Masked Hard Coverage Mechanism on Pointer-generator Network for Natural Language Generation

Ting Hu and Christoph Meinel
Hasso Plattner Institute, Potsdam University, Potsdam, Germany

Keywords: Natural Language Generation, Coverage Mechanism, Pointer-generator Network.

Abstract: Natural Language Generation (NLG) task is to generate natural language utterances from structured data. Seq2seq-based systems show great potentiality and have been widely explored for NLG. While they achieve good generation performance, over-generation and under-generation issues still arise in the generated results. We propose maintaining a masked hard coverage mechanism in the pointer-generator network, a seq2seq-based architecture that trains a switch policy to produce output sequences by partially copying from input structured data. The proposed mechanism can be regarded as the inner controlling module to keep track of the copying history and force the network to generate sentences accurately covering all information provided in structured data. Experimental results show that our coverage mechanism alleviates the over-generation and under-generation issues and achieves decent performance on the E2E NLG dataset.

1 INTRODUCTION

Natural Language Generation (NLG) is converting structured data, like Mean Representations (MRs), into meaningful sentences in the form of natural language. An MR is an unordered set of attribute(slot)-value pairs, where the attribute is a string, and the value is a sequence of words, or rather, keywords. For instance, in the restaurant domain, an MR and the corresponding reference are as follows.

MR: *name[Blue Spice], eatType[coffe shop], customerRating[1 out of 5].*

Reference: *The Blue Spice coffee shop has a customer rating of 1 out of 5.*

In the MR, *name[Blue Spice]* is a typical attribute[value] pair, and words *Blue Spice, coffee shop, 1 out of 5* are considered as keywords. NLG systems are supposed to produce fluent and coherent utterances that cover all the information provided in the MRs.

Traditional NLG systems are mainly templates-based (Smiley et al., 2018) and rules-based (Nguyen and Tran, 2018), which usually require mined or manually designed templates and handcrafted rules. Data-driven sequence-to-sequence models (Chen et al., 2018; Gong, 2018) become the mainstream on NLG tasks. These models can generate fluent and diverse utterances, whereas large amounts of data are supposed to be available for training. In the case of insufficient data, other strategies like data augmentation are utilized.

The recent emergence of pre-trained models (Radford et al., 2019; Yang et al., 2019) in the NLP field has provided us with new possibilities. The pre-trained models have yet been trained on massive amounts of text data and have robust inferencing capability. Based on them, only a small amount of data in a specific domain is required for fine-tuning and good results can be obtained on downstream tasks. Chen et al. (2019) propose a few-shot NLG system with the pre-trained GPT-2 (Radford et al., 2019) as the decoder of a seq2seq framework. They train a switch module so that the model learns when to copy from the input MR and when to generate words by the decoder, similar to other pointer-generator networks (Gu et al., 2016; See et al., 2017). Though based on the powerful pre-trained GPT-2 model, it still suffers from over-generation and under-generation problems like other NLG systems. Over-generation means that some keywords repeat themselves in the generated sentences, or unspecified keywords are generated in the utterances. Under-generation means the generated sentences' incomplete coverage of the information given in MRs.

We analyze that the reason is the lack of a precise internal control mechanism to record the coverage history of attribute-value pairs during decoding and to constrain the model to have all given informa-

Hu, T. and Meinel, C.
Masked Hard Coverage Mechanism on Pointer-generator Network for Natural Language Generation.
DOI: 10.5220/0010341211771183
In *Proceedings of the 13th International Conference on Agents and Artificial Intelligence (ICAART 2021) - Volume 2*, pages 1177-1183
ISBN: 978-989-758-484-8

tion exactly covered in generation results. This paper proposes introducing a masked hard coverage mechanism into the few-shot pointer-generator network (Chen et al., 2020) to impose robust coverage control on slot-value pairs. We borrow the term "masked" from pre-trained models, indicating the coverage control is merely exerted on these keywords of the input MRs. "Hard" demonstrates that the computation of the coverage control at each slot is hard, that is, all or nothing. Moreover, one additional loss term is brought in the loss function to penalize inappropriate sentence-level coverage. We show that the proposed coverage mechanism and the additional loss term effectively mitigate over-generation and under-generation issues, and yield better generation results on E2E NLG dataset compared with the original few-shot pointer-generator network.

