IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 12 October 2023, accepted 12 November 2023, date of publication 16 November 2023,
date of current version 22 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3334139

RESEARCH ARTICLE

Content Order-Controllable MR-to-Text

KEISUKE TOYAMA, (Graduate Student Member, IEEE), KATSUHITO SUDOH, AND SATOSHI NAKAMURA, (Life Fellow, IEEE)
Nara Institute of Science and Technology, Ikoma, Nara 630-0192, Japan
Corresponding author: Keisuke Toyama (toyama.keisuke.tb5@is.naist.jp)

Part of this work was supported by JSPS KAKENHI Grant Number JP21H05054.

ABSTRACT Content order is critical in natural language generation (NLG) for emphasizing the focus of a generated text passage. In this paper, we propose a novel MR (meaning representation)-to-text method that controls the order of the MR values in a generated text passage based on the given order constraints. We use an MR-text dataset with additional value order annotations to train our order-controllable MR-to-text model. We also use it to train a text-to-MR model to check whether the generated text passage correctly reflects the original MR. Furthermore, we augment the dataset with synthetic MR-text pairs to mitigate the discrepancy in the number of non-empty attributes between the training and test conditions and use it to train another order-controllable MR-to-text model. Our proposed methods demonstrate better NLG performance than the baseline methods without order constraints in automatic and subjective evaluations. In particular, the augmented dataset effectively reduces the number of deletion, insertion, and substitution errors in the generated text passages.

INDEX TERMS Controllable text generation, data augmentation, data-to-text, meaning representation, natural language generation.

I. INTRODUCTION

DATA-to-text tasks generate a text passage from such (semi-) structured data as concepts, tables, graphs, etc. Such input data usually include multiple elements that will be mentioned in the output passage (Table 1), meaning that we can control the text generation's structure, length, and word order by *content planning*. This work focuses on the order of the contents, i.e., the data values, since the order is crucial for data-to-text that emphasizes important words or phrases. Here we explain two examples of the importance of controlling the content order.

A. PRESERVING THE ORDER TO AVOID CHANGING THE FOCUS OF A TEXT PASSAGE

Suppose we replace the value "*Italian*" of the `food` with "*French*" in Table 1. In this case, the ideal text passage becomes "*Wildwood is a restaurant that serves French food located near Raja Indian Cuisine in the area of riverside. Unfortunately, it is not kid friendly*." In this passage, the word "*Italian*" is changed to "*French*," although the content order is unchanged. In the reference text passage and the ideal text passage, the "*no*" of the `familyFriendly` attribute is emphasized by placing an independent sentence at the end of the text passage: "*Unfortunately, it is not kid friendly*." However, if we cannot control the order of a data-to-text system, it may generate a text passage with an unintended order: "*Wildwood is a restaurant that is not family friendly. It serves French food located near Raja Indian Cuisine in the riverside*." This text passage does not emphasize the "*no*" of

TABLE 1. Example of E2E dataset.

	Attribute	Value
MR	`name`	Wildwood
	`eatType`	restaurant
	`food`	Italian
	`priceRange`	(empty)
	`customer rating`	(empty)
	`area`	riverside
	`familyFriendly`	no
	`near`	Raja Indian Cuisine
Text	Wildwood is a restaurant that serves Italian food located near Raja Indian Cuisine in the area of riverside. Unfortunately, it is not kid friendly.	

The associate editor coordinating the review of this manuscript and approving it for publication was Gustavo Olague.

the `familyFriendly` attribute because of the location of the phrase "*that is not family friendly*" in the middle of the sentence.

B. CONTROLLING THE ORDER TO CHANGE THE FOCUS OF A TEXT PASSAGE

Suppose we emphasize the value "*riverside*" of the `area` attribute and "*Raja Indian Cuisine*" of the `near` attribute in Table 1. In this case, these values should appear at the beginning of the passage: "*In the riverside area near Raja Indian Cuisine, you can visit a restaurant called Wildwood where Italian food is served. It is not family-friendly.*"

Concerning such a content order problem in data-to-text, Kasner and Dušek [1] proposed a method that generates a text passage from resource description framework (RDF) triples in a zero-shot setting. RDF triples are converted using a template to a set of sentences, which are ordered to maximize their coherency by sentence ordering. However, this method automatically determines the sentence order without directly controlling the order using explicit constraints. Leng et al. [2] proposed a data-to-text method that controls sentence splits, the entity order, and the text length. They controlled the entity order by aligning the input data with the reference text. However, their method's performance in such standard automatic measures as BLEU [3], ROUGE_L [4], and METEOR [5] was lower than their baselines. Su et al. [6] proposed a data-to-text method that first predicts the suitable order of attributes from the data input and then generates output sentences from both the data input and the predicted order. Their method explicitly outputs ordering information as a content plan so that the order can be edited arbitrarily. Their human evaluation showed that the generated sentences accurately reflected the ordering information if the order was unedited. However, when the order was randomly shuffled, the accuracy was worse than without shuffling. Luo et al. [7] proposed a few-shot table-to-text generation method using a pre-trained language generation model. To generate sentences, the prompt "summarize the following table:" followed by the order of attributes that appear in the generated sentence and the input data are given to a pre-trained generation model. However, their human evaluation showed that the word order's accuracy was 0.84, which is insufficient.

In this work, we propose an MR-to-text (MR2T) method that controls the order of the MR values in generated text passages with additional value order annotations using an MR-text dataset called the E2E refined dataset [8]. First, we train an order-constrained MR2T model with a text-to-MR (T2MR) model. Then we augment the MR-text dataset to obtain a better-balanced distribution in terms of the number of non-empty attributes by synthesizing the attribute-value pairs that do not appear in the dataset. We investigate the performance of an MR2T model trained using the augmented dataset by automatic and subjective evaluations and show that it can control the order of the MR values with high accuracy. Our code is available at https://github.com/KSKTYM/content_order-controllable_mr-to-text.

TABLE 2. Example of E2E refined dataset.

	Attribute	Value	Order
MR	`name`	NAME (WILDWOOD)	1
	`eatType`	restaurant	2
	`food`	Italian	3
	`priceRange`	(empty)	0
	`customer rating`	(empty)	0
	`area`	riverside	5
	`familyFriendly`	no	6
	`near`	NEAR (RANA INDIAN CUISINE)	4
Text	NAME(WILDWOOD) is a restaurant that serves Italian food located near NEAR(RAJA INDIAN CUISINE) in the area of riverside. Unfortunately, it is not kid friendly.		

II. DATASET

The E2E dataset [9] used in the E2E NLG Challenge [10] is widely utilized in MR2T studies. It consists of a set of pairs of a British English text passage and a corresponding MR with the following eight attributes in a restaurant recommendation domain: `name`, `eatType`, `food`, `priceRange`, `customer rating`, `area`, `familyFriendly`, and `near`. However, some of its MR-text pairs suffer from the following errors:

- *deletion*: some attribute-value pairs are not mentioned in the text;
- *insertion*: some empty attributes are mentioned incorrectly with unintended values;
- *substitution*: some MR values are replaced by wrong ones.

