Data Augmentation for Robust Natural Language Generation Based on Phrase Alignment and Sentence Structure

Kenta Yamamoto, Seiya Kawano, Tatsuya Kawahara, Koichiro Yoshino

Abstract Recent natural language generation (NLG) systems, which take meaning representations (MRs) as input and generate corresponding texts (utterances), can present fluent sentences with neural networks behind them; however, the system often suffers from hallucination, a problem in generating information that is not part of the given MR. This study focuses on suppressing the hallucination problem using augmented data. We propose a data augmentation method that creates variations of the training data based on phrase alignment and sentence structure. The proposed method extracts correspondence between slots in the MR and terms in the sentence by phrase alignment and syntactic structure of the sentence. It uses them to prepare new MRs and their corresponding sentence as augmented data. Experimental results showed that the system trained by our augmented data realized a robust NLG system with high naturalness and informativeness even though it can suppress the hallucination.

1 Introduction

The demand for dialogue systems has been fueled by the spread of speech applications. Meaning representation (MR) is widely used to express both a user's and a system's intent as a machine-interpretable expression in task-oriented systems as restaurant searches [28] or tourist information navigation [27]. MRs are defined with several slots in which each one has a corresponding value, such as an entity. A natural language generation (NLG) task converts MRs into utterance [25].Data-driven methods for NLG have been studied using datasets including the E2E NLG Challenge [3].

Kenta Yamamoto
Osaka University e-mail: kentayamamoto@sanken.osaka-u.ac.jp

Seiya Kawano and Koichiro Yoshino
RIKEN e-mail: {seiya.kawano, koichiro.yoshino}@riken.jp

Tatsuya Kawahara
Kyoto University e-mail: kawahara@i.kyoto-u.ac.jp

A variety of models have been proposed: rule-based models [3], template-based models [14, 19] and neural networks (NNs) based models [7, 1, 4, 8].

NLG methods based on supervised learning often cause a hallucination problem. Hallucination is a phenomenon in which the generated sentences contain content that is not specified in the given condition [3, 10]. This is particularly problematic in NLG tasks for dialogue systems, where content that does not exist in the input MR is mentioned in the utterance. Hallucination undermines trust in dialogue systems because they cite aspects unintended by the system designers/users. It can also cause inconsistencies or contradictions in a dialogue history, since such systems generate utterances that do not correspond to the dialogue management results.

When attempting to deal with hallucinations, most existing research is known to devise inputs to the model, such as Retrieval Augmented Generation (RAG) [18], or to filter the outputs [29]. On the other hand, it is also possible to deal with such hallucinations by devising training data. Generation models based on NNs can produce very fluent utterances for combinations of MRs in the training data; however, they often produce incorrect content for combinations of MRs that do not exist in the training data because they highly depend on training data. In other words, if the training data adequately covers the patterns needed for generation, more appropriate generation is possible.

This study addresses this problem with a simple idea; adding unseen MR patterns to the training data by data augmentation. We make the augment training data (Data-R) from the original training data (Data-O) by deleting some slots in existing MRs and also editing the corresponding sentence considering content correspondences and sentence structure. In this study, we proposed using an alignment tool and an attention weight to acquire correspondences. In our method, we obtain the part of the sentence that corresponds to the removed slot values and edit the sentence based on this information to create augmentation data. This editing process uses syntactic information to eliminate any syntactic/semantic errors to achieve qualified augmentation data. Our experimental results show that the proposed data augmentation method suppressed the hallucination of test data that contained unknown MR patterns and improve the informativeness of the generated sentences.

2 Preliminaries

In this study, we focus on NLG tasks that uses MR as input, such as those used in dialogue systems, and solve the hallucination problem in NLG tasks. This section defines our task and explains our approach to resolving hallucination problems in existing studies.

Inform	Values
MR (input)	name [the Wrestlers], priceRange [cheap], customerRating [low]
Generated sentence (output)	The wrestlers offers competitive prices, but isn't highly rated by customers.

Table 1 Example of MRs and generated sentences: restaurant guide

Slot name	Example of values	Slot name	Example of values
name	*Eagle*, ...	eatType	*restaurant*, *pub*, ...
familyFriendly	*Yes* / *No*	priceRange	*cheap*, *expensive*, ...
food	*French*, *Italian*, ...	near	*Zizzi*, *Cafe Adriatic*, ...
area	*riverside*, *city center*, ...	customerRating	*1 of 5 (low)*, ...

