A General Model for Neural Text Generation from Structured Data

Shuang Chen[*]
School of Computer Science and Technology
Harbin Institute of Technology, Harbin, China
hitercs@gmail.com

Abstract

In this paper, we introduce an end-to-end neural networks model for Data-to-Text which aims to generate natural language description from structured data. Our model is based on the sequence-to-sequence model where three additional components are added to tackle challenges in Data-to-Text. First, we apply structured data embedding with both field name and field value information to represent the table better. Second, we apply the copy mechanism to tackle the rare or unknown problem in Data-to-Text. Third, we apply a kind of coverage mechanism to discourage the model from generating repetitive contents. Experiments on E2E NLG shared task indicate that our system show promising results in some of the automatic metrics (eg. BLEU) and naturalness from human evaluation.

1 Introduction

Automatic text generation from structured data (Data-to-Text) is an import task in natural language generation. Data-to-Text is often formulated into two subproblems: *content selection* which decides what contents should be included in the text and *surface realization* which determines how to realize the text based on selected contents. Previous researches exploit this task in various domains, so several datasets are created in the meantime. For example, Liang et al. (2009) generates weather forecasts from meteorological records in **WeatherGov** dataset and Chen and Mooney (2008) generates sportscasting text from robot soccer events in **RoboCup** dataset. Recently, end-to-end (E2E) neural networks model based on encoder-decoder has been widely explored in this task (Wen et al., 2015; Mei et al., 2015; Dušek and Jurčíček, 2016) etc.

However, E2E neural networks model is limited to small dataset with generic patterns. Recently, Novikova et al. (2017) provided a new crowd-sourced data sets of 50k instances in the restaurant domain. Each instance is paired with a dialogue act-based meaning representation (MR) and 8.1 references on average. In contrast to previous Data-to-Text dataset, this dataset shows additional characteristics, such as open vocabulary, complex syntactic structures and diverse discourse phenomena. Moreover, multiple references in this dataset make the automatic evaluation metrics based on words overlap more convincible.

We propose a general neural networks model based on attentional sequence-to-sequence model (Sutskever et al., 2014; Bahdanau et al., 2014) with three additional components to tackle the challenges of this problem. Compared with neural machine translation and abstractive summarization, Data-to-Text using sequence-to-sequence model not only shares some common challenges but also has unique characteristics. First, we apply structured data embedding with both field name and field value representation (Sha et al., 2017) to encode the table information in contrast to representation of unstructured text in machine translation or summarization. Second, compared with machine translation and summarization, rare or unknown words problems in Data-to-Text is more serious because of occurrences of certain entity names in the input table. Although some research works (Dušek and Jurčíček, 2016) try to solve this problem by pre-processing and post-processing. That is, they anonymized certain fields with an abstract placeholder(eg. ”name”, ”near”). However, we declare that this simple approach will loss certain information related with the anonymized fields, and be infeasible when there is no clear correspondence between source and target. So we apply the copy mechanism (See et al., 2017) to tackle

[*] Part of this work was conducted during internship at Microsoft Research Asia

the rare and unknown words efficiently. Finally, we found that sequence-to-sequence model tends to generate repetitive contents especially for long sentence. So we apply a kind of coverage mechanism (See et al., 2017) to discourage the model from generating repetitive contents by remembering what has been attended so far.

To sum up, our system is an attention based sequence-to-sequence model with three additional components: structured data embedding, copy mechanism and coverage mechanism. We conduct experiments on the E2E NLG shared task. Experiment results (Dušek et al., 2018) indicate that our model achieve state of art performance with respect to some automatic evaluation metrics (rank 6 of 60 systems in BLEU) and human evaluation on naturalness (rank in the second cluster among primary systems). However, our primary system perform poorly in human evaluation on quality. Next, we are prepared to organize this article in the following order. Firstly, we will introduce our system components one by one, specifically in this order: attention based sequence-to-sequence model, structured data embedding, copy mechanism and coverage mechanism. Secondly, we will introduce the experiments part. Thirdly, we will give a conclusion and put forward some insights which we found during this challenge.

2 Task Definition

We formulate the Data-to-Text task as generating text S conditioned on table T. The table T consists of several field-value records. In dialogue, the set of records is called dialogue act-based meaning representation (MR), while the fields and corresponding values are called as slots and values.