Such errors must be fixed to properly control a text passage's content by MR2T. Although updates have rectified some of these errors [11], [12], the updated datasets still include such errors. We refined the E2E dataset with additional error fixes by manual annotations of the MR values and made it public as the *E2E refined dataset*[1] [8]. It also includes the order of the MR values in the corresponding text passage (Table 2). Here all of the `name` and `near` values are delexicalized, because they appear as-is in the text passages. We used the E2E refined dataset in this work to train the MR2T models that control the order of the MR values.

III. DATA AUGMENTATION

The number of training instances with non-empty attributes is shown in the second column of Table 3. For example, the number of training instances with eight non-empty attributes is only 1,190; that with five non-empty attributes is 12,442. Such an imbalanced training data distribution causes poor performance for instances with many non-empty attributes, as shown later in the experimental results.

[1]https://github.com/KSKTYM/E2E-refined-dataset (distributed under the CC 4.0-BY-SA license)

TABLE 3. Number of training data in terms of non-empty attributes: NEA denotes non-empty attributes.

# NEA	Original data	Augmented data	Merged data
1	55	0	55
2	393	7	400
3	2,910	926	3,836
4	8,119	4,323	12,442
5	12,442	0	12,442
6	10,058	2,384	12,442
7	5,393	7,049	12,442
8	1,190	11,252	12,442
Total	40,560	25,941	66,501

TABLE 4. All variations of MR values in E2E refined dataset.

Attribute	# variations	MR values (delexicalized)
`name`	1	NAME
`eatType`	4	(empty), coffee shop, pub, restaurant
`food`	11	(empty), American, Canadian, Chinese, English, fast food, French, Indian, Italian, Japanese, Thai
`priceRange`	7	(empty), £20-25, cheap, expensive, less than £20, moderate, more than £30
`customer rating`	7	(empty), 1 out of 5, 3 out of 5, 5 out of 5, average, high, low
`area`	3	(empty), city centre, riverside
`familyFriendly`	3	(empty), no, yes
`near`	2	(empty), NEAR

TABLE 5. Number of every possible combination of MR values and MR orders and obtained samples: NEA denotes non-empty attributes.

# NEA	MR value	MR order	MR value & order	Obtained samples
1	1	1	1	0
2	30	2	60	11
3	355	6	2,130	1,167
4	2,142	24	51,408	30,949
5	7,096	120	851,520	112,139
6	12,912	720	9,296,640	206,353
7	11,952	5,040	60,238,080	191,212
8	4,320	40,320	174,182,400	69,120
Total	38,808	-	244,622,239	610,955

Data augmentation is a promising approach for mitigating such data imbalance. Existing studies on data augmentation for NLG use text generation and text analysis models. Kedzie et al. [13] used noise injection sampling. First, they synthesized the under-represented MR in the training data. Second, they converted the MR to a text passage using their MR2T model by injecting Gaussian noise into the decoder hidden states. Then they obtained an MR using their MR parser. Finally, a pair of the obtained MR and the generated text passage was accepted as augmented data. Unfortunately, noise injection sampling made insertion and deletion errors in the generated text passage.

Chai et al. [14] proposed a feedback-aware self-training method for their conditional text generation. First, they generated a text passage whose condition is different from the original. Then a classifier predicted the condition from the generated text passage. A condition-passage pair was used as augmented data if the input and predicted conditions matched.

We applied an idea that resembles Chai's approach to augment the training data with synthetic examples generated in the following steps.

STEP 1

Generate every possible combination of MR values and orders. The number of MR value patterns is calculated from the variation of the MR values (Table 4). The number of MR order patterns is a factorial of their non-empty attributes. Since the number of generated combinations is enormous (244.6 million), we randomly sampled them to 16 patterns for an MR order, removed the MRs included in the original dataset to avoid data leakage, and obtained 610,955 MR combinations (Table 5).

STEP 2

Convert the MRs obtained in Step 1 to text passages by the MR2T model trained using the original training data. The model's details are explained in Section IV-D.

STEP 3

Convert the text passages obtained in Step 2 to MRs using the T2MR model trained using the original training data. The T2MR model is also explained in Section IV-D.

STEP 4

Augment the training data with the pairs of an MR and a text passage as *synthetic MR-text pairs* when the result of Step 3 matches the MR generated in Step 1. Here our motivation is to balance the data distribution. We sampled the pairs for a total of 12,442 (the maximum number in the original training data with five non-empty attributes) for each non-empty attribute (Table 3). Finally, we obtained 66,501 MR-text pairs as the augmented training data.

IV. METHOD

Next we explain the MR2T methods used in this work: TGEN [15], JUG [16], Transformer-based MR2T, and the T2MR methods based on Transformer.

A. TGEN

As our baseline, we use TGEN[2] [15], based on an LSTM-based sequence-to-sequence model with an attention mechanism. TGEN is also used as the baseline for the E2E NLG Challenge [10]. Note that TGEN does not constrain the order of the MR values in its text generation.

[2]https://github.com/UFAL-DSG/tgen (distributed under the Apache License 2.0)

B. JUG

As another baseline, we use JUG[3] [16], based on an LSTM-based generative model for joint natural language understanding (NLU) and generation, which couples NLU and NLG through a shared latent variable. According to [16], the BLEU score for the original E2E dataset was better than that of TGEN. This method does not constrain the order of the MR values in its text generation either.

C. TRANSFORMER WITHOUT ORDER CONSTRAINTS

We use Transformer [17] for MR2T, configured as shown in Fig. 1. The Transformer-based MR2T model takes a sequence of MR values as its input tokens and generates output tokens one by one. For example, it takes [“<*sos*>”, “*NAME*”, “*restaurant*”, “*Italian*”, “*riverside*”, “*no*”, “*NEAR*”, and “<*eos*>”] as the input tokens when the MR shown in Table 2 is used. Here each MR value is treated as one token without further tokenization into subwords. “<*sos*>” and “<*eos*>” are special symbol tokens that express a sequence’s start and its end. The MR-value tokens are given in a fixed order from `name` to `near`, and empty values are excluded from the input. Note that since all the MR values are unique (Table 4), the input tokens do not need to include MR attributes. For the positional values, we use $[0, 1, \ldots, n+1]$ (where n equals the number of non-empty attributes) as the input vector. These values are embedded with a trainable embedding.

For T2MR, we use Transformer with the same structure as that for MR2T. It takes a sequence of text tokens as input and predicts the MR values one by one in the fixed order of the attributes. We use the `word_tokenizer` module in the Python NLTK library for the text tokenization. For example, it takes [“<*sos*>”, “*NAME*”, “*is*”, “*a*”, “*restaurant*”, …, “*kid*”, “*friendly*”, “.”, and “<*eos*>”] as input to induce the corresponding MR value sequence as output: [“<*sos*>”, “*NAME*”, “*restaurant*”, “*Italian*”, “*riverside*”, “*no*”, “*NEAR*”, and “<*eos*>”].

For the inferences, MR2T runs greedily to generate a text passage, and T2MR predicts its MR values. If the T2MR result matches the original MR, the generated text passage is deemed reliable. If the MRs are not identical, MR2T generates text passages using beam search (width = 5) and applies T2MR to check whether the result is reliable and chooses the reliable one with the best score.

