of consecutive repetitions in both natural and predicted texts. In particular, the natural text contained no phrase- and subsentence-level consecutive repetitions. The probability gap of word-level consecutive repetitions between reference and transfomer was only 0.01% over the entire corpus, indicating a limited number

Word	Frequency	Frequency Rank	Sum of Repeat		Reduced Repeat
			LocalJoint	+RRM	
you	45794	12	50	31	19
of	73774	6	76	62	14
a	67343	7	80	67	13
on	16749	27	18	7	11
they	21064	19	31	22	9
and	96381	4	76	68	8
in	50081	10	56	48	8
do	11485	39	10	4	6
how	7722	59	9	3	6
to	78411	5	73	68	5
where	4913	91	6	1	5
could	4503	98	8	3	5
through	2617	141	7	2	5
is	41409	14	35	31	4
"	22866	18	15	11	4
these	9016	49	4	0	4
their	6187	75	11	7	4
the	134603	3	129	126	3
one	11115	40	5	2	3
would	6084	77	4	1	3
belief	120	1868	0	2	−2
generally	101	2206	0	2	−2
determine	86	2497	0	2	−2
defined	85	2523	0	2	−2
colleague	63	3210	0	2	−2
eliminating	20	7370	0	2	−2
celestial	15	8869	0	2	−2
joints	15	8870	0	2	−2
mutilated	11	10851	0	2	−2
anatomic	5	17811	0	2	−2
humiliated	5	17812	0	2	−2
for	18902	22	7	10	−3
can	15244	29	11	14	−3
people	10653	42	8	11	−3
someone	695	415	1	4	−3
river	146	1572	1	4	−3
compromised	12	10263	0	3	−3
had	6648	70	6	11	−5
's	36495	15	39	47	−8
,	191365	1	211	224	−13

Table 7 Top and bottom 20 words based on the degree of repeat reduction. They are listed in descending order of the repeat reduction by +RRM ($\alpha = 0.3$) on top of LocalJoint for the IWSLT 2014 De-En test dataset at *long* length. Frequency denotes the word frequency in the training dataset, and its rank is denoted as frequency rank.

Data	Model	Repeat	BLEU	Meteor
All	Transformer (Vaswani et al. 2017)	—	27.3	—
	Transformer	1.503	26.33	**29.07**
	+RRM	**1.388**†	**26.78**	28.93
Short	Transformer	0.558	25.83	**28.61**
	+RRM	**0.528**‡	**25.99**	28.48
Medium	Transformer	2.337	26.32	**29.17**
	+RRM	**2.130**†	**26.91**	29.06
Long	Transformer	4.988	27.67	**30.02**
	+RRM	**4.680**†	**28.34**	29.79

Table 8 Experimental results on the WMT 2014 En-De test dataset. α was fixed to 0.3, which yielded the least repeat on the validation dataset. **Bold** indicates the least repeat, and highest BLEU and Meteor scores with respect to different length settings.