Table 2 Example of slots in MRs: restaurant guide

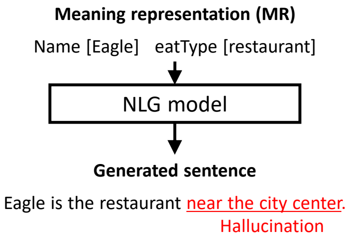
Meaning representation (MR)
Name [Eagle]
eatType [restaurant]
NLG model
Generated sentence
Eagle is the restaurant near the city center.
Hallucination

Fig. 1 An example of hallucination in NLG task

2.1 *Natural Language Generation*

The NLG task generates utterances corresponding to the given MR. Formally, the NLG model takes multiple slot values as an MR as input and outputs sentences that reflect all the contents of the slot values. Examples of MRs and generated sentences for a restaurant guide task are shown in Table 1. In this task's MR, there are multiple slots related to restaurants as defined in Table 2 in accordance with E2E NLG challenge [3]. These slots are given with specific values, and the goal is to generate a sentence that introduces the restaurant based on the input (Table 1). Such data are generally prepared manually to correspond to the input MR and to build a model to generate sentences from such data.

2.2 *Hallucination*

NN-based models have been widely used for generating word sequences. However, most language generation models based on NNs suffer from the hallucination problem, as shown in Figure 1. Hallucination is a phenomenon in which the generated sentence includes content that does not appear in the MR. This problem complicates an information transfer because the model generates sentences that contain more information than was initially intended in the MR.

One of the causes of hallucination in NLG tasks is overfitting the model to the dataset. NN-based language generation models are highly dependent on training data. In other words, they can generate fluent sentences for the combination patterns of MRs that are included in the training data. They can also generate unnatural sentences for combinations that are absent from the training data, especially for combination patterns where some of the MRs are missing from Data-O. However, creating a dataset for every slot combination is difficult because training datasets are generally created manually.

Several methods have been proposed to cope with this gap between MRs and generation results, including decoding algorithms [5] and data augmentation using external data [17, 24]. The data augmentation approach compensates for the distribution of Data-O with Data-R for the actual variation of the given MR [7, 8]. Our study adopts this policy but applies a simpler idea: deleting some slots and their corresponding terms from the existing training data. When creating such data, we have to delete the expressions in the output sentences that correspond to the deleted slots. We expect our proposal to suppress hallucination because we can prepare pseudo-training data for possible slot combinations that do not exist in Data-O, even though it does not require any additional data.

3 Data Augmentation Based on Alignment and Sentence Structure

We describe our proposed method for data augmentation in this section. We used paired data of MRs and generated sentences as training data. First, some slots in the MR in a pair of Data-O are randomly deleted. Then we prepare a sentence that corresponds to the MR with deleted slots. The sentences in Data-O are edited based on their correspondence with the slots in MR and the sentence structure to prepare such corresponding sentences. Figure 2 overviews the proposed method. “Original dataset (Data-O)” indicates a pair of MRs and corresponding output sentences. The data are converted to the “Augmented dataset (Data-R)” in the last part of the figure. Data augmentation consists of two steps. “Alignment” indicates the relationships between the slot values of the MR and the terms of the output sentences. From the alignment results, “filtering based on syntax” uses the sentence structure to eliminate unnatural sentence editing. The following sections describe these steps in detail.

3.1 Alignment of Meaning Representation and Output Sentence

We obtained the correspondence between the slot values of the MR and the terms in the output sentences by alignment methods. We prepared two alignment methods: an alignment tool for machine translation [11] and the attention weights of the Transformer-based language model [23].

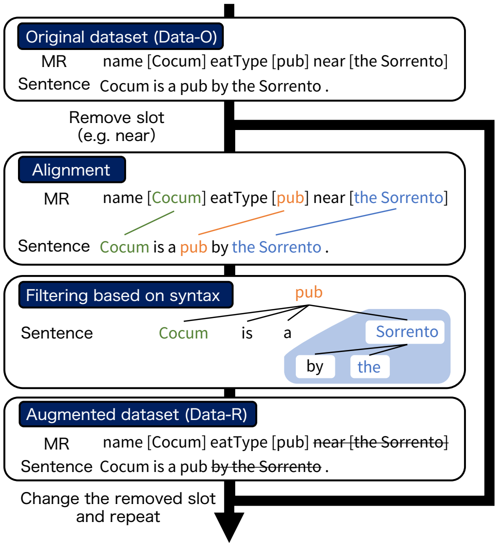
Original dataset (Data-O)
MR name [Cocum] eatType [pub] near [the Sorrento]
Sentence Cocum is a pub by the Sorrento .
Remove slot
(e.g. near)
Alignment
MR name [Cocum] eatType [pub] near [the Sorrento]
Sentence Cocum is a pub by the Sorrento .
Filtering based on syntax
pub
Sentence Cocum is a Sorrento
by the
Augmented dataset (Data-R)
MR name [Cocum] eatType [pub] ~~near [the Sorrento]~~
Sentence Cocum is a pub ~~by the Sorrento~~ .
Change the removed slot
and repeat

Fig. 2 Data augmentation procedure: Words with no correspondence by alignment are corrected by filtering based on the syntax tree.