D. TRANSFORMER WITH ORDER CONSTRAINTS

We propose another Transformer-based MR2T model that takes the content order constraints. For example, it takes [“<*sos*>”, “*NAME*”, “*restaurant*”, “*Italian*”, “*NEAR*”, “*riverside*”, “*no*”, and “<*eos*>”] as input, where MR-value tokens appear in the corresponding order with their mentions in the text passage. The order-constrained T2MR model also has identical Transformer architecture, although it is trained to predict the MR values in the order of their mentions in the input text. We train these models with the original and augmented training datasets.

[3]https://github.com/andy194673/Joint-NLU-NLG

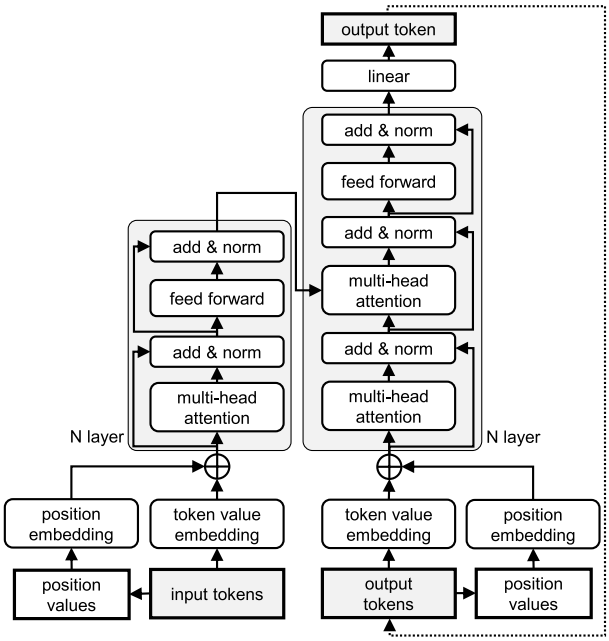

FIGURE 1. Transformer model.

V. EXPERIMENTS

We investigated the effect of the order constraints in MR2T by the following experiments.

A. DATA

We used the E2E refined dataset with 40,560 MR-text pairs for training and 4,555 pairs for evaluation. We also used the augmented training data (66,501 MR-text pairs) to train the Transformer with an order constraints model. To investigate the MR2T performance with different MR orders, we augmented the test data by reordering the MR values and used them as inputs. We sampled four different orders for the test data with three or more non-empty attributes and the alternative MR value order for those with two non-empty attributes and obtained 21,140 additional test instances. We then used four random seeds for the test data augmentation and obtained four sets of augmented test data (21,140 × 4 sets). Hereafter, we call them the *reordered test data*.

B. MODEL CONFIGURATION

We set the embedding vector size to 256, the feed-forward network vector size to 512, the head number to 8, and the layer number to 3. For training, we used the following settings: a batch size of 128, a learning rate of 0.0005, 100 epochs, a dropout rate of 0.1, and a clip norm of 1.0. The model was optimized using Adam [18]. We used one NVIDIA GeForce RTX 3090 GPU. It took about 33 minutes to train the MR2T model with the original training data and 53 minutes to train

TABLE 6. Accuracy of T2MR models. Bold indicates best result.

Method	MR value & order	MR value	MR order
Transformer w/o order	n/a	98.18	n/a
Transformer w/ order	98.62	98.95	98.84
Transformer w/ order (augmented training data)	**98.79**	**99.25**	**99.06**

it with the augmented training data. We chose the best model that resulted in minimum loss on the validation data in the E2E refined dataset among those 100 trained models at the end of the training epochs. The loss was saturated in ten epochs by the MR2T model without order constraints, nine epochs by that with the order constraints using the original training data, and ten epochs by that using the augmented training data. For TGEN and JUG, we used the distributed programs "*as-is*".

C. METRICS

We evaluated the MR2T performance by BLEU [3], NIST [19], METEOR [5], ROUGE_L [4], and CIDEr [20], all of which can be obtained using the E2E challenge metrics script.[4] To calculate these metrics, we prepared the following two types of references for the test instances:

- *Order-independent references*: those corresponding to the given MR without order constraints (the average number of reference text passages: 6.67);
- *Order-dependent references*: those corresponding to the given MR with order constraints (the average number of reference text passages: 1.21).

We also evaluated the performance by checking whether the original MR to MR2T and the predicted MR from T2MR are identical. We named this method *MRcheck*. Since the T2MR performance is highly accurate, as shown in Table 6, we believe it can be used to evaluate MR2T. We used the order-constrained T2MR model trained with the augmented training data for MRcheck.

D. RESULTS

Table 7 shows the scores. The Transformer MR2T model without order constraints outperformed TGEN and JUG, indicating Transformer's effectiveness in this task. Our proposed order-constrained MR2T model clearly outperformed the model without order constraints. The proposed model trained using the augmented data also outperformed the model trained with the original data. Although an advantage was observed even with the order-independent references, it was much larger with the order-dependent references.

Table 8 shows the MRcheck results. The order-constrained Transformer preserved the MR order very accurately (99.58%) for the original test data, although the baselines failed to do so (6.74%, 5.93%, and 7.42%). Even though the accuracy of the reordered test data (88.07%) was worse than that for the original test data (99.58%), the performance of the proposed model trained using the augmented training data was almost perfect (99.95%).

Furthermore, we found more MRcheck errors in the test instances with a larger number of non-empty attributes (Table 9): for example, 12 and 1,055.8 errors for the instances with eight non-empty attributes for the original and reordered test data. However, the amount of proposed model training using the augmented data significantly improved from 1,055.8 to 5.5. The numbers of deletion/insertion/substitution errors were also reduced, from 583.5 to 2.8 for deletion errors in the MR values, from 1.8 to 0.0 for insertion errors in the MR values, from 109.8 to 2.5 for substitution errors in the MR values, and from 1,842.5 to 4.8 for substitution errors in the MR order. These results suggest the effectiveness of our data augmentation method.

VI. HUMAN EVALUATION

We also conducted a human evaluation on the MR2T results. Native English-speaking crowdworkers[5] rated the naturalness, the adequacy [21], and the focus of the generated text packages of 150 selected examples from the original test data. The 150 examples were randomly selected after conditioning the attribute distribution that appeared first in the text package to be approximately uniform, as shown in Table 10. Three workers evaluated each text passage. To avoid any misunderstanding that the values of attributes `name` and `near` are emphasized because they are shown in uppercase, we capitalized only the first letter of each word of those attributes. We gave the instructions shown in Figs. 2, 3, and 4 and showed a pair of the MR and the text passages from each of the 150 examples (Figs. 5, 6, and 7) to the evaluators who evaluated them.

a: NATURALNESS

The evaluators gave scores on a 6-point Likert scale (higher is better) on the following four questions:

1) Is the sentence *natural*?
2) Is the sentence *grammatical*?
3) Is the sentence *comprehensible*?
4) Is the sentence *acceptable* as English, even if it is not natural/grammatical/comprehensible?

b: ADEQUACY

The evaluators answered the following two questions with either "*yes*" or "*no*":

1) Does the generated sentence meet all the MR values?
2) Does the generated sentence meet all the MR values and the MR orders?