2 RELATED WORK

NLG Systems. Manually designed rules or templates mined from data are indispensable in traditional NLG systems (Mille and Dasiopoulou, 2017; Nguyen and Tran, 2018; Smiley et al., 2018; Puzikov and Gurevych, 2018). They can generate sentences covering all information in the MRs, yet with a lack of diversity. Recent neural-based seq2seq systems (Dušek and Jurčíček, 2016; Juraska et al., 2018; Chen et al., 2018; Agarwal et al., 2018) are widely explored, for example, in the E2E NLG challenge. These systems achieve good generation results on automatic evaluation metrics and human evaluation, while large amounts of training data are the necessities. Under the circumstances of small amounts of training data, Elsahar et al. (2018) propose zero-shot learning to generate questions from knowledge graphs, wherein many training instances are required before entering the transfer learning stage. Chen et al. (2020) put forward a few-shot learning NLG framework with a pre-trained language model, where a small number of training instances contribute to good generation quality on the WIKIBIO NLG task.

Coverage Mechanism. Neural Machine Translation systems suffer from the common problem of the lack of coverage; namely, not all source words are covered or translated in target sentences. Tu et al. (2016) propose a coverage mechanism to alleviate over-translation and under-translation problems. A coverage vector is maintained during the decoding process to facilitate future generation decisions and significantly improve the alignment between source and target sentences. See et al. (2017) propose a more straightforward method of accumulating the coverage vector and define an additional coverage loss to penalize attention repetition in text summarization. Considering that some words are allowed to appear multiple times when generating summarizations, their coverage loss is not strict enough for our NLG task, in which all essential information in input MRs should be exactly covered.

Our work is based on the framework of Chen et al. (2020). The differences lie in that: (i) We apply a masked hard coverage mechanism on all keywords in the MRs. To be specific, "masked" means the coverage vector is only updated when keywords appear in generation result. "Hard" means each element's value in the coverage vector is 0 or 1, with no other values in between. (ii) We employ one additional loss term in the loss function to force the model to cover all the given information and avoid repeated coverage. Our coverage loss term is a more robust constraint than that of See et al. (2017).

3 COVERAGE MECHANISM

The original framework (Chen et al., 2020) consists of a field-gating table encoder with dual attention (Liu et al., 2018) and a pre-trained language model GPT-2 (Radford et al., 2019) as the decoder. We list one example of the input and the expected output of the framework in Figure 1. The attribute-value pairs and the relative position information of these keywords are concatenated and fed into the encoder. The input of the decoder is the reference, together with the keywords' attribute and relative position information. The model aims to train a switch module to control when to copy keywords from input MRs and when to let the decoder generate tokens independently.

At each time step t, the switch policy controlled by p_{copy} is computed by

$$p_{copy} = \sigma(W_h h_t^* + W_s s_t + W_x x_t + b)$$
$$h_t^* = \sum_i \gamma_t^i h_i, \tag{1}$$

Where h_t^* denotes the context vector, which is the weighted sum of all encoder hidden states $\{h_i\}$, s_t is the decoder hidden state, and x_t is the decoder input. The dual attention weight distribution γ_t is the element-wise production between word-level attention weight α_t known as vanilla attention (Bahdanau et al., 2014) and attribute-level attention weight β_t (Liu et al., 2018),

$$\gamma_t^i = \alpha_t^i \cdot \beta_t^i. \tag{2}$$

Meanwhile, a coverage vector is maintained to record which keywords have been copied. Assume all key

