Generic Mechanism for Reducing Repetitions in Encoder-Decoder Models

Ying Zhang[1], **Hidetaka Kamigaito**[1], **Tatsuya Aoki**[1],
Hiroya Takamura[2] **and Manabu Okumura**[1]
[1]Tokyo Institute of Technology
[2]National Institute of Advanced Industrial Science and Technology (AIST)
{zhang,kamigaito,aoki,oku}@lr.pi.titech.ac.jp
takamura.hiroya@aist.go.jp

Abstract

Encoder-decoder models have been commonly used for many tasks such as machine translation and response generation. As previous research reported, these models suffer from generating redundant repetition. In this research, we propose a new mechanism for encoder-decoder models that estimates the semantic difference of a source sentence before and after being fed into the encoder-decoder model to capture the consistency between two sides. This mechanism helps reduce repeatedly generated tokens for a variety of tasks. Evaluation results on publicly available machine translation and response generation datasets demonstrate the effectiveness of our proposal.

1 Introduction

Sequence-to-sequence (seq2seq) 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). As Mi et al. (2016) reported, however, basic seq2seq models (Bahdanau et al., 2015; Luong et al., 2015b) sometimes suffer from a repetition problem. One reason is that the attention mechanism does not explicitly consider which source side tokens have already been covered in the past attentions. As a result, the *encoder* repeatedly attends to the same token in the decoding steps, which leads to redundant generation.

Many researchers have proposed variants of the seq2seq model to tackle the problem. The coverage mechanism (Tu et al., 2016; Mi et al., 2016) prevented the model from generating redundant outputs by taking into account the coverage of the attention distribution. These approaches can be easily incorporated into the seq2seq model with only a single attention distribution between the *encoder* and the *decoder*. However, for seq2seq models with multiple attentions, such as Transformer (Vaswani et al., 2017), we cannot calculate the coverage of attentions, because the encoder attempts to attend to multiple attentions on each layer in the decoder. Thus, it is challenging to incorporate the coverage mechanism into the multi-attention based seq2seq models.

As another solution, Suzuki and Nagata (2017) proposed word-frequency estimation (WFE) that predicts the upper-bound frequency for each output token from the 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. While WFE and SPM have an advantage in not depending on the structure of a seq2seq model, it is difficult to apply these approaches to some tasks other than text summarization because WFE and SPM assume that the input sentence contains more tokens than the output.

To cope with the above problems, in this work, we propose a generic approach for reducing the repetition, focusing on the differences between the embedding spaces of the source and target sides. Based on the assumption of distributional semantics, our approach regards the representations of an input sentence on both sides as word vectors, and attempts to minimize their difference during the training step. Hence, the seq2seq model can explicitly take into account the source side context also in the *decoder*.

Our experimental results on the IWSLT 2014 German-to-English translation task (Cettolo et al., 2014) and the PERSONA-CHAT response generation task (Zhang et al., 2018) showed that the proposed method effectively alleviates the repetition

Proceedings of Recent Advances in Natural Language Processing, pages 1606–1615
Sep 1–3, 2021.
https://doi.org/10.26615/978-954-452-072-4_180

problem for both tasks.

2 Seq2seq Model

Given a source sentence $X = \{x_1, .., x_I\}$, the seq2seq model generates a target sentence $Y = \{y_1, ..., y_J\}$, 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 a source sentence X, and the *decoder* generates a target sentence by decomposing the conditional probability:

$$p(Y|X) = \prod_{j=1}^{J} p(y_j|y_{<j}, X). \quad (1)$$

3 Repetition Reduction Module

3.1 Overview

An overview of our proposed method, the repetition reduction module (RRM), is illustrated in Figure 1. We employ Transformer for both the *encoder* and *decoder* in the explanation in this section. Let $\tilde{x}$ be the source side sentence representations of source sentence X. With RRM, inspired by Kiyono et al. (2018) and Luong et al. (2015a), we consider $\tilde{x}$ as the correct representation of X and try to reconstruct $\tilde{x}$ in the target side. We use $\tilde{q}$ to represent the reconstructed $\tilde{x}$. Then the seq2seq model predicts not only the target side sequence Y but also $\tilde{x}$. The prediction is written as follows:

$$p(Y, \tilde{x}|X) = p(\tilde{x}|Y, X)p(Y|X). \quad (2)$$

The conditional probability $p(\tilde{x}|Y, X)$ has the role of preventing either over- or under-generation of Y by predicting the source side context until the decoding step ends. $p(\tilde{x}|Y, X)$ can be simplified as $p(\tilde{x}|X)$ if $\tilde{q}$ does not depend on Y. Since $p(Y|X)$ is predicted by the seq2seq model as shown in Eq. (1), we give details of $p(\tilde{x}|Y, X)$ in the next section.