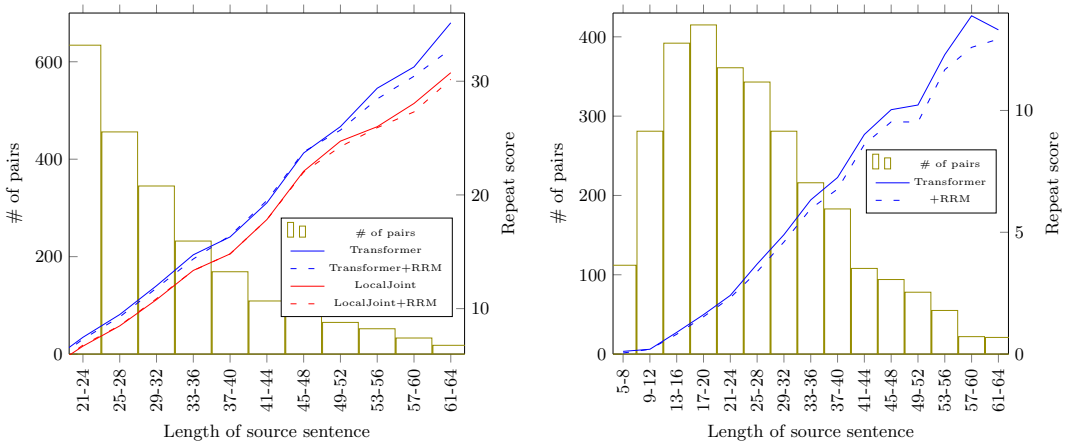

Figure 3 Distribution of baselines with and without RRM on the IWSLT 2014 De-En (left) and WMT 2014 En-De (right) test datasets.

of redundant consecutive repetitions generated by the transformer. For the total repetitions, the transformer supervised by the RRM could generate fewer repetitions than the unsupervised transformer.

4.7 Response Generation Results

The experimental results of the response generation task are presented in Table 10. The RRM for TransferTransfo reduced the repeat scores by 0.056 (sentence-level) (1-gram) and 0.471

Data	Model	Word	Phrase #2	Phrase #3	Phrase #4	Subsentence	Total
All	Source	0.01	0.00	0.00	0.00	0.00	8.00
	Reference	0.03	0.00	0.00	0.00	0.00	5.66
	Transformer	**0.04**	**0.01**	**0.00**	**0.00**	**0.00**	7.47
	+RRM	**0.04**	**0.01**	**0.00**	**0.00**	**0.00**	**7.20**†
Short	Source	0.00	0.00	0.00	0.00	0.00	4.19
	Reference	0.02	0.00	0.00	0.00	0.00	2.70
	Transformer	**0.02**	0.01	**0.00**	**0.00**	**0.00**	3.86
	+RRM	**0.02**	**0.00**	**0.00**	**0.00**	**0.00**	**3.73**†
Medium	Source	0.01	0.00	0.00	0.00	0.00	11.82
	Reference	0.04	0.00	0.00	0.00	0.00	8.61
	Transformer	**0.06**	**0.00**	**0.00**	**0.00**	**0.00**	11.16
	+RRM	**0.06**	0.01‡	**0.00**	**0.00**	**0.00**	**10.73**†
Long	Source	0.01	0.00	0.00	0.00	0.00	18.98
	Reference	0.03	0.00	0.00	0.00	0.00	14.18
	Transformer	**0.04**	**0.01**	**0.00**	**0.00**	**0.00**	17.35
	+RRM	**0.04**	**0.01**	**0.00**	**0.00**	**0.00**	**16.91**†

Table 9 Probability (%) of word-, phrase (number of words)-, subsentence-level consecutive, and total repetitions on the WMT 2014 En-De test dataset. α was fixed to 0.3, which yielded the least repeat on the validation dataset. **Bold** indicates the numbers closest to the reference scores with respect to different length settings.

(dialog-level) (1-gram) points. The results indicate that combining the RRM with TransferTransfo improves F1 and repeat, whereas the performance of the RRM in reducing perplexity is limited. We suppose that there are two reasons for the performance of the RRM on perplexity. First, the probability calculation method proposed in Wolf et al. (2019) is indirect. Second, the responses were not fixed for a given source. The pre-trained seq2seq model, BART, generated fewer sentence-level repetitions and achieved a higher F1 score of 21.14 but a higher perplexity of 85.48 than TransferTransfo. This high perplexity may have also been caused by indirect probability calculations. Using the RRM for BART mitigates repeats by 0.013 (sentence-level) (1-gram) and 0.361 (dialog-level) (1-gram) points. These results indicate the effectiveness of the RRM for the TransferTransfo model to reduce the number of sentence-level repetitions than BART.