3.1.1 Alignment Tools (GIZA++)

Alignment tools, which estimate phrase correspondence between parallel sentences, are widely used in statistical machine translation [12]. We use Giza++[1] with IBM-model-4 to output the alignment results. To input MRs to the tool, we alternately input both slot names and their values in the MR, such as "slot name 1 [slot value 1] slot name 2 [slot value 2] ..."

As indicated in Figure 2, the alignment tool provides the corresponding words in the output sentence for each slot. We can acquire the words that should be deleted from the sentence in Data-R. In the example in Figure 2, when the slot "near [the Sorrento]" is deleted, "the Sorrento" is estimated as the deletion target based on the alignment result. However, this process alone results in an unnatural sentence because "by" remains in the edited sentence. Therefore, we use a correction with a syntax tree (described in Section 3.2) to avoid unnatural sentences.

3.1.2 Attention Weights

We also propose using attention weights in the attention mechanism of the Transformer model as correspondence alignment. Here we use a Transformer consisting of a two-layer encoder and a two-layer decoder, as shown in Figure 3. The cross-attention layer of the second decoder layer maps the slots in the MR to words in the

[1] https://github.com/moses-smt/giza-pp

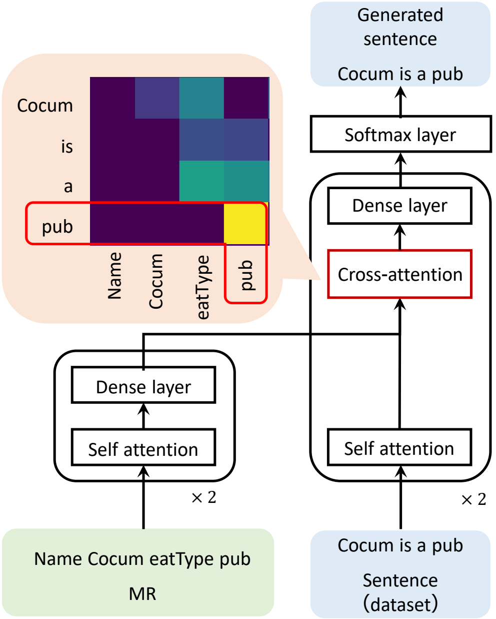

Fig. 3 Alignment using attention weight

output sentence. A language generation model based on this Transformer architecture is trained using the paired data of MRs and output sentences. When a target sentence is input, cross-attention weights are extracted from the trained model. The decoder is trained to output the same sentence. The average value $\bar{x}$ of all the elements of the matrix is set as the threshold value (see the confusion matrix in Figure 3), and element x_{ij} of the matrix that exceeds the threshold value $\bar{x}$ is obtained. For element x_{ij}, the i-th word (slot names or values) in the MR corresponds to the j-th word in the output sentence.

3.2 *Filtering Based on Syntax*

The alignment results are used to edit the sentences. However, as described in Section 3.1.1, the alignment results alone are insufficient for proper sentence editing, and as a result, unnatural sentences are often generated. Therefore, we filter the unnatural sentences based on their sentence structure. We expect the filtering to remove words in the generated sentences corresponding to the deleted slots that are not captured by the alignment alone. Although the alignment results may contain omissions in the correspondence, phrases that contain the words to be deleted can be appropriately removed based on the sentence structure. Moreover, by considering syntax trees, we can suppress the generation of sentences whose dependency relations are unknown. Thus, parsing information is also used to determine which slots to remove. We can determine the removed slots to keep the minimum elements of the edited sentences, i.e., a predicate and its obligatory cases are retained in Data-R.