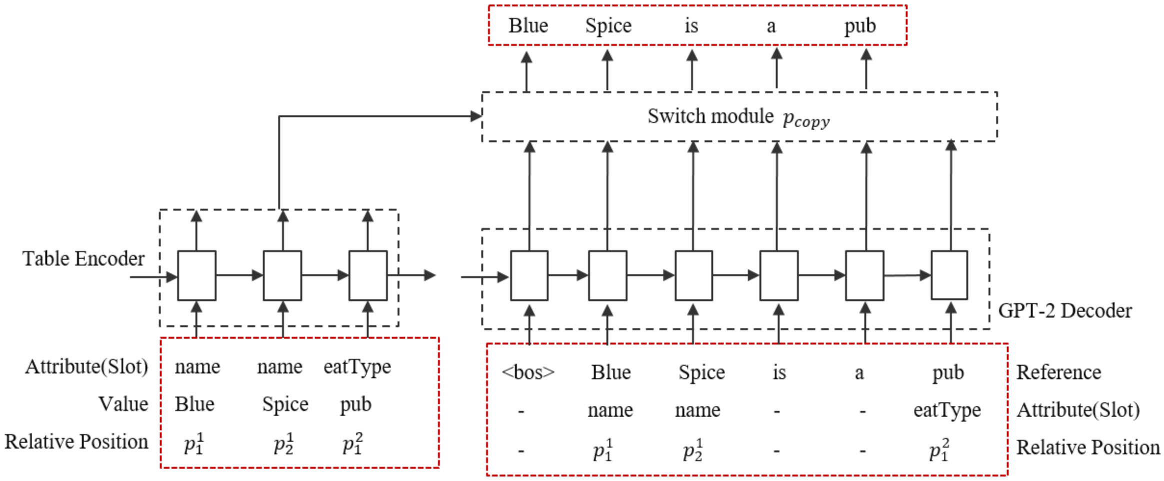

Figure 1: The architecture of the employed pointer-generator network. The inputs and expected output of the model are within three red dashed boxes, respectively. Some connections are omitted for clarity. The coverage vector c is only updated at the positions where the words in the reference are from the given MRs.

words in the given MR is $x = [x_1, x_2, x_3, x_4]$. The initial coverage vector is $c_0 = [0,0,0,0]$ indicating that no keyword has been copied. For the next time step, if word x_1 is copied, the coverage vector turns to $c_1 = [1,0,0,0]$, or if the generated word is not from the input MR, the coverage vector keeps unchanged $c_1 = [0,0,0,0]$. In general, the masked hard coverage vector is updated as:

$$c_t = c_{t-1} + \text{one hot}(\arg\max(\gamma_t)) * m_t \tag{3}$$

where m_t with a value of 1 or 0 is the mask denoting whether the word belongs to keywords or not, respectively. The coverage vector indicating copying history also affects how dual attention weight γ_t is obtained by

$$\begin{aligned} g_t &= tanh(W_c c_{t-1} + b_c) \\ \gamma_t^j &= \gamma_t^j \cdot g_t \end{aligned} \tag{4}$$

where g_t is coverage penalty that has further impact on the switch policy through Eq. 1. In other words, to pay attention to the same keywords of input MR is to be penalized. Hence, the switch module actually make a decision to copy or to generate with the past copy history taken into consideration.

Furthermore, In order to force the model to attend to all key words in input MRs and avoid copying the same words, we employ one additional coverage loss term in the loss function to constrain the coverage vectors to be close to all-ones vectors, as the third term in the loss function,

$$L = L_c + \alpha \sum_{w_j \in K} (1 - p_{copy}^j) + \beta ||c - \mathbf{1}||_2^2 \tag{5}$$

where c denotes the final coverage vectors of all instances, K denotes the set of all keywords in the MRs that appear in output sequences, and weights α and β are to be tuned. The first term is the reconstruction loss between the input and output of the model, and the second term is the switch policy, where p_{copy} is maximized at the positions where keywords occur in output sequences.

For comparison, we describe another coverage mechanism of the pointer-generator network (See et al., 2017) proposed for text summarization. According to the soft coverage mechanism, when employed in the current framework, the coverage vector c_t is accumulated by the dual attention distribution of each decoder timestep

$$c_t = c_{t-1} + \gamma_t. \tag{6}$$

In other words, the coverage vector is maintained to record the copying history of all words. To penalize repetitive attention to the same words, the coverage loss of each timestep is

$$covloss_t = \sum_i \min(\gamma_t^j, c_t^i). \tag{7}$$

Apparently, for text summarization, the facts that some words may even occur multiple times and uniform coverage is not required result in the above soft coverage mechanism. By contrast, our masked hard coverage mechanism offers more powerful constraints on keywords coverage for NLG tasks. In the below experiments, the comparisons of these two mechanisms are also provided.

4 EXPERIMENTS

4.1 Datasets

We conduct experiments on the E2E NLG dataset, where each instance is a pair of an MR and the corresponding human-written reference sentences. Since an MR could correspond to several references, we take all different MRs in the training set. For each MR, we randomly pick up one of its corresponding references. The statistics of data used are listed in Table 1. Compared with the total number of 42061 training instances in the original E2E NLG dataset, only a small fraction of instances are used in our experiment.