We conducted extensive experiments to investigate whether the RRM has the potential to reduce the number of repetitions by considering the following conditions: First, beam search is an optimized decoding method that generates fewer repetitions than greedy decoding, which may limit the performance of the RRM. We investigated whether decoding methods influenced the

Model	Repeat					Perplexity	F1
	1-gram	2-gram	3-gram	4-gram	5-gram		
Sentence-Level							
Source	0.452	0.033	0.002	0.000	0.000	—	—
Reference	0.482	0.030	0.002	0.000	0.000	—	100.00
TransferTransfo (Wolf et al. 2019)	—	—	—	—	—	16.28	19.50
TransferTransfo	0.755	0.244	0.107	0.056	0.025	**16.31**	18.22
+RRM	0.699†	0.210†	0.090†	**0.045**†	**0.018**†	16.33	18.36
BART	0.675	0.189	**0.077**	0.046	0.024	85.48	**21.14**
+RRM	**0.662**§	**0.186**	**0.077**	0.048	0.027	82.37	21.13
Dialog-Level							
Source	20.125	3.112	0.474	0.118	0.051	—	—
Reference	20.840	3.278	0.434	0.091	0.027	—	—
TransferTransfo	28.423	14.319	7.786	4.822	2.800	—	—
+RRM	**27.952**†	**13.982**†	**7.605**§	**4.743**	**2.791**	—	—
BART	28.817	14.244	7.884	5.024	3.013	—	—
+RRM	28.456†	13.969†	7.765	4.936	2.958	—	—

Table 10 Experimental results on the PERSONA-CHAT test dataset. α was fixed to 0.3 for both baselines, which yielded the least repeat (sentence-level) under 1-gram on the validation dataset. **Bold** indicates the least repeat, perplexity, and highest F1 scores for different levels.

performance of the RRM by comparing beam search with greedy decoding.

Second, Wolf et al. (2019) used persona information and historical utterances as input sequences for their model to generate a response that differed from the history and contained part of the persona information. However, in the task, our model shares its vocabulary between the source and target sides; in Eq. (17), we utilize the cosine similarity to force $\tilde{q}$ to be similar to $\tilde{x}$, which may make the generated response similar to the input sequence. Therefore, we investigated whether using historical utterances in the input sequence during inference could affect the performance of the RRM. "w/o history" indicates the case where the historical utterances were not used for an input sequence during inference.

Third, historical utterances may contain part of the persona information, which can cause additional repetitions in Eq. (18). The RRM could be misled into producing more repetitions at the sentence-level and hence more repetitions at the dialog-level. Therefore, we investigated whether using the persona information and historical utterances in Eq. (18) during training affects the performance of the RRM. The *full* setting indicates the usage of the persona information, historical utterances, and query as a source in Eq. (18) during training, whereas the *part* setting

indicates the usage of only the query. We also set *divide* to divide the input sequence in Eq. (18) into three parts: $\tilde{x}_p$, $\tilde{x}_h$, and $\tilde{x}_l$, depending on the personal information, historical utterances, and query, and uses the corresponding W_{q_p}, W_{q_h}, and W_{q_l} in Eq. (20) to compute $\tilde{q}_p$, $\tilde{q}_h$, and $\tilde{q}_l$, respectively. Subsequently, the average cosine similarity was calculated between each divided $\tilde{x}$ and $\tilde{q}$.

The results of the extensive experiments are presented in Table 11. Clearly, when using beam