3 Our Approach

Figure 1 gives an overview of our approach, the model is based on encoder-decoder framework. First, the table is flatted into a sequence of tokens in the form of field value word and corresponding field name. And these sequence of tokens are fed into the encoder sequentially to learn the table representation. Second, the decoder learns to generate words based on the table representation. We augment the decoder with copy mechanism proposed by (See et al., 2017). Third, we also apply the coverage mechanism proposed by (See et al., 2017) to relieve the repetition problems.

3.1 Sequence-to-sequence attentional model

As shown in Figure 1, our model is built on the sequence-to-sequence attentional model (Bahdanau et al., 2014). The encoder is based on the bi-directional Gated Recurrent Unit (GRU) (Cho et al., 2014), while the decoder is a single direction GRU with attention mechanism. The role of encoder is to read the input sequence $x = (x_1, x_2, ..., x_n)$, where x_i is the i^{th} token representation, to build basic representation. Here we apply bidirectional GRU (BiGRU) as the recurrent unit, where GRU is defined as:

$$z_i = \sigma(\mathbf{W}_z[x_i, h_{i-1}]) \tag{1}$$

$$r_i = \sigma(\mathbf{W}_r[x_i, h_{i-1}]) \tag{2}$$

$$\tilde{h_i} = \tanh(\mathbf{W}_h[x_i, r_i \odot h_{i-1}]) \tag{3}$$

$$h_i = (1 - z_i) \odot h_{i-1} + z_i \odot \tilde{h_i} \tag{4}$$

where $\mathbf{W}_z$, $\mathbf{W}_r$ and $\mathbf{W}_h$ are weight matrices.

The BiGRU consists of a forward GRU and a backward GRU, where the forward GRU reads the sequence from left to right while the backward GRU reads them in reverse order. The final encoder states $(h_1, h_2, ..., h_n)$ is a concatenation of the forward hidden states $(\overrightarrow{h}_1, \overrightarrow{h}_2, ..., \overrightarrow{h}_n)$ and backward hidden states$(\overleftarrow{h}_1, \overleftarrow{h}_2, ..., \overleftarrow{h}_n)$:

$$\overrightarrow{h}_i = \mathrm{GRU}(x_i, \overrightarrow{h}_{i-1}) \tag{5}$$

$$\overleftarrow{h}_i = \mathrm{GRU}(x_i, \overleftarrow{h}_{i-1}) \tag{6}$$

$$h_i = [\overrightarrow{h}_i; \overleftarrow{h}_i] \tag{7}$$

The decoder is a single directional GRU with attention mechanism. Concretely, at decoding time step t, GRU update the hidden states s_t with the word embedding w_{t-1} of previously generated word and previous context vector c_{t-1}. The formula is defined as:

$$s_t = \mathrm{GRU}(w_{t-1}, c_{t-1}, s_{t-1}) \tag{8}$$

$$s_0 = \tanh(\mathbf{W}_d \overleftarrow{h}_1 + b) \tag{9}$$

where $\mathbf{W}_d$ is the weight matrix, b is the bias vector, and $\overleftarrow{h}_1$ is the last backward encoder hidden states.

The attention mechanism is defined as follows:

$$\alpha_{t,i} = \frac{exp(e_{t,i})}{\sum_{i=1}^{n} exp(e_{t,i})} \tag{10}$$

$$e_{t,i} = v_a^T \tanh(\mathbf{W}_a s_{t-1} + \mathbf{U}_a h_i) \tag{11}$$

$$c_t = \sum_{i=1}^{n} \alpha_{t,i} h_i \tag{12}$$

where $\mathbf{W}_a$ and $\mathbf{U}_a$ are learnable parameters.