3.2 Prediction of Source Side Context

Instead of using count-based discrete representations as in Kiyono et al. (2018), we incorporate continuous representations for both source and target sides to capture deeper semantic relations (Mikolov et al., 2013). We assume $p(\tilde{x}|Y, X)$ is proportional to the similarity between the representations of the source sentence X before and after being encoded and decoded:

$$p(\tilde{x}|Y, X) \propto \exp(\alpha(\cos(\tilde{x}, \tilde{q}))), \quad (3)$$

where α is a scaling factor.

Next, we explain the representations of the source sentence X in the source and target sides. Letting V_s be the source vocabulary, we define the indicator vector for the presence of source tokens as $x_i \in \{0, 1\}^{|V_s|}$. The source side representation $\tilde{x}$ of the source sentence is defined as follows:

$$\tilde{x} = \sum_{i}^{I} E_{src} x_i, \quad (4)$$

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

Similarly, we define the target side representation $\tilde{q}$ of the source sentence as follows:

$$\tilde{q} = \sum_{j}^{J} E_{src} q_j, \quad (5)$$

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

$$q_j = SoftMax(W_q \tilde{z}_j + b_q), \quad (6)$$

where $\tilde{z}_j$ is the final hidden state from the decoder, W_q is a weight matrix, and b_q is a bias term. Note that this softmax layer is only used in the training step.

3.3 Objective Function

By considering the negative log-likelihood of Eq. (2) , we can induce our objective function G_t as follows:

$$G_t = \sum_{(X,Y) \in D} \{-\log p(Y|X) - \alpha(\cos(\tilde{x}, \tilde{q}))\}, \quad (7)$$

where D is a parallel training corpus.

4 Experiments

4.1 Experimental Settings

We first used the IWSLT 2014 German-to-English translation task to evaluate our method. The dataset is split into 160k/7k/7k sentences for training, validation, and test. Since Cho et al. (2014) reported that seq2seq models tend to produce few unknown tokens and yield high BLEU scores for short sentences in the neural machine translation task, we supposed longer sentences are vulnerable to be

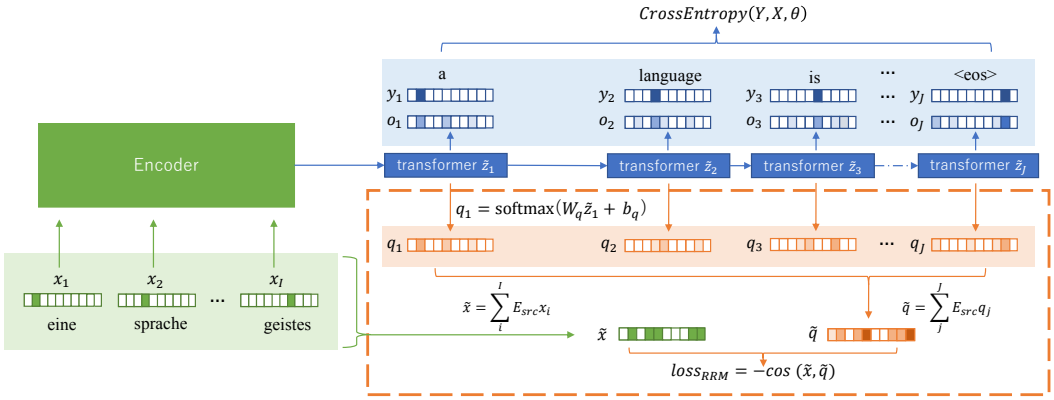

Figure 1: Overview of transformer-based encoder-decoder model with repetition reduction module (RRM). The part inside a dashed rectangular box represents RRM. o_j represents the probability distribution over the target vocabulary at the j-th decoding step.

over-translation, and our proposal would perform better for longer sentences. Therefore, we divided the test data into 3 parts: *Short*, *Medium*, and *Long*. In *Short* with 4927 pairs, the source contains no more than 25 byte pair encoding (BPE) (Sennrich et al., 2016) tokens. In *Medium* with 1524 pairs, the source contains 26 to 50 BPE tokens. In *Long* with 299 pairs, the source contains more than 50 BPE tokens.