[4]https://github.com/tuetschek/e2e-metrics

[5]We utilized *Prolific* (https://www.prolific.co/) and paid each worker thirteen pounds per hour.

TABLE 7. Results of automatic evaluation: **Bold** indicates best result for order-independent reference, and ***Italic*** indicates best result for order-dependent reference.

Method	Reference	BLEU(↑)	NIST(↑)	METEOR(↑)	ROUGE_L(↑)	CIDEr(↑)
TGEN	order-independent	0.5626	7.8907	0.4278	0.6614	2.4066
	order-dependent	0.3339	5.6815	0.3762	0.5262	2.0885
JUG	order-independent	0.5733	7.6896	0.4337	0.6488	2.3972
	order-dependent	0.3505	5.6242	0.3871	0.5201	2.0580
Transformer w/o order	order-independent	0.5840	7.8227	0.4384	0.6659	2.5141
	order-dependent	0.3600	5.7151	0.3910	0.5344	2.1753
Transformer w/ order	order-independent	0.6280	8.6083	0.4595	0.7551	2.7783
	order-dependent	0.4836	*7.0947*	0.4356	0.7422	3.3062
Transformer w/ order (augmented training data)	order-independent	**0.6335**	**8.6198**	**0.4624**	**0.7575**	**2.7855**
	order-dependent	*0.4914*	7.0941	*0.4393*	*0.7453*	*3.3383*

TABLE 8. MRcheck results of accuracy: **Bold** indicates best result for original test data, and ***Italic*** indicates best result for reordered test data).

Method	Test data	MR value & order	MR value	MR order
TGEN	original	6.74	99.41	6.74
JUG	original	5.93	98.24	5.93
Transformer w/o order	original	7.42	100.0	7.42
Transformer w/ order	original	99.58	99.96	99.58
	reordered	88.03	96.74	88.07
Transformer w/ order (augmented training data)	original	**100.0**	**100.0**	**100.0**
	reordered	*99.95*	*99.98*	*99.95*

TABLE 9. MRcheck errors (vd: deletion errors in MR values, vi: insertion error in MR values, vs: substitution error in MR values, os: substitution error in MR order). NEA denotes non-empty attributes.

# NEA	Original test data			Reordered test data		
	# data	# errors		# data	# errors	
		Transformer w/ order	Transformer w/ order (augmented training data)		Transformer w/ order	Transformer w/ order (augmented training data)
1	0	0	0	0	0.0	0.0
2	1	0	0	0	0.0	0.0
3	92	0	0	4	0.0	0.0
4	311	2	0	546	30.0	1.0
5	631	0	0	2,990	203.5	0.5
6	1,114	0	0	5,570	427.8	2.0
7	1,422	5	0	7,110	814.0	1.0
8	984	12	0	4,920	1,055.8	5.5
Total	4,555	19	0	21,140	2,531.0	10.0
		vd: 2	vd: 0		vd: 583.5	vd: 2.8
		vi: 0	vi: 0		vi: 1.8	vi: 0.0
		vs: 0	vs: 0		vs: 109.8	vs: 2.5
		os: 17	os: 0		os: 1,842.5	os: 4.8

TABLE 10. Number of samples in terms of attributes that first appear in text passage.

First attribute	Training data	Validation data	Test data	Selected data for human evaluation
`name`	25,783	2,787	2,727	19
`eatType`	1,608	260	299	19
`food`	2,573	137	197	18
`priceRange`	2,426	165	230	19
`customer rating`	1,389	167	83	18
`area`	2,773	407	363	19
`familyFriendly`	2,000	258	267	19
`near`	2,008	308	389	19
Total	40,560	4,489	4,555	150

c: FOCUS

In this experiment, we assumed that the emphasized attribute-value pair appears first in the text passage [22]. The evaluators answered the following question with either "*yes*" or "*no*":

- Is [(`attribute`) value] the focused attribute-pair in the sentence?

Here "[(`attribute`) value]" is the attribute-value pair which appears first in each text passage.

The results are shown in Table 11. The naturalness scores of the three baselines are comparable: TGEN, JUG, and

TABLE 11. Results for human evaluation in naturalness with 95% confidence interval, adequacy, and focus. Bold indicates best result.

Method	Naturalness				Adequacy		Focus
	Grammatical	Comprehensible	Natural	Acceptable	MR value	MR value & order	
Reference	4.12±0.13	4.96±0.10	3.91±0.13	5.15±0.09	89.11	88.89	86.22
TGEN	4.76±0.10	5.24±0.08	**4.86**±0.09	5.35±0.08	92.22	4.22	56.44
JUG	4.78±0.10	**5.28**±0.07	4.77±0.10	**5.36**±0.07	94.44	4.00	53.78
Transformer w/o order	**4.80**±0.10	5.27±0.07	4.86±0.09	**5.36**±0.08	**95.56**	5.56	55.33
Transformer w/ order	4.44±0.12	5.10±0.09	4.36±0.12	5.26±0.09	94.67	**92.89**	**91.33**
Transformer w/ order (augmented training data)	4.47±0.12	5.08±0.09	4.36±0.12	5.27±0.09	94.89	92.44	90.67

TABLE 12. Examples with five lowest scores for human evaluations (G: Grammatical; C: Comprehensible; N: Natural; A: Acceptable).

Metrics	Score	Method	Text
G	1.33	TGEN	NAME is a children friendly pub in the city centre near NEAR. It is in the high price range.
	1.33	Reference	5 out of 5 for this pub, although no facilities for children. It is close to NEAR, called NAME near to the riverside has a price list of more than £30 serves Japanese cuisine.
	1.67	Reference	There is a children friendly English restaurant in the riverside area. It is high price range. It is called NAME, and is located near NEAR.
	1.67	Transformer w/ order	Moderately priced NAME is a non kid friendly restaurant located in the city centre, near NEAR. It serves Chinese food.
	1.67	Transformer w/ order	There is a 5 out of 5 pub that is not children friendly near NEAR called NAME in the riverside area. The price range is more than £30 and Japanese food.
C	3.00	Transformer w/ order	Near NEAR is a restaurant that is family friendly in the riverside area called NAME. It serves Italian food and is cheap.
	3.00	Reference	A kids friendly Japanese pub along the riverside is called NAME and is next to NEAR.
	3.33	Reference	Near NEAR their is a restaurant that is family friendly along the riverside named NAME which serves Italian food on a cheap price range.
	3.33	TGEN	NAME is a children friendly pub in the city centre near NEAR. It is in the high price range.
	3.33	Reference	Located near NEAR in the riverside area is an eat type pub called NAME is a children friendly that serves Japanese food and has a price range more than £30.
N	1.33	TGEN	NAME is a children friendly pub in the city centre near NEAR. It is in the high price range.
	1.33	Reference	5 out of 5 for this pub, although no facilities for children. It is close to NEAR, called NAME near to the riverside has a price list of more than £30 serves Japanese cuisine.
	1.67	Reference	3 out of 5 star restaurant style restaurant NAME offers child friendly atmosphere near NEAR.
	2.00	Reference	If you are looking for a high quality, family-friendly dining experience in the heart of city centre, NAME is for you. This pub I near NEAR and serves fast-food like it's gourmet.
	2.00	Reference	Near NEAR by riverside is a pub that is yes family friendly with a low customer rating called NAME and the prices are less than £20.
A	3.67	Reference	Located near NEAR in the riverside area is an eat type pub called NAME is a children friendly that serves Japanese food and has a price range more than £30.
	3.67	Transformer w/o order	NAME is a Japanese pub in riverside near NEAR. It is children friendly and has a high customer rating.
	3.67	Transformer w/ order (augmented training data)	Moderately priced NAME is a non kid friendly restaurant located in the city centre. It serves Chinese food and is near NEAR.
	3.67	TGEN	NAME is a children friendly Japanese pub in the city centre near NEAR with a high price range and a customer rating of 5 out of 5.
	3.67	Reference	5 out of 5 rated child friendly NAME, serves Japanese food in a pub at higher than normal prices. Located in the city centre near NEAR.