Decode	Model	Repeat					Perplexity	F1
		1-gram	2-gram	3-gram	4-gram	5-gram		
				Sentence-Level				
Beam	TransferTransfo (Wolf et al. 2019)	—	—	—	—	—	16.28	19.50
	TransferTransfo	0.755	0.244	0.107	0.056	0.025	**16.31**	18.22
	+RRM ($\alpha = 0.3, full$)	**0.699**†	**0.210**†	**0.090**†	0.045†	0.018†	16.33	**18.36**
	+RRM ($\alpha = 1, divide$)	0.746	0.248	0.114	0.063‡	0.028	16.34	18.20
	+RRM ($\alpha = 0.2, part$)	0.703†	0.212†	**0.090**†	**0.043**†	**0.017**†	16.40	18.27
Beam	TransferTransfo w/o history	0.902	0.336	0.135	0.067	0.026	**17.96**	**17.30**
	+RRM ($\alpha = 0.05, full$)	0.842†	0.275†	0.100†	**0.043**†	**0.014**†	18.04	17.14
	+RRM ($\alpha = 1, divide$)	0.905	0.338	0.146‡	0.080†	0.034†	**17.96**	17.16
	+RRM ($\alpha = 0.2, part$)	**0.836**†	**0.266**†	**0.096**†	**0.043**†	0.015†	18.00	17.17
Greedy	TransferTransfo	1.275	0.477	0.187	0.089	0.037	—	18.02
	+RRM ($\alpha = 0.3, full$)	**1.247**†	**0.454**†	**0.178**	**0.083**	0.034	—	**18.09**
	+RRM ($\alpha = 1, divide$)	1.255§	0.473	0.199‡	0.099‡	0.042§	—	17.87
	+RRM ($\alpha = 0.2, part$)	1.265	0.469	0.188	0.085	**0.033**	—	18.08
				Dialog-Level				
Beam	TransferTransfo	28.423	14.319	7.786	4.822	2.800	—	—
	+RRM ($\alpha = 0.3, full$)	**27.952**†	**13.982**†	**7.605**§	**4.743**	2.791	—	—
	+RRM ($\alpha = 1, divide$)	28.034†	14.275	7.894	4.955	2.931§	—	—
	+RRM ($\alpha = 0.2, part$)	27.956†	14.066‡	7.663	4.762	**2.773**	—	—
Beam	TransferTransfo w/o history	33.058	19.399	11.940	7.960	5.180	—	—
	+RRM ($\alpha = 0.05, full$)	**32.306**†	**18.650**†	**11.265**†	**7.330**†	**4.671**†	—	—
	+RRM ($\alpha = 1, divide$)	33.340§	19.814†	12.347†	8.331†	5.465†	—	—
	+RRM ($\alpha = 0.2, part$)	32.696‡	18.934†	11.559†	7.698‡	5.040	—	—
Greedy	TransferTransfo	32.960	17.208	8.852	5.022	2.805	—	—
	+RRM ($\alpha = 0.3, full$)	**32.559**†	**16.741**†	**8.532**†	4.852§	2.706	—	—
	+RRM ($\alpha = 1, divide$)	32.692§	17.115	8.872	5.110	2.867	—	—
	+RRM ($\alpha = 0.2, part$)	32.678‡	16.919‡	8.599‡	**4.822**‡	**2.689**	—	—

Table 11 Results of the extensive experiments on the PERSONA-CHAT test dataset. **Bold** indicates the least repeat, perplexity, and highest F1 scores for each setting. Because perplexity does not depend on the decoding method, we report it only once in the table. For each setting, we fixed α to the value that yielded the least repeat (sentence-level) under 1-gram on the validation dataset.

search, the RRM reduced the number of repetitions at both the sentence- and dialog-level (1-gram) and improved the F1 scores of the TransferTransfo model more than for greedy decoding. Compared to that excluding historical utterances, the RRM utilizing historical utterances during inference could improve F1 scores more but reduce fewer repetitions for the TransferTransfo model. In addition, when the input sequence excluded the history during the inference, the RRM trained with *part* setting in Eq. (18) is more effective in reducing sentence-level repetitions (1-gram) than when using *full* setting. This indicates that our third supposition is incorrect and the *part* setting is unstable. We believe that the cause for the unstable performance is that when using the *part* setting, $\tilde{q}$ and $\tilde{x}$ in Eq. (17) are generated from the full input sequence and only a part of the input sequence, which makes the information unbalanced. The *divide* setting performed the worst among the *full*, *divide*, and *part* settings. The results of BART and BART+RRM when greedy decoding is used on the PERSONA-CHAT test dataset are presented in Table 12. Using RRM for BART reduced repeats by 0.027 (sentence-level) (1-gram) and 0.09 (dialog-level) (1-gram) points. Compared with beam search, when using greedy decoding, the RRM reduced more sentence-level repetitions (1-gram) and fewer dialog-level repetitions (1-gram) for BART. The statistics for different repetitions of the PERSONA-CHAT test dataset are summarized in Table 13. Distinct from the two machine translation datasets, TransferTransfo and transformer started to generate phrases #3- and #4-level consecutive repetitions. This could be because the PERSONA-CHAT was an open-ended dataset with a high probability of the popular phrases, e.g., "I do not." When using greedy decoding, the generated phrases #3- and #4-level consecutive, and total repetitions would be approximately twice as long as when using beam search. This result indicates the effectiveness of beam search in mitigating the number of repetitions. Compared with beam search, both baselines, TransferTransfo and BART, with RRM can reduce phrases #3- and