We used Stanza [15], a Python wrapper for a Stanford Core NLP, as our syntax parser, which extracts the case information of words with their dependency relations. First, slots corresponding to predicates and obligatory cases are excluded from the deletion target, and then the target slots are randomly determined. The words corresponding to the slots to be deleted are marked, and a subtree is identified that covers all of these words. All of these subtrees are deleted as if they were the subtrees corresponding to the deleted slots. If a word to be deleted contains a word corresponding to any non-deleted slots, the sample is not added to Data-R. Pairs of MRs and sentences with some of the slots removed are used as a part of Data-R. We determine the number of slots in the MR to delete and list the combinations of them. This procedure is repeated for all the combinations of slots that can be deleted. This allows the data to be augmented without compromising the syntactic naturalness of the sentences in Data-R.

4 Experimental Settings

We experimentally confirmed the robustness of the NLG models trained by Data-R. We used the E2E Challenge dataset [3] in our testbed. In addition to the test set of the E2E NLG Challenge, we also used comprehensive test data from it, edited it to remove some slots and added new correct sentences to them to check the effect on the hallucination problem. In addition to the automatic evaluations, the naturalness, the informativeness, and the amount of hallucination were evaluated manually. The details of the experiments are described below.

4.1 E2E Challenge Dataset

E2E Challenge dataset contains slots of MRs and corresponding sentences for a restaurant information task. The structure of the data is shown in Table 3. In this dataset, the number of generated sentences exceeds the number of MRs because identical MRs are manually generated by multiple annotators.

4.2 Baseline Model and De-lexicalization

In this study, we use the SLUG model [7], which achieved the highest score on the E2E NLG challenge leader-board with an simple architecture, as the language generation model. In SLUG, the training data are de-lexicalized during training. De-

Data type	Number	Variation of slots
Training	42,061	4,862
Validation	4,672	547
Test	4,693	630
Total	51,426	6,039

Table 3 Composition of training data on baseline model

lexicalization replaces the corresponding words in the slot values and the generated sentences with placeholder tokens [5]. This reduces the vocabulary and the amount of required train data. Placeholder tokens in the generated sentences are replaced with the original words in a post-processing step. The three slots are de-lexicalized: name, near, and food. Words that appear in these slots can be replaced by rule-based replacement because the same words appear in the generated sentences. On the other hand, in such slots as "pricerange" and "area", there are cases where the slot value "less than $20" is expressed as "cheap" in the generated text or where "riverside" is replaced with "by the river." Thus, they are not de-lexicalized.

If we simply apply de-lexicalization, the words around the de-lexicalized slot affect vocabulary choices. We define placeholder tokens by focusing on the following two points. The first is whether the nouns in the slot values are singular or plural, including the usage of "a" or "an" for singular nouns. The second point is whether the value of the "food" slot contains a surface word "food". For example, the expression "serves placeholder food" is valid in the "food[french]" slot, although it is undesirable in the "food[fast food]" slot. They are distinguished by the "cuisine" list.

Both the baseline and the proposed method use such de-lexicalized data. In some cases, de-lexicalized place-holders are generated that are not given in the input MR. These patterns are removed during decoding. This is one way to suppress hallucination during decoding.

4.3 Test Data for hallucination Evaluation

As evaluation data, we use the test data contained in the E2E challenge dataset (Test-O). However, our focus is on hallucination when an MR is given that is not covered by Data-O. To evaluate whether systems can generate sentences with sufficient and necessary information, we prepared a new evaluation dataset to evaluate hallucination using 630 types of MRs in the test data. We randomly deleted slots in the MRs from the original test data and prepared new evaluation data for these partially missing MRs (Test-R). When deciding which MRs to delete, we used the method described in Section 3.2 to avoid removing predicates and obligatory cases. For each of the 630 pairs of slot values of MRs created in this way, we prepared a new corresponding correct answer by an annotator. We showed triplet of original sentences, original MRs, and MRs with the deleted slots to add appropriate sentences that corresponded to the MRs with the deleted slots.

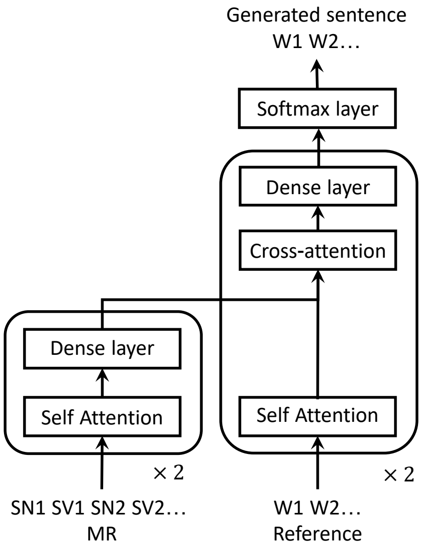

Fig. 4 Model architecture: slot name (SN), slot value (SV), word of the generated sentence (W)