TABLE 13. One example for an MR(`name`: NAME (order=3), `eatType`: restaurant (order=2), `customer rating`: 3 out of 5 (order=1), `familyFriendly`: yes (order=4), and `near`: NEAR (order=5)) where TGEN and JUG have good scores and reference has bad scores for human evaluation (G: grammatical; C: comprehensible; N: natural; A: acceptable).

Method	Text	Score			
		G	C	N	A
Reference	3 out of 5 star restaurant style restaurant NAME offers child friendly atmosphere near NEAR.	1.67	4.67	1.67	5.33
TGEN	NAME is a three star family friendly restaurant located near NEAR.	6.00	5.67	6.00	6.00
JUG	NAME is a family friendly restaurant with a customer rating of 3 out of 5. It is located near NEAR.	6.00	5.67	6.00	6.00
Transformer w/o order	NAME is a family friendly restaurant located near NEAR. It has a customer rating of 3 out of 5.	6.00	5.67	5.67	6.00
Transformer w/ order	There is a 3 star restaurant NAME that is family friendly located near NEAR.	4.00	5.33	4.33	5.67
Transformer w/ order (augmented training data)	There is a three star restaurant NAME that is family friendly located near NEAR.	5.00	5.67	4.00	6.00

Transformer w/o order. On the other hand, the scores for our proposed methods with order constraints and reference are slightly lower than those of the baselines, because it is natural for restaurant recommendation sentences to start with a `name`

TABLE 14. One example of MR value, MR order, and generated text passages.

		Attribute	Value	Order
MR		name	NAME (THE WATERMAN)	1
		eatType	pub	2
		food	Italian	5
		priceRange	less than £20	6
		customer rating	(empty)	0
		area	city centre	3
		familyFriendly	no	7
		near	NEAR (RAJA INDIAN CUISINE)	4
Text	Reference	NAME pub is located in the city centre area near NEAR. It has Italian food in the £20 or less price range and is not family-friendly.		
	TGEN	NAME is an Italian pub located in the city centre near NEAR. It is not family-friendly and has a price range of less than £20.		
	JUG	NAME is a pub in the city centre near NEAR. It serves Italian food for less than £20. It is not family-friendly.		
	Transformer w/o order	NAME is a pub that serves Italian food. It is located in the city centre near NEAR. It is not family-friendly and has a price range of less than £20.		
	Transformer w/ order	NAME is a pub in the city centre near NEAR. It serves Italian food for less than £20 and is not family-friendly.		
	Transformer w/ order (augmented training data)	NAME is a pub in the city centre near NEAR. It serves Italian food for less than £20 and is not family-friendly.		

TABLE 15. Another example of generated text passages: MR value is identical to Table 14, although MR order is different.

		Attribute	Value	Order
MR		name	NAME (THE WATERMAN)	1
		eatType	pub	3
		food	Italian	4
		priceRange	less than £20	5
		customer rating	(empty)	0
		area	city centre	7
		familyFriendly	no	2
		near	NEAR (RAJA INDIAN CUISINE)	6
Text	Reference	NAME is a non family-friendly pub that serves Italian food for less than £20. It is located near NEAR in the city centre area.		
	TGEN	NAME is an Italian pub located in the city centre near NEAR. It is not family-friendly and has a price range of less than £20.		
	JUG	NAME is a pub that serves Italian food for less than £20. It is located in the city centre near NEAR and is not family-friendly.		
	Transformer w/o order	NAME is a pub that serves Italian food. It is located in the city centre near NEAR. It is not family-friendly and has a price range of less than £20.		
	Transformer w/ order	NAME is a non family-friendly pub serving Italian food for less than £20 near NEAR in the city centre.		
	Transformer w/ order (augmented training data)	NAME is a non family-friendly pub that serves Italian food for less than £20. It is located near NEAR in the city center.		

TABLE 16. Another example of generated text passages: Except the MR value of eatType, all MR values and MR orders are identical to Table 14.

		Attribute	Value	Order
MR		name	NAME (THE WATERMAN)	1
		eatType	restaurant	2
		food	Italian	5
		priceRange	less than £20	6
		customer rating	(empty)	0
		area	city centre	3
		familyFriendly	no	7
		near	NEAR (RAJA INDIAN CUISINE)	4
Text	Reference	NAME restaurant is located in the city centre area near NEAR. It has Italian food in the £20 or less price range and is not family-friendly.		
	TGEN	NAME is a non family-friendly Italian restaurant in the city centre near NEAR with a price range of less than £20.		
	JUG	NAME is a restaurant that serves Italian food for less than £20. It is located in the city centre near NEAR and is not family-friendly.		
	Transformer w/o order	NAME is a non family-friendly Italian restaurant located in the city centre near NEAR. It has a price range of less than £20.		
	Transformer w/ order	NAME is a restaurant in the city centre near NEAR. It serves Italian food for less than £20 and is not family-friendly.		
	Transformer w/ order (augmented training data)	NAME is a restaurant located in the city centre near NEAR. It serves Italian food for less than £20. It is not family-friendly.		

attribute, and sentences starting with other attributes lose some naturalness. Table 10 shows that the text passages in more than half of the E2E refined dataset start with the name attribute. This distribution causes the value of the name attribute to always appears first in the generated text passages of the baseline methods. Table 12 also lists the five examples

An AI system generates sentences from MR (meaning representation) data. Each MR data has up to 8 attribute-value pairs.

```
[MR data]
[1] (name) Blue Spice
[2] (eatType) pub
[3] (food) English
[4] (area) riverside
[5] (familyFriendly) no
[6] (near) Rainbow Vegetarian Cafe

[sentence]
(A) There is a riverside pub near the Rainbow Vegetarian Cafe called Blue Spice. It serves English food but is not family-friendly.
(B) In the riverside area is a pub near Rainbow Vegetarian Cafe called Blue Spice. It serves English food and is not family-friendly.
(C) In riverside, there is a pub near Rainbow Vegetarian Cafe called Blue spice that serves English food. It is not family-friendly.
(D) Blue Spice is an English pub near Rainbow Vegetarian Cafe in the riverside area. It is not family-friendly.
(E) Blue Spice is a pub that serves English food. It is located in riverside near Rainbow Vegetarian Cafe and is not family-friendly.
```