We used PERSONA-CHAT for response generation as another dataset. This is the official dataset of The Conversational Intelligence Challenge 2 (ConvAI2)[1] for testing chatbots. It contains 164k/15k/15k utterances (corresponding to 10k/1k/1k dialogs) for training, validation, and test. It also contains corresponding persona information for each dialog.

We used the model of Fonollosa et al. (2019) as a baseline for the machine translation task. And we regarded the best performing model (Wolf et al., 2019) in ConvAI2 as our baseline for the response generation task. Wolf et al. (2019) adopted a Generative Pretrained Transformer (Radford et al.) based *encoder* and a 12-layer Transformer *decoder*, and concatenated the persona information, up to two turns of history utterances, and the query (the utterance) together as an input sequence. To investigate the effectiveness of our proposed module, we compared the experimental results between the models with and without RRM on top of the baseline models.

For 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 task. Repeat is defined as follows. Following the definition by Kiyono et al. (2018), we think that a model causes a repetition if it outputs the same token more than once. For each pair of a generated translation and its corresponding reference in the dataset, while we considered some tokens might occur more than once in the reference, 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.

For the response generation task, we used the official evaluation metrics, F1 and Perplexity. Note that the official method, offered by ParlAI (Miller et al., 2017), ignores words {a, an, the} and punctuation when computing F1. To compute Perplexity, Wolf et al. (2019) indirectly predicted the word probability on the basis of the ratio of probabilities of subwords since they utilized a BPE vocabulary.

Different from the machine translation task, the response generation task has no fixed answer. Therefore, in this task, we ignored the reference sequence when we computed Repeat. For each generated sequence, Repeat was computed by subtracting 1 from the frequency of tokens that occur more than once in the generated sequence. While ignoring the words {a, an, the} and punctuation, we calculated Repeat scores under an n-gram setting at sentence-level and dialog-level. At the sentence-level, we calculated Repeat only with each generated response. At the dialog-level, we calculated Repeat with the concatenation of a sequence of the generated responses in a dialog.

For the machine translation task, we followed the experimental settings of Fonollosa et al. (2019),

[1] http://convai.io/

Model	Repeat	BLEU	Meteor
Fonollosa et al. (2019)*	-	35.70	-
Fonollosa et al. (2019)	1.244	35.61	35.76
+RRM	**1.229**	**35.71**	**35.77**

Table 1: Experimental results on the IWSLT 2014 De-En test dataset. **Bold** indicates the best scores. Results were the average over 3 runs by random seeds. * indicates the reported scores by Fonollosa et al. (2019). For both models using cosine similarity and euclidean distance, α was fixed to 0.3, that yielded the best with Repeat on the validation dataset.

Data	Model	Repeat	BLEU	Meteor
Short	Fonollosa et al. (2019)	**0.552**	37.41	**36.83**
	+RRM	0.554	**37.47**	36.81
Medium	Fonollosa et al. (2019)	2.484	34.28	34.91
	+RRM	**2.467**	**34.38**	**35.00**
Long	Fonollosa et al. (2019)	6.371	33.11	**34.12**
	+RRM	**6.036**	**33.36**	33.99

Table 2: Experimental results on the IWSLT 2014 De-En test dataset at different lengths. **Bold** indicates the best scores. Results were the average over 3 runs by random seeds. For both cosine similarity and euclidean distance, α was fixed to 0.3, that yielded the best with Repeat on the validation dataset.

and tuned the scaling factor α in Eq. (7) with Repeat as the evaluation metric on the validation dataset. For the response generation task, we carried on the experimental settings of Wolf et al. (2019), and tuned the scaling factor α with Repeat (Sentence-Level) under 1-gram as the evaluation metric on the validation dataset. See Appendix A for a complete list of hyperparameter settings.

4.2 German-to-English Results

Table 1 shows the experimental results for the German-to-English translation task. The results suggest that combining RRM with the model of Fonollosa et al. (2019) helps to improve Repeat, BLEU, and Meteor scores.