Model	Repeat (Sentence-Level)					Repeat (Dialog-Level)					F1
	1-gram	2-gram	3-gram	4-gram	5-gram	1-gram	2-gram	3-gram	4-gram	5-gram	
Source	0.452	0.033	0.002	0.000	0.000	20.125	3.112	0.474	0.118	0.051	—
Reference	0.482	0.030	0.002	0.000	0.000	20.840	3.278	0.434	0.091	0.027	100.00
BART	1.170	0.380	0.136	0.074	0.037	33.780	17.194	8.750	5.187	3.028	**20.08**
+RRM	**1.143**†	**0.363**†	**0.126**‡	**0.069**§	**0.034**	**33.690**	**17.072**	**8.674**	**5.116**	**2.953**	20.03

Table 12 Experimental results for BART and BART+RRM on the PERSONA-CHAT test dataset with greedy decoding. α was fixed to 0.2, which yielded the least repeat (sentence-level) under 1-gram on the validation dataset. **Bold** indicates the least repeat and highest F1 scores for different levels.

Decode	Model	Word	Phrase #2	Phrase #3	Phrase #4	Total
—	Source	0.10	0.02	0.01	0.00	3.96
—	Reference	0.06	0.02	0.00	0.00	4.23
Beam	TransferTransfo	0.18	0.18	0.77	0.79	8.52
	+RRM ($\alpha = 0.3$)	0.12‡	0.15	0.56†	0.62‡	7.84†
	BART	**0.04**	0.14	**0.22**	0.35	7.31
	+RRM ($\alpha = 0.3$)	**0.04**	**0.13**	0.24	**0.31**	**7.22**
Greedy	TransferTransfo	0.12	0.19	1.74	1.60	13.84
	+RRM ($\alpha = 0.3$)	0.14	0.20	1.39†	1.27†	13.51†
	BART	**0.02**	0.18	0.71	0.70	12.31
	+RRM ($\alpha = 0.2$)	0.01§	**0.15**‡	**0.64**	**0.67**	**12.08**†

Table 13 Probability (%) of word-, phrase (number of words)-level consecutive, and total repetitions on the PERSONA-CHAT test dataset. **Bold** indicates the numbers closest to the reference scores for each setting.

#4-level consecutive repetitions when using greedy decoding. Meanwhile, the two baselines with RRM still achieved a closer probability of total repetitions to natural texts compared to those without RRM.

These results indicate that when the RRM is utilized, the method for combining multiple pieces of information for the input sequence is important. Furthermore, the decoding method affects the performance of the RRM. A sample dialog is presented in Table 2. Similar to the machine translation task, the TransferTransfo model generated repeated phrases, which was alleviated by the proposed model. In particular, "i 'm a real estate agent" in the second turn is a 5-gram nonconsecutive repetition of the one in the first turn at the dialog-level when the word "a" is ignored. "they are twins" in the third turn is a 3-gram consecutive repetition at both the sentence- and dialog-level because it is generated twice in a response.

5 Related Work

To overcome the problem of repetition in neural machine translation, Tu et al. (2016) and Mi et al. (2016) introduced the coverage mechanism into a seq2seq model; thus, the *decoder* can pay attention to the *encoder* information without duplication. See et al. (2017) extended the coverage model by incorporating a pointer-generator network based on Tu et al. (2016). However, it is difficult to utilize these coverage methods for multihead attention-based models

because multihead attention is a stack of several attention layers, and each layer is trained to capture its own distribution. Furthermore, Tu et al. (2016) and Mi et al. (2016) are based on one-to-one correspondence generation, which cannot be applied to "lossy" compression tasks such as summarization.