4.4 Model setting

We have changed the encoder of the baseline model (SULG) [7] from LSTM, which was used in the original paper, to a Transformer. Both the encoder and decoder consist of two layers (Figure 4). The batch size for training is 1024, and the latent variables are 256-dimensional. For each slot in the MR, pairs of slot names and slot values are input in word units. A slot name placeholder is treated as a single word and identified as a slot name, e.g., "name_slot." The decoder outputs the raw sentences word by word. In the experiment, we applied the following hallucination suppression methods in both the baseline and the proposed model. During training, we applied teacher forcing to the decoder. During inference, the decoder used top-K decoding and used its 1-best. However, if the 1-best directly outputs slot name ("eattype", "pricerange", "costomerrating") or the de-lexicalized slot value is not included in MR, the generated sentence is excluded, and the next candidate sentence in top-K decoding is used as the output. We evaluated NLG models trained on Data-O and Data-R.

4.5 Evaluation Metrics

Automatic Evaluation We applied automatic evaluation scores to verify the effectiveness of our proposed data augmentation method. For this evaluation, we used the evaluation data of the original E2E challenge (Test-O) and the newly prepared evaluation data to focus on hallucination (Test-R). We used BLEU and Entity-F1.

BLEU is a lexical overlap metric used to measure similarity between generated sentences and references. Entity-F1 evaluates whether the corresponding words of the slots are output. The precision is calculated from the number of generated phrases that appear in the MR N_{tp} and do not appear in the MR N_{fp} using the formula $N_{tp}/(N_{tp}+N_{fp})$. The recall is calculated from N_{tp} and the number of phrases not in a generated sentence but in the MR N_{fn} using the formula $N_{tp}/(N_{tp}+N_{fn})$. Entity-F1 is the harmonic mean of them. If the precision is low and recall is high, hallucination probably occurred. Entity-F1 requires the expected output word for each slot in the MR to be calculated. For the de-lexical slots; "name," "food," and "near," it counts the slot value output in the generated sentence. A list of possible expressions is prepared for each slot for other slot values. The expressions included in the list are judged to ascertain whether they appear in the generated sentences.

Human Subjective Evaluation In the human evaluation, an annotator assessed the quality of the sentences generated by the model by their naturalness, informativeness and the number of hallucinations. The data for the evaluation consisted of 200 MRs randomly selected from the test data (test-R), and the annotators evaluated the generation results corresponding to each model. The evaluation was performed by one annotator other than the author of the paper. They were presented with the generated sentences and MRs and rated the following items on a 5-point scale.

- Naturalness: whether the generated sentence is natural;
- Informativeness: whether the generated sentence reflects the given MR.

Moreover, the annotator counted the number of hallucination to calculate the ratio of slot values described in the generated sentence even though they are not included in the MR. The human evaluation was done blindly, including the reference sentence in the test set to guarantee our evaluation quality.

4.6 Compared Models

In the evaluation, the case without data augmentation was used as the baseline model (**Baseline**). In contrast, we compared a case with data augmentation using an alignment tool (**Tool**), using an attention weights (**Attention**), and both methods (**Both methods**). For these data augmentations, we set "removed slots" based on the maximum number of slots to be removed. **Removed slots=1,2,3** allows a maximum of three slots to be randomly removed during data augmentation, increasing the amount of data augmentation. In **Both methods**, we use both augmented data by tool and attention.

Another method to suppress the hallucination is the using N-best during decoding. This method generates N-best candidates and selects the one with the highest Entity-F1 score for a given MR. We conducted an automatic evaluation for both with and without N-best decoding. In the human evaluation, we evaluated the outputs without 5-best decoding to measure the proposed method's pure effect in our experiments.

			Augment test set (Test-R)				E2E test set (Test-O)			
Model	Removed slots	Training data	BLEU	Entity			BLEU	Entity		
				Precision	Recall	F1		Precision	Recall	F1
Baseline	0	42,061	0.632	0.953	0.928	0.940	0.694	1.0	0.900	0.947
Tool	1	63,055	0.634	0.953	0.913	0.932	0.689	0.999	0.903	0.950
Tool	1,2	84,184	0.629	0.952	0.916	0.933	0.687	0.998	0.899	0.946
Tool	1,2,3	97,137	0.612	0.993	0.907	**0.949**	0.681	1.0	0.892	0.942
Attention	1	49,239	0.623	0.922	0.921	0.922	0.687	0.999	0.903	0.949
Attention	1,2	52,055	0.635	0.944	0.921	0.932	0.690	1.0	0.905	0.950
Attention	1,2,3	52,721	**0.637**	0.941	0.918	0.929	0.686	1.0	0.899	0.947
Both methods	1	70,233	0.635	0.930	0.904	0.917	0.687	0.999	0.900	0.947
Both methods	1,2,3	107,797	0.632	0.952	0.919	0.935	0.690	1.0	0.906	0.950