In this case, the MR data has six attribute-value pairs, "[1] (name) Blue Spice", "[2] (eatType) pub", "[3] (food) English", "[4] (area) riverside", "[5] (familyfriendly) no", and "[6] (near) Rainbow Vegetarian Cafe". The AI system can generate sentences by meeting all attribute-value pairs. The system can generate not only one sentence but 2~6 sentences with different styles, as shown above.
In this study, all variations of values for each attribute are listed below:

```
(name) The Cricketers, Giraffe, Blue Spice, etc.
(eatType) coffee shop, pub, restaurant
(food) Chinese, English, French, Indian, Italian, Japanese, fast food
(priceRange) cheap, expensive, less than £20, moderate, more than £30, £20-25
(customer rating) 1 out of 5, 3 out of 5, 5 out of 5, average, high, low
(area) city centre, riverside
(familyFriendly) no, yes
(near) All Bar One, Crowne Plaza Hotel, Raja Indian Cuisine, etc.
```

Sometimes, the values appear in the sentence with different expressions.

```
[MR data]
[1] (name) The Wrestlers
[2] (eatType) restaurant
[3] (food) Japanese
[4] (priceRange) expensive
[5] (area) riverside
[6] (familyFriendly) yes
[7] (near) Raja Indian Cuisine

[sentence]
There is a high price range Japanese restaurant in riverside near Raja Indian Cuisine that is child friendly called The Wrestlers.
```

In this case, the value "expensive" of the (priceRange) attribute does not appear in the sentence. However, "high price" means expensive, so we can say that the generated sentence meets the attribute-value pair of "(priceRange) expensive".

Some examples are listed below:

```
(food) Italian: pasta, spaghetti, etc.
(priceRange) cheap: inexpensive, low price, etc.
(priceRange) moderate: average price, moderately priced, etc.
(priceRange) expensive: high price, upper range price, etc.
(customer rating) 1 out of 5: one star, one to five, 1 star, etc.
(familyFriendly) yes: family friendly, child friendly, children friendly, kid friendly, etc.
(familyFriendly) no: not family friendly, no child friendly, non-children friendly, not kid friendly, etc.
```

FIGURE 2. Instructions for human evaluation.

with the lowest naturalness scores, and the generated text passages with lower scores in terms of naturalness tend to start with attributes other than `name`. However, looking at the acceptable scores in Table 11, there is no significant difference between the baseline methods and our proposed methods, and although the order control has slightly lost some naturalness, our proposed methods can generate good English without any problems. Table 13 shows an example of the generated text passages for an MR. The reference text's style is rather free, whereas the others resemble templates. This

Question (A) asks if the generated sentence meets all the attribute-value pairs.

```
[MR data]
[1] (near) The Cricketers
[2] (area) city centre
[3] (eatType) pub
[4] (name) The Mill
[5] (familyFriendly) no
[6] (food) fast food
[7] (priceRange) moderate

[sentence]
Near The Cricketers in city centre is the pub The Mill. This adult establishment serves fast food at average prices.
```

In this case, the sentence meets all seven attribute-value pairs. Here, the value "moderate" of the (priceRange) attribute is replaced by "average prices" in the sentence. However, in this case, "moderate" and "average" are the same in meaning. Furthermore, the value "no" of the (familyFriendly) attribute is expressed as "adult establishment". Thus, "Yes" should be chosen for the Question (A).

```
[MR data]
[1] (near) All Bar One
[2] (name) Clowns
[3] (eatType) pub
[4] (customer rating) 3 out of 5
[5] (food) Chinese

[sentence]
Clowns is a coffee shop near All Bar One. It is rated 3 out of 5 and is not family-friendly.
```

In this case, there are three errors.

- The value "Chinese" of the (food) attribute does not appear in the sentence.
- The value "pub" of the (eatType) attribute is changed to "coffee shop" in the sentence.
- The MR has no value for the (familyFriendly) attribute, but the phrase "is not family-friendly" is appeared in the sentence. Thus, "No" should be chosen for the Question (A).

Question (B) asks if the generated sentence meets all the attribute-value pairs in collect order.

```
[MR data]
[1] (eatType) coffee shop
[2] (area) city centre
[3] (name) Blue Spice

[sentence]
A coffee shop in the city centre area called Blue Spice.
```

In this case, the sentence meets all three attribute-value pairs. Also, the order of the attribute-value in the generated sentence is "[1] (eatType) coffee shop" -> "[2] (area) ceity centre" -> "[3] (name) Blue Spice", so the sentence meets the order of the attribute-value pairs in the MR data. Thus, "Yes" should be chosen for the Question (B).

```
[MR data]
[1] (name) Blue Spice
[2] (eatType) restaurant
[3] (food) Chinese
[4] (area) riverside
[5] (near) The Valuts

[sentence]
Blue Spice is a Chinese restaurant in riverside near The Vaults.
```

In this case, the sentence meets all five attribute-value pairs. However, the order of the attribute-value in the generated sentence is "[1] (name) Blue Spice" -> "[3] (food) Chinese" -> "[2] (eatType) restaurant" -> "[4] (area) riverside" -> "[5] (near) The Vaults". Hence, the sentence does not meet the order of the attribute-value pairs in the MR data. Thus, "No" should be chosen for the Question (B).

FIGURE 3. Instructions for human evaluation for adequacy.

situation might explain why the reference scores are worse than the others.

For adequacy, we found that the generated text passages of the proposed methods appropriately met almost all the MR values and orders. A comparison of the adequacy results with those of MRcheck (Table 8) identifies a clear correlation, suggesting that MRcheck works as an effective automatic measure of the reliability of synthetic data.

Suppose you are looking for a restaurant/coffee shop/pub and ask an AI system to recommend a restaurant/coffee shop/pub for you. The AI system extracts some attribute-value pairs from your question, then generates a response according to the extracted attribute-value pairs.

```
[Your question]
I'm in the city centre now. Can you tell me any Chinese restaurants around here?

[MR data]
[1] (area) city centre
[2] (food) Chinese
[3] (eatType) restaurant

[sentence]
In the city centre, there is a Chinese restaurant called The Wrestler.
```

The AI system extracts three attribute-value pairs from your question, then generates the response. In this case, the system searched the restaurant "The Wreslter" from its database, so the generated sentence includes the name.

According to the question, the system supposes that it is important for the restaurant to be placed in the city centre. So, the AI system would emphasize the attribute-value pair "(area) city centre" in the generated sentence. The emphasized attribute-value pair is named "focused attribute-value pair" in the subsequent studies.

FIGURE 4. Instructions for human evaluation for focus.