Suzuki and Nagata (2017) proposed WFE, which used several linear transformations to map the hidden states of the *encoder* to the upper-bound occurrence of each target vocabulary and controlled the generation using the estimated occurrence. However, we could not apply WFE for certain generation tasks, such as the response generation task, in which the frequency of the target tokens is irrelevant to the source sentence. Kiyono et al. (2018) proposed an SPM and assumed that the output sentences are shorter than the input sentences (i.e., a summary or a headline of the input). To ensure that the lengths of the input and output sentences were equal, special <pad> tokens were added at the end of the target sentence. Although this method helps SPM estimate over- or under-generation using the Euclidean distance, it limits the application of the SPM. Because our approach does not rely on the above assumptions, the RRM is more scalable for other downstream tasks, including machine translation and response generation.

6 Conclusion

In this study, we propose a novel mechanism to suppress repetitions in machine translation and response generation. Our model attempts to estimate the semantic vectors from a source sentence on both sides of an *encoder-decoder* model that considers semantic repetitions and does not rely on attention features. Therefore, our proposed method can be applied to other seq2seq models, which is an advantage over previous methods.

The experimental results of the IWSLT 2014 German-to-English, WMT 2014 English-to-German machine translation tasks, and PERSONA-CHAT response generation task demonstrated the effectiveness of our proposal. The results of extensive experiments on the response generation task demonstrated that the RRM can handle a concatenated input sequence.

Because our proposal considers semantic repetitions, we believe that it can reduce the number of repetitions among semantically similar words. This will be verified in a future study.

Acknowledgement

This paper is an extended and revised version of Zhang et al. (2021) accepted for publication at the 13th Conference on Recent Advances in Natural Language Processing (RANLP 2021).

This version has revised the grammar and introduced more details about the seq2seq model, autoencoder model, coverage mechanism, types of repetition, and our proposed RRM in the Abstract, Sections 1, 2, 3, and 4. This version also added experiments to evaluate the performances of models concerning diverse repetitions and on the WMT 2014 English-to-German dataset and added the vanilla transformer model as a baseline for comparison.

References

Bahdanau, D., Cho, K. H., and Bengio, Y. (2015). "Neural Machine Translation by Jointly Learning to Align and Translate." In *3rd International Conference on Learning Representations, ICLR 2015*.

Bengio, Y. et al. (2009). "Learning Deep Architectures for AI." *Foundations and Trends® in Machine Learning*, **2** (1), pp. 1–127.

Bojar, O., Buck, C., Federmann, C., Haddow, B., Koehn, P., Leveling, J., Monz, C., Pecina, P., Post, M., Saint-Amand, H., Soricut, R., Specia, L., and Tamchyna, A. (2014). "Findings of the 2014 Workshop on Statistical Machine Translation." In *Proceedings of the 9th Workshop on Statistical Machine Translation*, pp. 12–58, Baltimore, Maryland, USA. Association for Computational Linguistics.

Brier, G. W. (1950). "Verification of Forecasts Expressed in Terms of Probability." *Monthly Weather Review*, **78** (1), pp. 1–3.

Cettolo, M., Niehues, J., Stüker, S., Bentivogli, L., and Federico, M. (2014). "Report on the 11th IWSLT Evaluation Campaign, IWSLT 2014." In *Proceedings of the International Workshop on Spoken Language Translation, Hanoi, Vietnam*, p. 57.

Cho, K., van Merriënboer, B., Bahdanau, D., and Bengio, Y. (2014). "On the Properties of Neural Machine Translation: Encoder–Decoder Approaches." In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pp. 103–111, Doha, Qatar. Association for Computational Linguistics.

Denkowski, M. and Lavie, A. (2014). "Meteor Universal: Language Specific Translation Evaluation for Any Target Language." In *Proceedings of the 9th Workshop on Statistical Machine Translation*, pp. 376–380, Baltimore, Maryland, USA. Association for Computational Linguistics.