Table 4 Automatic evaluation results

			Augment test set (Test-R)				E2E test set (Test-O)			
Model	Removed slots	Training data	BLEU	Entity			BLEU	Entity		
				Precision	Recall	F1		Precision	Recall	F1
Baseline	0	42,061	0.615	0.964	0.934	0.949	0.672	1.0	0.908	0.951
Tool	1	63,055	0.516	0.969	0.931	0.950	0.650	0.999	0.911	0.953
Tool	1,2	84,184	0.567	0.968	0.930	0.948	0.656	0.999	0.909	0.952
Tool	1,2,3	97,137	0.587	0.994	0.928	**0.960**	0.659	0.999	0.908	0.952
Attention	1	49,239	0.609	0.963	0.926	0.944	0.665	1.0	0.911	0.953
Attention	1,2	52,055	0.617	0.970	0.931	0.950	0.653	1.0	0.916	0.956
Attention	1,2,3	52,721	**0.623**	0.981	0.930	0.955	0.670	1.0	0.909	0.952
Both methods	1	70,233	0.530	0.963	0.933	0.948	0.645	0.999	0.913	0.954
Both methods	1,2,3	107,797	0.602	0.987	0.931	0.958	0.643	0.999	0.907	0.951

Table 5 Automatic evaluation results (with N-best decoding)

5 Experimental Results

5.1 Automatic Evaluation Results

The experiment's results are shown in Tables 4 and 5, which respectively use 1-best and N-best decoding. First, we compared the evaluation results for the original E2E challenge (Test-O) and the newly prepared evaluation data (Test-R)[2]. In the Test-R results, the precision scores decreased, but the recall scores increased, indicating some hallucinations in Test-R. When we look at the Entity-F1 scores, using the alignment tool achieved the best F1 scores in both settings, even though their BLEU scores decreased. In contrast, the methods based on attention weights slightly improved the BLEU scores. Moreover, the number of augmented data (the "training data" in the table) from the attention weights was less than it from the alignment tool. This is because using the attention weights created more fluent augmented data. If we combine both methods, their scores are comparable to the baseline. This suggests that the results may be intermediate between the two methods. However, we have to

[2] We will open the test data when we publish the paper.

Model	Human evaluation		Hallucination rate
	Naturalness	Informativeness	(average hallucination per sample)
Baseline	**4.945***	4.050	12.6% **(0.305)**
Tool	4.500	3.900	15.3% **(0.455)**
Attention	4.900	**4.245****	**7.0%**** **(0.185)**
Both methods	4.350	3.930	21.3% **(0.595)**
Reference (gold)	4.880	4.855	0.6% **(0.020)**

($* < 0.05$, $** < 0.01$)

Table 6 Human evaluation for quality of generated sentences and number of hallucination. Wilcoxon signed-rank test between results of **Baseline** and **Attention** is used for analysis of naturalness and informativeness. A paired two-sided t-test between results of **Baseline** and **Attention** is used for the analysis of hallucination rate.

look at the human evaluation results because the correlation between the automatic evaluation metrics and human scores is not high.

If we compare the N-best decoding with 1-best decoding, N-best improved the Entity-F1 scores, and even their BLEU scores are decreased as general trends. This is because the N-best results decoding are optimized to the Entity-F1 scores. Using the alignment tool achieved the best Entity-F1 scores in both decodings, and using attention weights achieved the best BLEU scores in both decodings.

5.2 Human Subjective Evaluation Results

We focused on three human evaluation scores: naturalness, informativeness, and hallucination. The former two criteria are widely used in NLG research, and the last is a new criterion on which to focus our research issue. The results are shown in Table 6. First, our evaluator added high naturalness and informativeness to the reference sentences with less hallucination; this result qualifies our evaluation results.

According to the results, using attention weights shows the lowest hallucination rate and highest informative score without a big drop from the highest naturalness score. The informativeness results were significantly different from the scores of the baseline method at a significance level of 1%; even though it decreased the naturalness slightly.