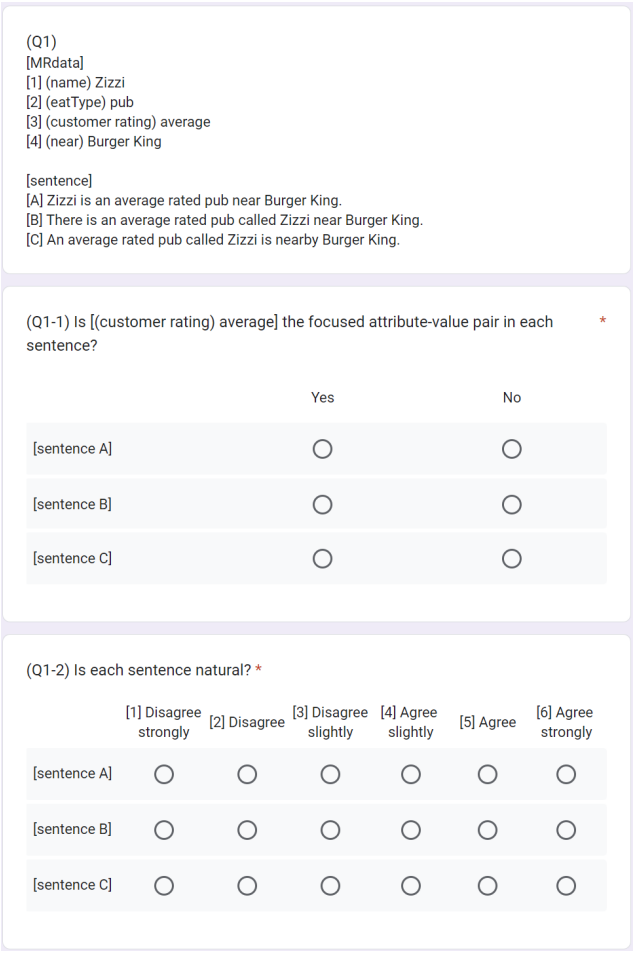

FIGURE 5. Screenshot of human evaluation for naturalness and focus (1).

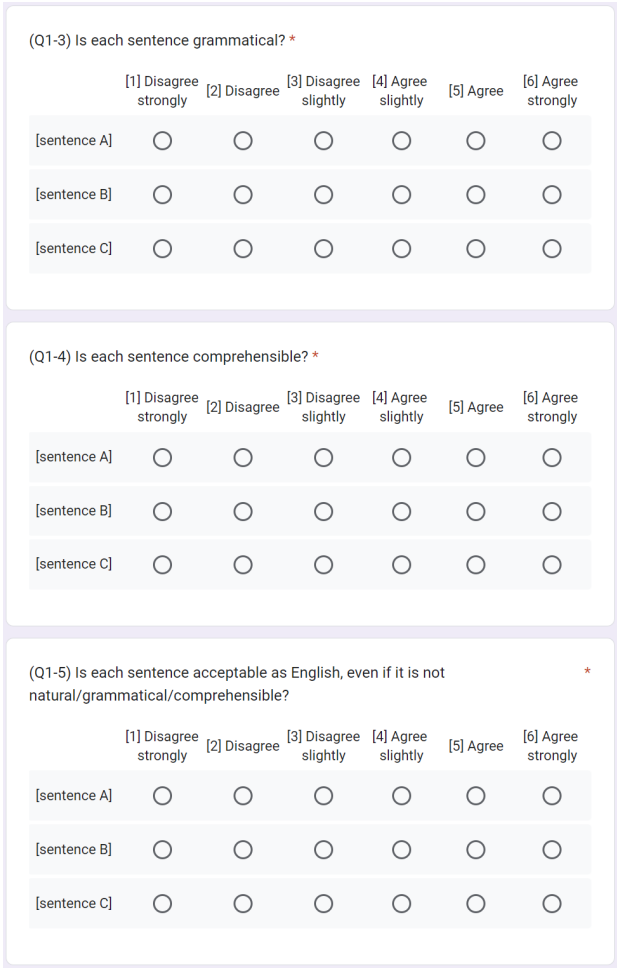

FIGURE 6. Screenshot of human evaluation for naturalness and focus (2).

For focus, the scores of our proposed methods significantly outperformed those of the baselines. A comparison of the adequacy and focus results shows a clear correlation. This means that correct control of the content order also allows for correct control of the emphasized attribute-value pair in the generated sentences.

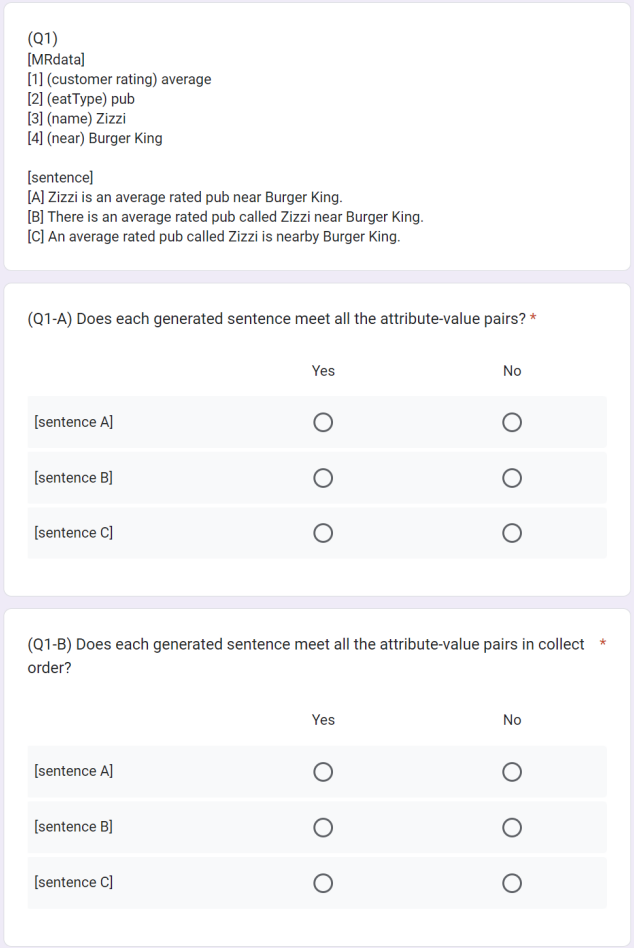

FIGURE 7. Screenshot of human evaluation for adequacy.

VII. CONCLUSION

We proposed an MR-to-text method that controls the order of the MR values in generated text passages using MR order constraints. Our proposed method worked effectively and precisely controlled the content order in automatic evaluations. Data augmentation also effectively improved the performance using the MR2T and T2MR models to balance the data distribution in terms of non-empty attributes. The human evaluation results suggest that our proposed methods can focus attribute-value pairs for correct emphasis by controlling the content order. Even though the proposed methods suffered slightly less naturalness compared with the baselines, they generated proper English sentences without any problem. Future work will control such other aspects as the text structure and the output length and apply such methods to other MR-to-text datasets.

APPENDIX A
EXAMPLES

Some examples are shown in Tables 14, 15, and 16. Comparing Tables 14 and 15, the Transformer models with order constraints accurately reflected the MR values and the MR order in the generated text passages, although TGEN, JUG, and the Transformer models without order constraints did not. Comparing Tables 14 and 16, the Transformer models with order constraints properly preserved the MR order in the generated text passages, although TGEN, JUG, and the Transformer models without order constraints did not.

APPENDIX B
INSTRUCTIONS FOR HUMAN EVALUATION

The instructions for those participating in the human evaluation are shown in Fig. 2. The instructions for adequacy and focus are shown in Figs. 3 and 4. Screenshots of the human evaluation are shown in Figs. 5, 6, and 7. These systems were designed using Google Forms.