Fan, A., Lewis, M., and Dauphin, Y. (2018). "Hierarchical Neural Story Generation." In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 889–898, Melbourne, Australia. Association for Computational Lin-

guistics.

Fonollosa, J. A., Casas, N., and Costa-jussà, M. R. (2019). "Joint Source-Target Self Attention with Locality Constraints." *arXiv preprint arXiv:1905.06596*.

Fu, Z., Lam, W., So, A. M.-C., and Shi, B. (2021). "A Theoretical Analysis of the Repetition Problem in Text Generation." *Proceedings of the AAAI Conference on Artificial Intelligence*, **35** (14), pp. 12848–12856.

Gehring, J., Auli, M., Grangier, D., Yarats, D., and Dauphin, Y. N. (2017). "Convolutional Sequence to Sequence Learning." In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 1243–1252. JMLR.org.

Hochreiter, S. and Schmidhuber, J. (1997). "Long Short-Term Memory." *Neural Computation*, **9** (8), pp. 1735–1780.

Holtzman, A., Buys, J., Du, L., Forbes, M., and Choi, Y. (2020). "The Curious Case of Neural Text Degeneration." In *International Conference on Learning Representations*.

Kiyono, S., Takase, S., Suzuki, J., Okazaki, N., Inui, K., and Nagata, M. (2018). "Reducing Odd Generation from Neural Headline Generation." In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*, Hong Kong. Association for Computational Linguistics.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020). "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension." In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7871–7880, Online. Association for Computational Linguistics.

Li, P., Lam, W., Bing, L., and Wang, Z. (2017). "Deep Recurrent Generative Decoder for Abstractive Text Summarization." In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2091–2100, Copenhagen, Denmark. Association for Computational Linguistics.

Liu, T., Luo, F., Xia, Q., Ma, S., Chang, B., and Sui, Z. (2019). "Hierarchical Encoder with Auxiliary Supervision for Neural Table-to-Text Generation: Learning Better Representation for Tables." In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and 9th AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press.

Luo, L., Xu, J., Lin, J., Zeng, Q., and Sun, X. (2018). "An Auto-Encoder Matching Model for Learning Utterance-Level Semantic Dependency in Dialogue Generation." In *Proceedings of*

the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 702–707, Brussels, Belgium. Association for Computational Linguistics.

Luong, M.-T., Le, Q. V., Sutskever, I., Vinyals, O., and Kaiser, L. (2015a). "Multi-task Sequence to Sequence Learning." *arXiv preprint arXiv:1511.06114*.

Luong, T., Pham, H., and Manning, C. D. (2015b). "Effective Approaches to Attention-based Neural Machine Translation." In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Ma, S., Sun, X., Lin, J., and Wang, H. (2018). "Autoencoder as Assistant Supervisor: Improving Text Representation for Chinese Social Media Text Summarization." In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 725–731, Melbourne, Australia. Association for Computational Linguistics.

Mi, H., Sankaran, B., Wang, Z., and Ittycheriah, A. (2016). "Coverage Embedding Models for Neural Machine Translation." In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 955–960, Austin, Texas. Association for Computational Linguistics.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). "Distributed Representations of Words and Phrases and Their Compositionality." In *Advances in Neural Information Processing Systems*, pp. 3111–3119.

Miller, A., Feng, W., Batra, D., Bordes, A., Fisch, A., Lu, J., Parikh, D., and Weston, J. (2017). "ParlAI: A Dialog Research Software Platform." In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 79–84, Copenhagen, Denmark. Association for Computational Linguistics.

Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D., and Auli, M. (2019). "fairseq: A Fast, Extensible Toolkit for Sequence Modeling." In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pp. 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). "Bleu: a Method for Automatic Evaluation of Machine Translation." In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Pasunuru, R. and Bansal, M. (2018). "Game-Based Video-Context Dialogue." In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 125–136,

Brussels, Belgium. Association for Computational Linguistics.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). "Improving Language Understanding by Generative Pre-training.".