Next, we compared the experimental results for *Short*, *Medium*, and *Long* to investigate the effectiveness of RRM at different source sentence lengths. Table 2 summarizes the experimental results. Similar to the results from Cho et al. (2014), both models tended to have lower BLEU scores and more repetitions for longer sentences. RRM performed relatively well for longer sentences. It reduced more repetitions and improved more BLEU on top of Fonollosa et al. (2019) for longer sentences. While RRM ($\alpha = 0.3$) showed no effect in reducing the repetition for short sentences, it assisted the model of Fonollosa et al. (2019) in reducing the Repeat score by 0.335 points and improved BLEU by 0.25 points for long sentences. These results suggest the importance of RRM for long sentences.

In Table 3, we list top and bottom 20 words based on the degree of Repeat reduction by RRM ($\alpha = 0.3$). These results show that RRM tended to reduce repetitions for high frequency words. RRM showed no effect of reducing repetitions for "." and "'s." See Appendix B for sample translations.

4.3 Response Generation Results

Tables 4 and 5 show the experimental results for the response generation task. Obviously, RRM reduced Repeat (Sentence-Level) by 0.056 (1-gram), and Repeat (Dialog-Level) by 0.471 (1-gram). The results suggest that combining RRM with the model of Wolf et al. (2019) helps to improve F1 and Repeat, while the performance of RRM in reducing perplexity is limited. We suppose that there are two reasons. One is that the probability calculation method offered by Wolf et al. (2019) is an indirect method. The other is that, the response is not fixed for a given source. See Appendix C for a sample dialog.

We conducted extensive experiments to investigate whether RRM has a potential to reduce more repetitions by considering the following conditions. First, beam search is an optimized decoding method which generates less repetitions than greedy decoding, and it may limit the performance of RRM. We investigate whether decoding methods might influence the performance of RRM by comparing beam search with greedy decoding.

Second, Wolf et al. (2019) used the persona information and the history utterances as an input sequence for their model to generate a response that differs from the history and contains a part of the persona information. However, in this task our model shared its vocabulary between source and target sides, and in Eq. (3) we utilized 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 investigate whether using history utterances in the input sequence during testing might influence the performance of RRM. "w/o

Word	Frequency	Frequency Rank	Sum of Repeat		Reduced Repeat
			Fonollosa et al. (2019)	+RRM ($\alpha = 0.3$)	
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 3: 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 Fonollosa et al. (2019) for the IWSLT 2014 De-En test dataset at *Long* length. Frequency is the frequency in the training dataset, and Frequency Rank is its rank in the training dataset.

Model	Repeat (Sentence-Level)					Perplexity	F1
	1-gram	2-gram	3-gram	4-gram	5-gram		
Wolf et al. (2019)*	-	-	-	-	-	16.28	19.50
Wolf et al. (2019)	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**

Table 4: Experimental results on the PERSONA-CHAT test dataset. **Bold** indicates the best scores. Results were the average over 3 runs by random seeds. * indicates the reported scores by Wolf et al. (2019). α was fixed to 0.3, that yielded the best with Repeat (Sentence-Level) under 1-gram on the validation dataset.

Model	Repeat (Dialog-Level)				
	1-gram	2-gram	3-gram	4-gram	5-gram
Wolf et al. (2019)	28.423	14.319	7.786	4.822	2.800
+RRM	**27.952**	**13.982**	**7.605**	**4.743**	**2.791**

Table 5: Experimental results on the PERSONA-CHAT test dataset. **Bold** indicates the best scores. Results were the average over 3 runs by random seeds. α was fixed to 0.3, that yielded the best with Repeat (Sentence-Level) under 1-gram on the validation dataset.

history" indicates the case where the history utterances were not used for an input sequence during testing.

Third, the history utterances might contain a part of the persona information, which can cause additional repetitions in Eq. (4). RRM might be misled

Decode	Model	Repeat (Sentence-Level) 1-gram	2-gram	3-gram	4-gram	5-gram	Perplexity	F1
Beam	Wolf et al. (2019)*	-	-	-	-	-	16.28	19.50
	Wolf et al. (2019)	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	Wolf et al. (2019) 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	Wolf et al. (2019)	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

Table 6: The results of the extensive experiments on the PERSONA-CHAT test dataset. **Bold** indicates the best scores for each setting. Results were the average over 3 runs by random seeds. * indicates the reported scores by Wolf et al. (2019). Because Perplexity does not depend on a decoding method, we report it only once in the table. For each setting, we fixed α to the value that yielded the best performance with Repeat (Sentence-Level) under 1-gram on the validation dataset.