To validate the proposed masked hard coverage mechanism, we conduct experiments on TV and Laptop NLG datasets. Each instance is a pair of Dialogue Act (DA) and corresponding natural language realization. One example of the format of instance is: DA: *inform_no_match(type=television; pricerange=cheap; hdmiport=2; family=l2)*; Ref: *There are no cheap televisions in the l2 family with 2 hdmi ports.* In total, there are 14 different DAs in each dataset. To have DA information attached in the input, we take DA strings, such as *inform_no_match*, as the prompt input of the GPT-2 decoder. The remaining slot-value pairs and references are used as the input of encoder and decoder, respectively. The statistics of these datasets are documented in Table 1.

Table 1: The statistics of three NLG datasets.

	E2E	Laptop	TV
Training set	4862	7944	4221
development set	547	2649	1407
Test set	630	2649	1407

4.2 Evaluation Metrics

For the E2E NLG dataset, we compare with other models using several automatic n-gram overlap evaluation metrics, including BLEU, NIST, METEOR, ROUGE-L, and CIDEr[1]. For the Laptop and TV NLG datasets, we evaluate our generation results on metrics BLEU, ERR,[2] and BERTScore (Zhang et al., 2019). ERR is computed by the number of erroneous slots in generated utterances divided by the total number of slots in given MRs. Unlike BLEU that merely measures n-grams overlap, BERTScore computes token-wise similarity using contextual embeddings for candidate sentences and references, which correlates better with human judgments.

4.3 Results and Analyses

For the E2E NLG dataset, evaluation results are listed in Table 2. TGen (Dušek and Jurčíček, 2016) is the strong baseline system that no other single system could outperform on all evaluation metrics. The best previous results on different metrics are reported by Roberti et al. (2019), Dušek et al. (2018), Puzikov and Gurevych (2018), Zhang et al. (2018), Gong (2018), respectively. "Original" is the framework (Chen et al., 2020) on which our mechanism is built. "Original+covloss" is the original model with the soft coverage mechanism (See et al., 2017).

Compared with the original framework, our model's generation results have most of the evaluation results improved. Further comparison with the soft coverage mechanism demonstrates the validity and powerful copying constraint of our masked hard coverage mechanism. Considering the best previous results are separately achieved by different models, our models yield fairly well and balanced results regarding all metrics, and outperforms the baseline model TGen on metrics BLEU, METEOR, ROUGE-L, and CIDEr.

We list two groups of sentences produced by the original framework and our model in Table 4. For the first example, over-generation occurs in the utterances generated by the original framework and original framework with soft coverage mechanism (See et al., 2017); that is, the keywords "coffee shop" appear twice. In the second group, other systems produce utterances without covering the information "Japanese" food. In contrast, sentences produced by our model have all the information included. Apparently, over-generation and under-generation are eliminated in the examples.

To further validate the proposed coverage mechanism, experimental results on Laptop and TV NLG datasets are displayed in Table 3. HDC (Wen et al., 2015a) is a handcrafted generator capable of covering all the slots. SCLSTM (Wen et al., 2015b) is a baseline model, a statistical language generator based on a semantically controlled LSTM structure. "Ori+Cov" is the original framework (Chen et al., 2020) with soft coverage mechanism (See et al., 2017).

In terms of BLEU, SCLSTM surpasses all others; however, the comparable results of BERTScore among different models demonstrate that all of them generate sentences similar to the references. Regarding ERR, our model outperforms SCLSTM in the Laptop domain shows its potentiality for more ap-

[1]The evaluation script is provided by https://github.com/tuetschek/e2e-metrics.

[2]The tool is from https://github.com/shawnwun/RNNLG.

Table 2: Evaluation results on the test set of the E2E NLG task. The best previous results are from Roberti et al. (2019), Dušek et al. (2018), Puzikov and Gurevych (2018), Zhang et al. (2018), Gong (2018), respectively.