Hallucination was also suppressed by a method other than baseline except by combining both methods. This indicates that the proposed data augmentation methods are adequate for the hallucination problem. However, surprisingly, using both methods did not contribute to the hallucination problem. This was caused by a gap between the distributions of the Data-R and the potential test dataset. **Both methods**, using the alignment tool and the attention mechanism, did not stabilize to learn because the generated sentence by each method has different characteristics.

In the manual evaluation, the method using attention was highly rated. This is probably because the method using the alignment tool did not consider the naturalness of the augmented sentences but just assumed syntactically consistent sentences if we compare it with the system based on attention weights. This result is consistent with the fact that the BLEU score was higher for the method using the attention weights than for the method using the alignment tool in the automatic evaluations. In the next section, we explain various methods proposed to suppress hallucination: decoding methods, design of the objective function during training, and data augmentation.

6 Related Work

Several decoding methods have been proposed to suppress hallucination [21, 17]. Tian et al., [21] used confidence scores calculated from attention weights to suppress hallucination. Shen et al., [17] estimated a slot in the given MR that corresponded to the segmentation of interest and used the slot value for the generation to suppress the hallucination. Using an NLG network that is robust for the unseen patterns is also proposed [22].

Several objective functions for training NLG models have been proposed to suppress hallucination [9, 13, 16]. They introduced reinforcement learning in the training process to improve the controllability of NLG models by penalizing a generated sentence that does not adequately represent a given MR. Multi-task learning is also applied to ensure the robust NLG [30].

Several data augmentation methods have been proposed to suppress hallucination [7, 8, 20, 26]. Juraska et al., [7] augmented the training data when a reference sentence consists of multiple sentences by splitting them into separate training samples with corresponding MRs. Their idea is similar to our approach, which improves the diversity of MR patterns in the training data, thereby suppressing hallucination. Other works [8, 2, 26] utilized sampled sentences from a pre-trained NLG model by using various MR patterns for the data augmentation. The data augmentation idea is also important in NLU tasks, the reversing task of NLG [6].

Our data augmentation method is common to conventional methods in that it improves the diversity of MR patterns in the training data. However, the performance of conventional methods based on sampling from pre-trained NLG models strongly depends on the original training data distribution. Thus, it suffers from the generation corresponding to rare MR patterns and the semantic unnaturalness of the generated sentences used for data augmentation. In contrast, the new training samples obtained by our data augmentation method cover a wide variety of MR patterns, even though they are natural and based on human-written sentences. Furthermore, our data augmentation method is more flexible than conventional data augmentation methods based on sentence/MR splitting, as they allow more fine-grained editing of the original sentences. It is also important to note that our method can be easily used with other data augmentation methods.

7 Conclusion

In this paper, we focused on suppressing the hallucination problem in NLG tasks. We proposed a data augmentation method using phrase alignment and filtering using sentence structure to solve this problem. We used an alignment tool and the attention weights of the NLG model to obtain correspondences and used the sentence structure to eliminate unnatural augmented sentences. Our proposed method based on the attention weights achieved the best BLEU score in the experiment and the highest human evaluation scores with hallucination suppression. We also confirmed that our data augmentation method could be used with other hallucination suppression methods, such as N-best decoding. Future work will validate our proposed method on a wider variety of data sets. The proposed method is simple and can be easily used in combination with other hallucination suppression methods, and the effectiveness of such combinations must also be investigated.

References

1. Agarwal, S., Dymetman, M., Gaussier, É.: Char2char generation with reranking for the E2E NLG challenge. In: Proceedings of the 11th International Conference on Natural Language Generation (INLG), pp. 451–456 (2018)
2. Du, W., Chen, H., Ji, Y.: Self-augmented data selection for few-shot dialogue generation. arXiv preprint arXiv:2205.09661 (2022)
3. Dušek, O., Novikova, J., Rieser, V.: Evaluating the state-of-the-art of end-to-end natural language generation: The e2e nlg challenge. Computer Speech and Language **59**(C), 123–156 (2020)
4. Elder, H., Gehrmann, S., O'Connor, A., Liu, Q.: E2E NLG challenge submission: Towards controllable generation of diverse natural language. In: Proceedings of the 11th International Conference on Natural Language Generation (INLG), pp. 457–462 (2018)
5. Elder, H., O'Connor, A., Foster, J.: How to make neural natural language generation as reliable as templates in task-oriented dialogue. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 2877–2888 (2020)
6. Glass, M., Rossiello, G., Chowdhury, M.F.M., Gliozzo, A.: Robust retrieval augmented generation for zero-shot slot filling. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 1939–1949 (2021)
7. Juraska, J., Karagiannis, P., Bowden, K., Walker, M.: A Deep Ensemble Model with Slot Alignment for Sequence-to-Sequence Natural Language Generation. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), pp. 152–162 (2018)
8. Kedzie, C., McKeown, K.: A good sample is hard to find: Noise injection sampling and self-training for neural language generation models. In: INLG (2019)
9. Li, Y., Yao, K., Qin, L., Che, W., Li, X., Liu, T.: Slot-consistent NLG for task-oriented dialogue systems with iterative rectification network. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 97–106 (2020)
10. Müller, M., Sennrich, R.: Understanding the properties of minimum bayes risk decoding in neural machine translation. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 259–272 (2021)
11. Och, F.J., Ney, H.: A systematic comparison of various statistical alignment models. Computational Linguistics **29**(1), 19–51 (2003)