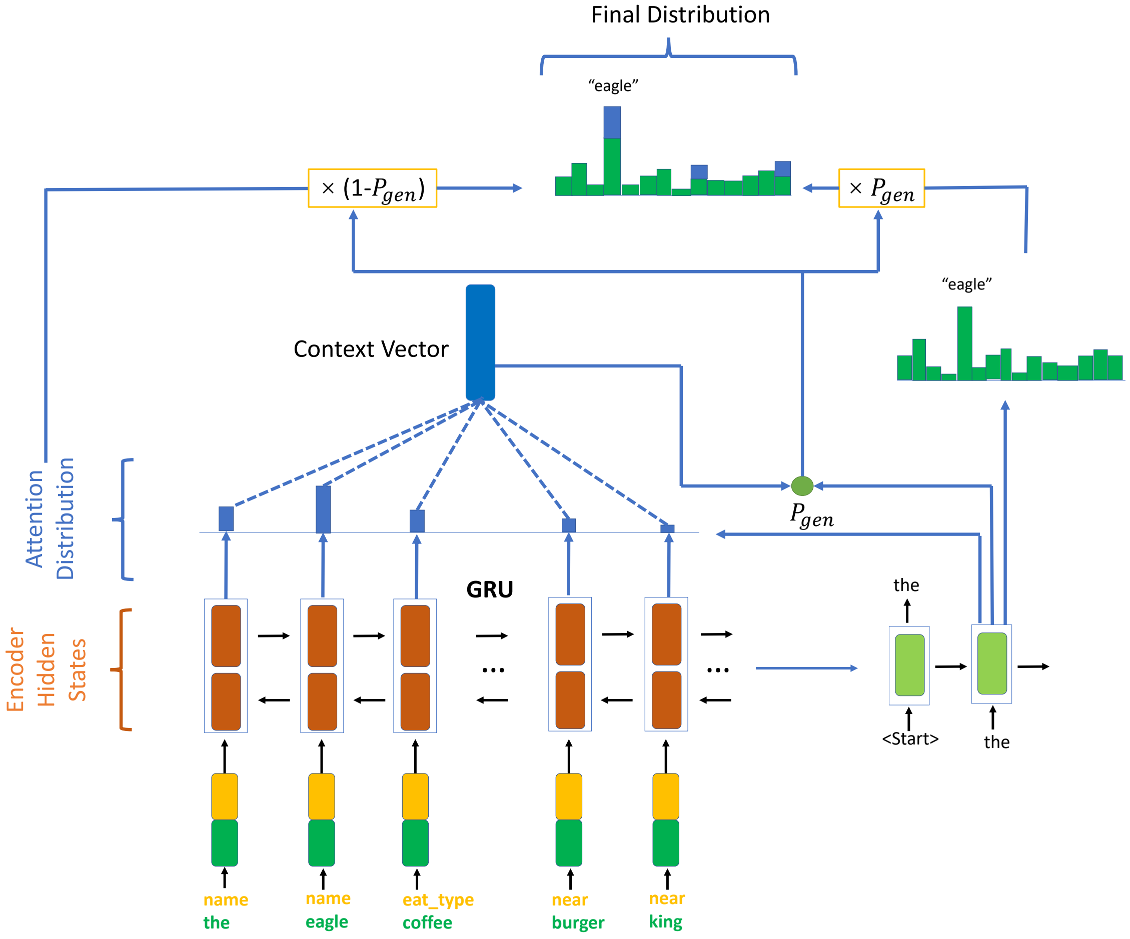

Figure 1: Overall model illustration

3.2 Structured data embedding

As shown in Table 1, we flatten the table into a sequence of tokens in the form of field value and corresponding field name. Each token in the sequence contains both field name and field value information. Here we apply the table representation methods which has been used in (Sha et al., 2017) to represent the table information better. Specially, let n be the number of field value words in the table T; let v_i and f_i be the embeddings of a value word and its corresponding field name, respectively $(i = 1...n)$. So the input to GRU x_i in Formula 1 is the concatenation of f_i and v_i which is defined as $x_i = [f_i; v_i]$.

3.3 Copy mechanism

To tackle the rare and unknown words in Data-to-Text, we augment the decoder with the copy mechanism from (See et al., 2017) where they call it the pointer-generator network. This mechanism allow the model to generate words from a fixed vocabulary while maintaining the ability to copy from source words. As shown in Figure 1, we introduce a gate called *generation probability* $p_{gen} \in [0, 1]$ at timestep t to control when to copy and when to generate. p_{gen} is calculated using the context vector c_t, the decoder states s_t and the decoder input w_{t-1}:

$$p_{gen} = \sigma(w_c^T c_t + w_s^T s_t + w_x^T w_{t-1} + b_{ptr}) \quad (13)$$

Field Name	Field Value
name	the
name	eagle
eat_type	coffee
eat_type	shop
...	...
near	berger
near	king