	BLEU↑	NIST↑	METEOR↑	ROUGE-L↑	CIDEr↑
TGen	0.6593	8.6094	0.4483	0.6850	2.2338
Best Previous Results	0.6705	8.6130	0.4529	0.7083	2.2721
Original (Chen et al., 2019)	0.6215	8.1044	0.4213	0.6421	1.6848
Original + Covloss (See et al., 2017)	0.6238	8.1101	0.4262	0.6424	1.7570
Ours	0.6610	8.4690	0.4486	0.6873	2.2426

Table 3: Evaluation results on Laptop and TV datasets. "Ori+Cov" is the original framework with soft coverage mechanism (See et al., 2017).

Laptop			
	BLEU↑	ERR↓	BERTScore↑
HDC	0.3761	0.00%	0.95
SCLSTM	0.5116	0.79%	0.94
Ori	0.4776	1.31%	0.95
Ori + Cov	0.4865	1.25%	0.95
Ours	0.4983	0.42%	0.95
TV			
	BLEU↑	ERR↓	BERTScore↑
HDC	0.3919	0.00%	0.95
SCLSTM	0.5265	2.31%	0.95
Ori	0.3818	14.57%	0.93
Ori + Cov	0.3852	14.11%	0.93
Ours	0.4183	4.34%	0.95

plications. We notice that the generated results are much better in the TV domain. The reason lies in that in few-shot learning, the model's performance is partially affected by training data size. The number of training instances in the TV domain is fewer than that in the Laptop domain, leading to worse performance.

To further analyze the results, we list two groups of sentences generated by different models in Table 5. In both groups, the slot "hasusbport" is not correctly covered in the sentences generated from our model, which is part of the reason why our model reach a higher ERR than others. In fact, the slot "hasusbport" is a ternary slot with three possible values: true, false, and dontcare, and they are preprocessed as "hasusbport", "hasnousbport" and "dontcare". These values might be inconsistent with their counterparts in the training references, for example, "hasusbport=true" may correspond to the phrase "does not have a usb port". Consequently, they are not labeled and included in the coverage mechanism when training, and are produced by the decoder when generating. Consequently, they are not labeled and included in the coverage mechanism when training and are produced by the decoder when generating. The powerful coverage constraint is not applied at the keywords corresponding to the slot hasusbport. To figure out how to cope with ternary slots more appropriately is one part of our future work.

5 CONCLUSION

This paper presents a masked hard coverage mechanism on a seq2seq-based pointer-generator network for the NLG task. The system employs a pre-trained language model GPT-2 as the decoder, and just small amounts of training data result in decent generation performance. Based on the original framework, we maintain a coverage vector to track the coverage of attribute-value pairs in input MRs. Moreover, we constrain the model to strictly cover all the given information through an additional loss term in the loss function. Experimental results show that our mechanism mitigates the over-generation and under-generation issues compared with the original framework and produces high-quality generation results and fewer slot errors.

REFERENCES

Agarwal, S., Dymetman, M., and Gaussier, E. (2018). Char2char generation with reranking for the e2e nlg challenge. *arXiv preprint arXiv:1811.05826*.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Chen, M., Lampouras, G., and Vlachos, A. (2018). Sheffield at e2e: structured prediction approaches to end-to-end language generation. *E2E NLG Challenge System Descriptions*.

Chen, Z., Eavani, H., Chen, W., Liu, Y., and Wang, W. Y. (2019). Few-shot nlg with pre-trained language model. *arXiv preprint arXiv:1904.09521*.

Chen, Z., Eavani, H., Chen, W., Liu, Y., and Wang, W. Y. (2020). Few-shot NLG with pre-trained language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 183–190, Online. Association for Computational Linguistics.

Dušek, O. and Jurčíček, F. (2016). Sequence-to-sequence

Table 4: Examples of utterances generated by the original framework (Chen et al., 2020), the original framework with coverage loss (See et al., 2017), and our work on E2E NLG dataset. Over-generation and under-generation occur in some sentences of these two groups, respectively.

MR: name:Zizzi, near:Burger King, **eatType:coffee shop**, customer_rating:high **Reference:** Near Burger King is a highly rated coffee shop called Zizzi. **Original:** Zizzi is a coffee shop near Burger King. It has a high customer rating **and is a coffee shop**. **Original + Covloss:** Zizzi is a coffee shop near Burger King. It has a high customer rating **and is a coffee shop**. **Ours:** Zizzi is a coffee shop near Burger King. It has a high customer rating.
MR: name:Green Man, eatType:pub, **food:Japanese**, area:riverside, near:Express by Holiday Inn **Reference:** Green Man is a pub serving **Japanese food**, it's located in the riverside area near Express by Holiday Inn. **Original:** Green Man is a pub near Express by Holiday Inn in the riverside area. **Original + Covloss:** Green Man is a pub near Express by Holiday Inn in the riverside area. **Ours:** A **Japanese** pub called Green Man is located in the riverside area near the Express by Holiday Inn.