See, A., Liu, P. J., and Manning, C. D. (2017). "Get To The Point: Summarization with Pointer-Generator Networks." In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Sennrich, R., Haddow, B., and Birch, A. (2016). "Neural Machine Translation of Rare Words with Subword Units." In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). "Sequence to Sequence Learning with Neural Networks." In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K. (Eds.), *Advances in Neural Information Processing Systems*, Vol. 27. Curran Associates, Inc.

Suzuki, J. and Nagata, M. (2017). "Cutting-off Redundant Repeating Generations for Neural Abstractive Summarization." In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pp. 291–297, Valencia, Spain. Association for Computational Linguistics.

Tu, Z., Lu, Z., Liu, Y., Liu, X., and Li, H. (2016). "Modeling Coverage for Neural Machine Translation." In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 76–85, Berlin, Germany. Association for Computational Linguistics.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). "Attention is All you Need." In *Advances in Neural Information Processing Systems*, Vol. 30, pp. 5998–6008. Curran Associates, Inc.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. (2020). "Transformers: State-of-the-Art Natural Language Processing." In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online. Association for Computational Linguistics.

Wolf, T., Sanh, V., Chaumond, J., and Delangue, C. (2019). "TransferTransfo: A Transfer Learning Approach for Neural Network Based Conversational Agents." *CoRR*, **abs/1901.08149**.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao,

Q., Macherey, K., et al. (2016). "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation." *arXiv preprint arXiv:1609.08144*.

Xu, J., Liu, X., Yan, J., Cai, D., Li, H., and Li, J. (2022). "Learning to Break the Loop: Analyzing and Mitigating Repetitions for Neural Text Generation." *arXiv preprint arXiv:2206.02369*.

Zhang, S., Dinan, E., Urbanek, J., Szlam, A., Kiela, D., and Weston, J. (2018). "Personalizing Dialogue Agents: I Have a Dog, Do You Have Pets Too?" In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2204–2213, Melbourne, Australia. Association for Computational Linguistics.

Zhang, Y., Kamigaito, H., Aoki, T., Takamura, H., and Okumura, M. (2021). "Generic Mechanism for Reducing Repetitions in Encoder-Decoder Models." In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pp. 1606–1615, Held Online. INCOMA Ltd.

Ying Zhang: Ying Zhang is currently a Ph.D. candidate in the Department of Information and Communications Engineering, Tokyo Institute of Technology. Before that, she received a B.S. in Management from Chongqing University in 2017 and her M.E. from the Tokyo Institute of Technology in 2019. Her current research interests include natural language processing and machine learning.

Hidetaka Kamigaito: Hidetaka Kamigaito received his Ph.D. from the Tokyo Institute of Technology and is currently an associate professor at the Nara Institute of Science and Technology. His current research interests include natural language processing with an emphasis on document-level text processing and knowledge graph completion.

Tatsuya Aoki: Tatsuya Aoki is a machine-learning engineer at Apple Inc., where he works on natural language processing for Eastern Asian languages. He received an M.S. in Engineering from the Tokyo Institute of Technology and B.S. in Information and Media from the University of Tsukuba.

Hiroya Takamura: Hiroya Takamura received his Ph.D. from the Nara Institute of Science and Technology. He has worked as a professor at the Tokyo Institute of Technology and is currently a research team leader at the AI Research Center of the National Institute of Advanced Industrial Science and Technology. His current research interests include natural language processing.

Manabu Okumura: Manabu Okumura was born in 1962. He received B.E.,

M.E., and Dr. Eng. from Tokyo Institute of Technology in 1984, 1986, and 1989, respectively. He was an assistant in the Department of Computer Science, Tokyo Institute of Technology, from 1989 to 1992, and an associate professor at the School of Information Science, Japan Advanced Institute of Science and Technology, from 1992 to 2000. He is currently a professor at the Institute of Innovative Research at Tokyo Institute of Technology, Japan. His current research interests include natural language processing, particularly text summarization, computer-assisted language learning, sentiment analysis, and text data mining.

(Received July 27, 2022)
(Revised November 29, 2022)
(Accepted January 18, 2023)