Table 1: Flattened table example

where vectors w_c, w_s, w_x and scalar b_{ptr} are learnable parameters. We define an *extended vocabulary* as the union of original fixed target vocabulary and all field value words appearing in the source table. Then the final word distribution $P(w)$ is a probability over the extend vocabulary. Since the gate p_{gen} is a latent variable, so we optimize the marginal probability distribution defined as:

$$P(w) = p_{gen}P_{vocab}(w) + (1 - p_{gen}) \sum_{i:w_i=w} a_i^t \quad (14)$$

So if w is an out-of-vocabulary word, then $P_{vocab}(w)$ is zero. Similarly, if w does not appear in source sentence, then $\sum_{i:w_i=w} a_i^t$ is zero.

3.4 Coverage mechanism

Sequence-to-sequence model tends to generate repetitive contents especially when generating long sentences. In order to relieve this problem, we also apply the coverage mechanism by See et al. (2017). The basic idea is to make the model to explicitly remember what has been attended so far, and pass this information to direct the attention model. Formally, we define a *coverage vector* acc^t, which is the sum of attention distributions until previous timesteps $t-1$:

$$acc^t = \sum_{t'=0}^{t-1} a^{t'} \quad (15)$$

Then the coverage vector is used as extra input to the attention calculation, modifying equation 11 to:

$$e_{t,i} = v_a^T \tanh(\mathbf{W}_a s_{t-1} + \mathbf{U}_a h_i + w_c acc_i^t) \quad (16)$$

where w_c is a learnable parameter vector of same length as v_a.

4 Experiments

4.1 Dataset

We conduct experiments on E2E NLG dataset (Novikova et al., 2017). The baseline model converts the original dataset into format used by TGen (Dušek and Jurčíček, 2016). In order to tackle the data sparsity problem, they delexicalized the *name* and *near* slots. We refer to this dataset as *Abstract Dataset*. Since our model consists of copy mechanism, we can process the data in its original form. So we precess another E2E dataset with its original form (keeping the words in ”near” and ”name” field instead of using anonymized placeholder) which we refer to as *Non-Abstract Dataset*. The E2E NLG dataset contains 42063 training samples and 4672 development samples. The test set contains 630 samples with only input. Basic statistics about this data set is shown in Table 2.

Data Set	Abstract	Non-Abstract
Source vocab size	35	112
Target vocab size	2392	2442
Number of fields	8	8

Table 2: Statistics of the training dataset

Data Set	Abstract	Non-Abstract
Source vocab size	37	114
Target vocab size	921	984
Max source #words	15	18
Max target #words	40	40

Table 3: Model parameters settings

4.2 Implementation Details

Model Parameters: As shown in Table 3, we set the vocabulary size according to the word frequency in the training data. Specially, we cover all the source words in the source vocabulary since the minimal frequency of source word is 2929 in *Abstract Dataset* and 348 in *Non-Abstract Dataset*. Similarly, we keep all the target words with frequency more than 5 times in the target vocabulary. Please note that *eos* and *unk* are included in all the vocabularies. Meantime, we set the maximum length of source words as 15 and 18 respectively. In addition, we set the source word embedding size, target word embedding size and field name word embedding size as 512 in both datasets. The dropout rate is set 0.5.

Model Training: We initialize model parameter randomly using a Gaussian distribution with Xavier scheme (Glorot and Bengio, 2010). During training, we use stochastic gradient descent (SGD) with batch size 100. The initial learning rate $\alpha = 0.5$. We validate the model performance (BLEU) on the development set for every 2000 batches. If the BLEU score drops for six consecutive tests on the development set, we halve the learning rate. We also apply gradient clipping (Pascanu et al., 2013) with range $[-1, 1]$ during training to avoid the exploding gradient problem.