REFERENCES

[1] Z. Kasner and O. Dusek, "Neural pipeline for zero-shot data-to-text generation," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics (Long Papers)*, vol. 1, 2022, pp. 3914–3932.
[2] Y. Leng, F. Portet, C. Labb, and R. Qader, "Controllable neural natural language generation: Comparison of state-of-the-art control strategies," in *Proc. 3rd Int. Workshop Natural Lang. Gener. From Semantic Web, 2020*, pp. 34–39.
[3] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2001, pp. 311–318.
[4] C.-Y. Lin and E. Hovy, "Manual and automatic evaluation of summaries," in *Proc. ACL Workshop Autom. Summarization*, 2002, pp. 74–81.
[5] M. Denkowski and A. Lavie, "Meteor universal: Language specific translation evaluation for any target language," in *Proc. 9th Workshop Stat. Mach. Transl.*, 2014, pp. 376–380.
[6] Y. Su, D. Vandyke, S. Wang, Y. Fang, and N. Collier, "Plan-then-generate: Controlled data-to-text generation via planning," in *Proc. Findings Assoc. Comput. Linguistics (EMNLP)*, 2021, pp. 895–909.
[7] Y. Su, Z. Meng, S. Baker, and N. Collier, "Few-shot table-to-text generation with prototype memory," in *Proc. Findings Assoc. Comput. Linguistics (EMNLP)*, 2021, pp. 6493–6504.
[8] K. Toyama, K. Sudoh, and S. Nakamura, "E2E refined dataset," 2022, *arXiv:2211.00513*.
[9] J. Novikova, O. Dušek, and V. Rieser, "The E2E dataset: New challenges for end-to-end generation," in *Proc. 18th Annu. SIGdial Meeting Discourse Dialogue*, 2017, pp. 201–206.
[10] O. Dušek, J. Novikova, and V. Rieser, "Findings of the E2E NLG challenge," in *Proc. 11th Int. Conf. Natural Lang. Gener.*, 2018, pp. 322–328.
[11] O. Dušek, D. M. Howcroft, and V. Rieser, "Semantic noise matters for neural natural language generation," in *Proc. 12th Int. Conf. Natural Lang. Gener.*, 2019, pp. 421–426.
[12] T. C. Ferreira, H. Vaz, B. Davis, and A. Pagano, "Enriching the E2E dataset," in *Proc. 14th Int. Conf. Natural Lang. Gener.*, 2021, pp. 177–183.
[13] C. Kedzie and K. McKeown, "A good sample is hard to find: Noise injection sampling and self-training for neural language generation models," in *Proc. 12th Int. Conf. Natural Lang. Gener.*, 2019, pp. 584–593.
[14] J. Chai, R. Pryzant, V. Ye Dong, K. Golobokov, C. Zhu, and Y. Liu, "FAST: Improving controllability for text generation with feedback aware self-training," 2022, *arXiv:2210.03167*.
[15] O. Dušek and F. Jurcicek, "Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics (Short Papers)*, vol. 2, 2016, pp. 45–51.
[16] B.-H. Tseng, J. Cheng, Y. Fang, and D. Vandyke, "A generative model for joint natural language understanding and generation," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 1795–1807.
[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, and I. Polosukhin, "Attention is all you need," in *Proc. 31st Conf. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.
[18] D. P. Kingma and J. L. Ba, "ADAM: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent.*, 2015, pp. 1–15.
[19] G. Doddington, "Automatic evaluation of machine translation quality using n-gram co-occurrence statistics," in *Proc. 2nd Int. Conf. Hum. Lang. Technol. Res.*, 2002, pp. 138–145.

[20] R. Vedantam, C. L. Zitnick, and D. Parikh, "CIDEr: Consensus-based image description evaluation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4566–4575.
[21] J. Amidei, P. Piwek, and A. Willis, "The use of rating and Likert scales in natural language generation human evaluation tasks: A review and some recommendations," in *Proc. 12th Int. Conf. Natural Lang. Gener.*, 2019, pp. 397–402.
[22] N. Suzuki and S. Nakamura, "Representing 'how you say' with 'what you say': English corpus of focused speech and text reflecting corresponding implications," in *Proc. Interspeech*, Sep. 2022, pp. 4980–4984.

KEISUKE TOYAMA (Graduate Student Member, IEEE) received the B.E. degree in engineering from Nagoya University, Japan, in 1996, the M.E. degree in engineering from the Nara Institute of Science and Technology, Japan, in 1998, and the M.Sc. degree (Hons.) in electronic engineering from the Queen Mary University of London, U.K., in 2008. He is currently pursuing the Ph.D. degree with the Nara Institute of Science and Technology.

Since 1998, he has been with Sony Corporation (renamed Sony Group Corporation, in 2021) as a Research Engineer of audio and music signal processing field.

Mr. Toyama is a member of the Acoustical Society of Japan.

KATSUHITO SUDOH received the B.S. degree in engineering and the M.S. and Ph.D. degrees in informatics from Kyoto University, in 2000, 2002, and 2015, respectively.

He was with NTT Communication Science Laboratories, from 2002 to 2017. He is currently an Associate Professor with the Nara Institute of Science and Technology. His research interests include machine translation and natural language processing.

Dr. Sudoh is a member of the Association for Computational Linguistics (ACL) and the International Speech Communication Association (ISCA).

SATOSHI NAKAMURA (Life Fellow, IEEE) received the B.S. degree from the Kyoto Institute of Technology, in 1981, and the Ph.D. degree from Kyoto University, in 1992.

From 1994 to 2000, he was an Associate Professor with the Graduate School of Information Science, Nara Institute of Science and Technology, Japan. He was the Director of ATR Spoken Language Communication Research Laboratories, from 2000 to 2008, and the Vice President of ATR, from 2007 to 2008. He was the Director General of Keihanna Research Laboratories and the Executive Director of the Knowledge Creating Communication Research Center, National Institute of Information and Communications Technology, Japan, from 2009 to 2010. He is currently a Professor with the Graduate School of the Science and Technology, Nara Institute of Science and Technology, and an Honorary Professor with the Karlsruhe Institute of Technology, Germany. He is also the Director of the Augmented Human Communication Laboratory and a Full Professor with the Graduate School of Information Science, Nara Institute of Science and Technology. He is one of the leaders of speech-to-speech translation research and has been serving for many years on various worldwide speech-to-speech translation research projects, including C-STAR, IWSLT, and A-STAR. His research interests include modeling and systems of speech-to-speech translation and speech recognition.

Prof. Nakamura has been an Elected Board Member of the International Speech Communication Association (ISCA), since June 2011, an Editorial Board Member of *IEEE Signal Processing Magazine*, since April 2012, and a Technical Committee Member of IEEE SPS Speech and Language, since 2013. He received the following awards: the Yamashita Research Award, the Kiyasu Award from the Information Processing Society of Japan, the Telecom System Award, the AAMT Nagao Award, the Docomo Mobile Science Award in 2007, and the ASJ Award for Distinguished Achievements in Acoustics. He received a Commendation for Science and Technology from the Minister of Education, Science and Technology and the Commendation for Science and Technology from the Minister of Internal Affair and Communications. He also received the LREC Antonio Zampolli Award in 2012. He is a ISCA Fellow, IPSJ Fellow, and ATR Fellow.