12. Och, F.J., Ney, H.: The alignment template approach to statistical machine translation. Computational linguistics **30**(4), 417–449 (2004)
13. Perez-Beltrachini, L., Lapata, M.: Bootstrapping generators from noisy data. In: 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1516–1527. Association for Computational Linguistics (2018)
14. Puzikov, Y., Gurevych, I.: E2E NLG challenge: Neural models vs. templates. In: Proceedings of the 11th International Conference on Natural Language Generation, pp. 463–471 (2018)
15. Qi, P., Zhang, Y., Zhang, Y., Bolton, J., Manning, C.D.: Stanza: A Python natural language processing toolkit for many human languages. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations (2020)
16. Rebuffel, C., Soulier, L., Scoutheeten, G., Gallinari, P.: Parenting via model-agnostic reinforcement learning to correct pathological behaviors in data-to-text generation. In: Proceedings of the 13th International Conference on Natural Language Generation, pp. 120–130 (2020)
17. Shen, X., Chang, E., Su, H., Niu, C., Klakow, D.: Neural data-to-text generation via jointly learning the segmentation and correspondence. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 7155–7165 (2020)
18. Shuster, K., Poff, S., Chen, M., Kiela, D., Weston, J.: Retrieval augmentation reduces hallucination in conversation. In: Findings of the Association for Computational Linguistics: EMNLP 2021, pp. 3784–3803 (2021)
19. Smiley, C., Davoodi, E., Song, D., Schilder, F.: The E2E NLG challenge: A tale of two systems. In: Proceedings of the 11th International Conference on Natural Language Generation(INLG), pp. 472–477 (2018)
20. Su, S.Y., Huang, C.W., Chen, Y.N.: Dual supervised learning for natural language understanding and generation. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 5472–5477 (2019)
21. Tian, R., Narayan, S., Sellam, T., Parikh, A.P.: Sticking to the facts: Confident decoding for faithful data-to-text generation. CoRR **abs/1910.08684** (2019). URL http://arxiv.org/abs/1910.08684
22. Tseng, B.H., Budzianowski, P., Wu, Y.C., Gasic, M.: Tree-structured semantic encoder with knowledge sharing for domain adaptation in natural language generation. In: Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue, pp. 155–164 (2019)
23. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (eds.) Advances in Neural Information Processing Systems, vol. 30 (2017)
24. Wang, Z., Wang, X., An, B., Yu, D., Chen, C.: Towards faithful neural table-to-text generation with content-matching constraints. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 1072–1086 (2020)
25. Wen, T.H., Gasic, M., Mrkšić, N., Su, P.H., Vandyke, D., Young, S.: Semantically conditioned lstm-based natural language generation for spoken dialogue systems. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1711–1721 (2015)
26. Xu, X., Wang, G., Kim, Y.B., Lee, S.: Augnlg: Few-shot natural language generation using self-trained data augmentation. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 1183–1195 (2021)
27. Yoshino, K., Suzuki, Y., Nakamura, S.: Information navigation system with discovering user interests. In: Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, pp. 356–359 (2017)
28. Young, S., Gašić, M., Keizer, S., Mairesse, F., Schatzmann, J., Thomson, B., Yu, K.: The hidden information state model: A practical framework for pomdp-based spoken dialogue management. Computer Speech & Language **24**(2), 150–174 (2010)
29. Zhao, Z., Cohen, S.B., Webber, B.: Reducing quantity hallucinations in abstractive summarization. In: Findings of the Association for Computational Linguistics: EMNLP 2020, pp. 2237–2249 (2020)

30. Zhu, C., Zeng, M., Huang, X.: Multi-task learning for natural language generation in task-oriented dialogue. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 1261–1266 (2019)