Models	BLEU	NIST	METEOR	ROUGE-L	CIDEr
TGen	0.6925	8.4781	**0.4703**	0.7257	2.3987
Our System(Abstract, Greedy)	0.7185	**8.6359**	0.4579	0.7192	2.2662
Our System(Abstract, Beam-1)	0.6783	8.1401	0.4428	0.7275	2.0787
Our System(Non-Abstract, Greedy)	**0.7245**	8.5613	0.4596	**0.7374**	**2.4181**
Our System(Non-Abstract, Beam-1)*	0.6823	7.9823	0.4310	0.7247	2.0416

Table 4: Automatic evaluation results on dev set (*: Primary system)

Models	BLEU	NIST	METEOR	ROUGE-L	CIDEr
TGen	0.6593	**8.6094**	**0.4483**	0.6850	**2.2338**
Our System(Abstract, Greedy)	**0.6635**	8.3977	0.4312	**0.6909**	2.0788
Our System(Abstract, Beam-1)	0.5854	5.4691	0.3977	0.6747	1.6391
Our System(Non-Abstract, Beam-1)*	0.5859	5.4383	0.3836	0.6714	1.5790

Table 5: Automatic evaluation results on test set. (*: Primary system)

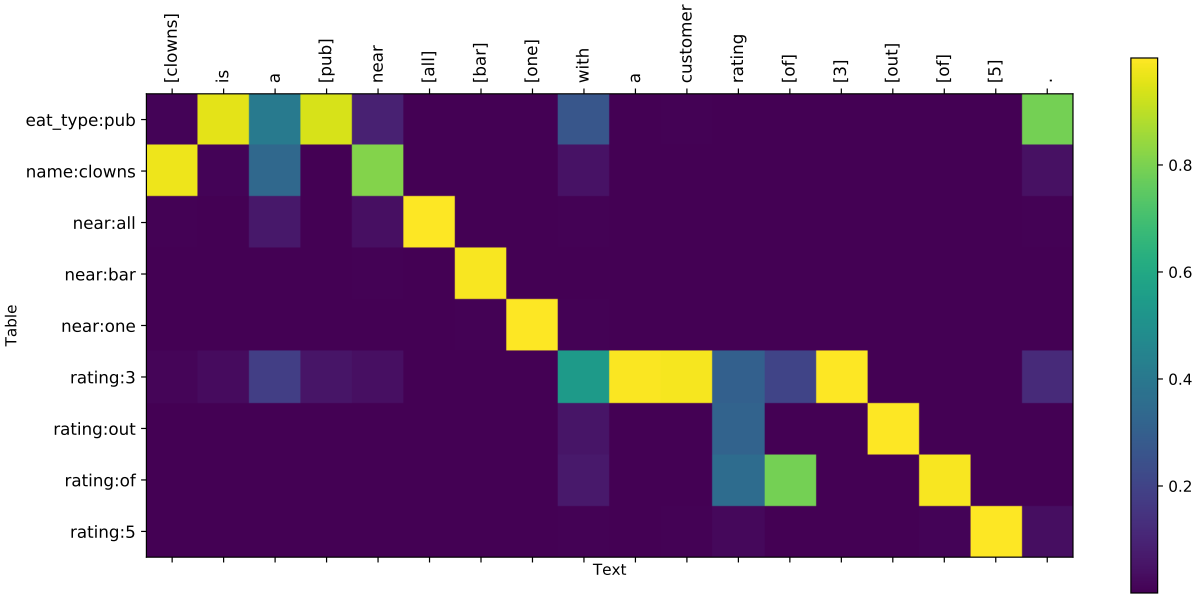

Figure 2: Generated example from our primary system

At test time, we use the both greedy search and beam search algorithms to generate texts.

4.3 Main Results

The E2E NLG Challenge uses both automatic evaluation metrics and human evaluation to assess the system performance. As for automatic evaluation, they apply some standard metrics, BLEU (Papineni et al., 2002), NIST (Doddington, 2002), METEOR (Denkowski and Lavie, 2014), ROUGE-L (Lin, 2004) and CIDEr (Vedantam et al., 2015) which have been used for evaluation in similar tasks. As for human evaluation, they obtain the human ratings from crowd workers.

We compare our model with the baseline model TGen (Dušek and Jurčíček, 2016). In addition, we also compare our model trained with the *Abstract Dataset* and *Non-Abstract Dataset*. As for *Non-Abstract Dataset*, we will perform lexicalization of the outputs, that it, filling the placeholder *near* and *name* with corresponding field value. We use the standard evaluation script[1] provided by E2E NLG Challenge to evaluate our systems. Meanwhile, we also compare the performance of decoding using greedy search algorithm and beam search algorithm.