Decode	Model	Repeat (Dialog-Level) 1-gram	2-gram	3-gram	4-gram	5-gram
Beam	Wolf et al. (2019)	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	Wolf et al. (2019) 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	Wolf et al. (2019)	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 7: The results of the extensive experiments on the PERSONA-CHAT test dataset. **Bold** indicates the best scores for each setting. Results were the average over 3 runs by random seeds. For each setting, we fixed α to the value that yielded the best performance with Repeat (Sentence-Level) under 1-gram on the validation dataset.

to producing more repetitions at the sentence-level and hence more repetitions at the dialog-level. We therefore investigated whether using the persona information and the history utterances in Eq. (4) during training influences the performance of RRM. *Full* indicates the usage of the persona information, the history utterances and the query as a source in Eq. (4) during training, while *part* indicates the usage of only the query. We also tried the setting *divide*, which divides the input sequence in Eq. (4) into three parts, $\tilde{x}_p, \tilde{x}_h, \tilde{x}_l$, depending on the persona information, the history utterances and the query, and uses the corresponding W_{q_p}, W_{q_h}, W_{q_l} in Eq. (6) to compute $\tilde{q}_p, \tilde{q}_h, \tilde{q}_l$ respectively. Then, the averaged cosine similarity was calculated between each divided $\tilde{x}$ and $\tilde{q}$.

Tables 6 and 7 show the results of our extensive experiments. Clearly, RRM reduced more repetitions and improved F1 scores more when using beam search and the $full$ setting. When the input sequence excluded history utterances during testing, RRM performed worse in F1 scores. Using only the query ($part$) in Eq. (4) during training reduced more repetitions at the sentence-level than using a full input sequence ($full$) when the input sequence excluded the history. But under other settings, RRM performed best when $full$ was used. $divide$ setting was the worst among the $full$, $divide$ and $part$ settings.

It indicates that our third supposition was wrong

and the $part$ setting was unstable. We think the reason of such unstable performance is, when using the $part$ setting, $\tilde{q}$ and $\tilde{x}$ in Eq. (3) were respectively generated from the full input sequence and only a part of the input sequence, which makes the information unbalanced.

The above results suggest that, when RRM was utilized, the method for combining multiple information for the input sequence was important. Furthermore, the decoding method would influence the performance of RRM.

5 Related Work

To overcome the repetition problem in neural machine translation, Tu et al. (2016) and Mi et al. (2016) introduced the coverage mechanism into a seq2seq model so that 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 hard to utilize these coverage methods for multi-head attention based models because multi-head attention is a stack of several attention layers, and each layer is trained to capture its own distribution. Furthermore, the works of Tu et al. (2016) and Mi et al. (2016) are based on one-to-one correspondence generation, which cannot be applied to a "lossy" compression task such as summarization.

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

6 Conclusion

In this work, we proposed 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, which takes semantic repetitions into consideration and does not rely on any attention features. Therefore, it is potential to apply our proposal to other sequence-to-sequence models, which is an advantage of our approach compared with previous methods.

Experimental results on the IWSLT 2014 German-to-English translation task and the PERSONA-CHAT response generation task demonstrated the effectiveness of our proposal. The results of the extensive experiments in the response generation task showed RRM has the ability to handle a concatenated input sequence.

Because our proposal takes the semantic repetitions into consideration, we believe it might have the ability to reduce the repetitions among semantically similar words. We will verify it as future work.

References

Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.

Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, and Marcello Federico. 2014. Report on the 11th iwslt evaluation campaign, iwslt 2014. In *Proceedings of the International Workshop on Spoken Language Translation, Hanoi, Vietnam*, page 57.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 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*, pages 103–111, Doha, Qatar. Association for Computational Linguistics.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 376–380, Baltimore, Maryland, USA. Association for Computational Linguistics.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for*

Computational Linguistics (Volume 1: Long Papers), pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

José AR Fonollosa, Noe Casas, and Marta R Costa-jussà. 2019. Joint source-target self attention with locality constraints. *arXiv preprint arXiv:1905.06596*.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR (Poster)*.

Shun Kiyono, Sho Takase, Jun Suzuki, Naoaki Okazaki, Kentaro Inui, and Masaaki Nagata. 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.