Table 5: Examples of utterances generated by HDC, SCLSTM and our work in TV domain. Our model fails to correctly cover the slot hasusbport.

MR: inform_count(count=38;type=television;hdmiport=dontcare;**hasusbport=dontcare**) **Reference:** count 38 television number of hdmi port dontcare has_usb_port dontcare. **HDC:** there are 38 televisions if you don't care about the number of hdmi ports and if you don't care about whether it has usb port or not. **SCLSTM:** there are 38 television -s if you do not care about the number of hdmi port -s if you do not care about the number of **hdmi port -s**. **Ours:** if you don't care about the number of hdmi ports there are 38 televisions available with **no usb ports**.
MR: inform(name=morpheus 93;type=television;**hasusbport=false**;hdmiport=3;screensize=48 inch) **Reference:** the morpheus 93 television comes with 3 hdmi ports, a 48 inch screen size and no usb ports. **HDC:** morpheus 93 is a television which does not have any usb ports, has 3 hdmi ports, and has a 48 inch screen size. **SCLSTM:** the morpheus 93 television has a 48 inch screen, 3 hdmi port -s and no usb port -s. **Ours:** the 48 inch morpheus 93 television has 3 hdmi ports as well as **a usb port**.

generation for spoken dialogue via deep syntax trees and strings. *arXiv preprint arXiv:1606.05491*.

Dušek, O., Novikova, J., and Rieser, V. (2018). Findings of the e2e nlg challenge. *arXiv preprint arXiv:1810.01170*.

Elsahar, H., Gravier, C., and Laforest, F. (2018). Zero-shot question generation from knowledge graphs for unseen predicates and entity types. *arXiv preprint* *arXiv:1802.06842*.

Gong, H. (2018). Technical report for e2e nlg challenge. *E2E NLG Challenge System Descriptions*.

Gu, J., Lu, Z., Li, H., and Li, V. O. (2016). Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393*.

Juraska, J., Karagiannis, P., Bowden, K. K., and Walker, M. A. (2018). A deep ensemble model with slot alignment for sequence-to-sequence natural language generation. *arXiv preprint arXiv:1805.06553*.

Liu, T., Wang, K., Sha, L., Chang, B., and Sui, Z. (2018). Table-to-text generation by structure-aware seq2seq learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Mille, S. and Dasiopoulou, S. (2017). Forge at e2e 2017.

Nguyen, D. T. and Tran, T. (2018). Structure-based generation system for e2e nlg challenge. *E2E NLG Challenge System Descriptions*.

Puzikov, Y. and Gurevych, I. (2018). E2e nlg challenge: Neural models vs. templates. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 463–471.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Roberti, M., Bonetta, G., Cancelliere, R., and Gallinari, P. (2019). Copy mechanism and tailored training for character-based data-to-text generation. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 648–664. Springer.

See, A., Liu, P. J., and Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Smiley, C., Davoodi, E., Song, D., and Schilder, F. (2018). The e2e nlg challenge: A tale of two systems. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 472–477.

Tu, Z., Lu, Z., Liu, Y., Liu, X., and Li, H. (2016). Modeling coverage for neural machine translation. *arXiv preprint arXiv:1601.04811*.

Wen, T.-H., Gasic, M., Kim, D., Mrksic, N., Su, P.-H., Vandyke, D., and Young, S. (2015a). Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking. *arXiv preprint arXiv:1508.01755*.

Wen, T.-H., Gasic, M., Mrksic, N., Su, P.-H., Vandyke, D., and Young, S. (2015b). Semantically conditioned lstm-based natural language generation for spoken dialogue systems. *arXiv preprint arXiv:1508.01745*.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.

Zhang, B., Yang, J., Lin, Q., and Su, J. (2018). Attention regularized sequence-to-sequence learning for e2e nlg challenge. *E2E NLG Challenge System Descriptions*.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2019). Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.