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

Table 4 and 5 list performance of the baseline model and our systems with respect to different automatic evaluation metrics. In the development set, our system trained with *Non-Abstract Dataset* using greedy search outperforms the baseline model in BLEU, NIST, ROUGE-L and CIDEr but loses in METEOR. We set the beam size as 8, and select the top one of the final beam as result which we called **Beam-1**. We also found that greedy search often performs better than beam search with respect to automatic evaluation metrics. In the test set, our system trained with *Abstract Dataset* outperforms the baseline model in BLEU and ROUGE-L while loses in NIST, METEOR and CIDEr. We also can find that there is a huge margin between performance on development set and test set. It shows that generalization is hard in this specific task. We also conduct human evaluation on the generate results to examine the overall quality on the test set. In particular, we found that the quality of generated result from (Non-Abstract, Greedy) model perform quite poorly, that's why we didn't submit it although it performs quite good in the development set w.r.t automatic metrics.

4.4 Determination on Primary System

Determination on primary system is difficult, because we found that there is no strong correlation between automatic evaluation metrics and human evaluation. In order to determine the primary system fairly, we randomly sample 10 generated results per system and invite human to judge the quality of them. The quality we consider here includes fluency, semantic correspondence to the input and naturalness etc. Due to limit time and resource, we didn't conduct quantitative analysis here and just one person involved into the human evaluation which may bring bias. Finally, we select our system trained with Non-Abstract data with beam search as the primary system. According to the human evaluation results on the test set from E2E NLG Challenge, our primary system ranks in the second cluster with respect to **TrueSkill** on naturalness while performs poorly on the quality evaluation.

4.5 Generated Examples

In order to visualize our model, we show the attention scores between the source and target. As shown in Figure 2, the attention scores are represented by the color map and column-wisely normalized. The generated word bracketed with "[]" has generation probability p_{gen} (defined in Formula 13) less than 0.5 indicating that it is more likely to be copied from the source table. From what we can see, our model can learn the semantic correspondence between the table and generated text.

5 Conclusion

In this technical report, we present a general model for Data2Text. This model is built on the attentional sequence-to-sequence model with three additional components: structured data embedding, copy mechanism and coverage mechanism. Our model achieves the state-of-art results on the E2E NLG Challenge. For further work, we would like to consider the characteristic of this dataset, and build specific improvement on it. Moreover, we would like to consider the reason why different metrics result in different rankings and try to find a better automatic evaluation method.

Acknowledgments

We would like to thank Jing Liu, Xiaocheng Feng for helpful discussion. Meanwhile, we would like to thank the E2E Organising Committee for hosting such a meaningful contest and providing helpful feedback.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* .

David L Chen and Raymond J Mooney. 2008. Learning to sportscast: a test of grounded language acquisition. In *Proceedings of the 25th international conference on Machine learning*. ACM, pages 128–135.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* .

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram cooccurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*. Morgan Kaufmann Publishers Inc., pages 138–145.

Ondřej Dušek and Filip Jurčíček. 2016. Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings. *arXiv preprint arXiv:1606.05491* .

Ondrej Dušek, Jekaterina Novikova, and Verena Rieser. 2018. Findings of the E2E NLG challenge. In *(in prep.)*.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. pages 249–256.

Percy Liang, Michael I Jordan, and Dan Klein. 2009. Learning semantic correspondences with less supervision. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*. Association for Computational Linguistics, pages 91–99.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*. Barcelona, Spain, volume 8.

Hongyuan Mei, Mohit Bansal, and Matthew R Walter. 2015. What to talk about and how? selective generation using lstms with coarse-to-fine alignment. *arXiv preprint arXiv:1509.00838* .

Jekaterina Novikova, Ondrej Dušek, and Verena Rieser. 2017. The E2E dataset: New challenges for end-to-end generation. In *Proceedings of the 18th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Saarbrücken, Germany. ArXiv:1706.09254. https://arxiv.org/abs/1706.09254.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, pages 311–318.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*. pages 1310–1318.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368* .

Lei Sha, Lili Mou, Tianyu Liu, Pascal Poupart, Sujian Li, Baobao Chang, and Zhifang Sui. 2017. Order-planning neural text generation from structured data. *arXiv preprint arXiv:1709.00155* .

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. pages 3104–3112.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. pages 4566–4575.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. *arXiv preprint arXiv:1508.01745* .