Piji Li, Wai Lam, Lidong Bing, and Zihao Wang. 2017. Deep recurrent generative decoder for abstractive text summarization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2091–2100, Copenhagen, Denmark. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101*.

Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2015a. Multi-task sequence to sequence learning. *arXiv preprint arXiv:1511.06114*.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015b. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Haitao Mi, Baskaran Sankaran, Zhiguo Wang, and Abe Ittycheriah. 2016. Coverage embedding models for neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 955–960, Austin, Texas. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Alexander Miller, Will Feng, Dhruv Batra, Antoine Bordes, Adam Fisch, Jiasen Lu, Devi Parikh, and Jason Weston. 2017. ParlAI: A dialog research software platform. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 79–84, Copenhagen, Denmark. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 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)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Ramakanth Pasunuru and Mohit Bansal. 2018. Game-based video-context dialogue. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 125–136, Brussels, Belgium. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training.

Abigail See, Peter J. Liu, and Christopher D. Manning. 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)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 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)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Jun Suzuki and Masaaki Nagata. 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*, pages 291–297, Valencia, Spain. Association for Computational Linguistics.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–85, Berlin, Germany. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Thomas Wolf, Victor Sanh, Julien Chaumond, and Clement Delangue. 2019. Transfertransfo: A transfer learning approach for neural network based conversational agents. *CoRR*, abs/1901.08149.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 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)*, pages 2204–2213, Melbourne, Australia. Association for Computational Linguistics.

A Hyperparmeters

For the machine translation task, we followed the experimental settings of Fonollosa et al. (2019), which used fairseq (Ott et al., 2019) and a 31K-token BPE source and target vocabulary. For the hyperparameters of Transformer, we used 14 layers, an embedding size of 256, a feedforward expansion size of 1024, and 4 attention heads. We used the Adam (Kingma and Ba, 2015) optimizer with a 4k mini-batch size and 85k training steps. The learning rate was linearly warmed from 1×10^{-7} to 0.001 in 4k steps and then decayed by a weight of 0.0001 (Loshchilov and Hutter, 2017). In the decoding steps, we used beam search (Wu et al., 2016) with a beam size of 5. We set the scaling factor α to $\{1, 0.3, 0.2, 0.05, 0.01\}$ and selected the best α with Repeat as the evaluation metric on the validation dataset. We pretrained the model of Fonollosa et al. (2019) in advance to extract word embeddings.

For the response generation task, we carried on the experimental settings of Wolf et al. (2019). We utilized a 40k-token BPE source and target vocabulary and trained the model with 2 epochs, a batch size of 32 sequences, and the Adam optimizer. The learning rate was linearly decayed from 6.25×10^{-5} to zero. In the decoding step, we adopted beam search, and top 20 sampling (Fan et al., 2018) was utilized before selecting four beams. We set the scaling factor α to $\{1, 0.3, 0.2, 0.05, 0.01\}$. Since RRM was designed to reduce repetitions at the sentence-level, we selected the best α with Repeat (Sentence-Level) under 1-gram as the evaluation metric on the validation dataset.

B Sample of German-to-English

Table 8 shows sample translations. While the model of Fonollosa et al. (2019) tended to generate repeating phrases, our model reduced such generation.

C Sample of Response Generation

We show a sample dialog in Table 9. Similar to the machine translation task, the model of Wolf et al. (2019) tended to generate repeating phrases, and our model helped to alleviate it. In particular, "i'm a real estate agent" in the second turn is a 5-gram repetition of the one in the first turn at the dialog-level when word "a" is ignored. "they are twins" in the third turn is a 3-gram repetition at both dialog-level and sentence-level because it is generated twice in a response.

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 .
Fonollosa et al. (2019)	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 8: Sample translations for *Short*, *Medium*, and *Long* data. Underline indicates repetitions that contain more than two words, and **bold** indicates wrong translations. α was fixed to 0.3, that yielded the best with Repeat on the validation dataset.

Response generation		
	Persona Information	i have two children that are in their kindergarten. i am 34 years old. 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
	Wolf et al. (2019)	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
	Wolf et al. (2019)	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
	Wolf et al. (2019)	yes, **they are twins**, they are twins
	+RRM	they're, they're in their early 20s.

Table 9: 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 history utterances, and it receives the concatenation of the persona information, the history utterances and the third query as an input. α was fixed to 0.3, that yielded the best with Repeat (Sentence-Level) under 1-gram on the validation dataset.