How Do Seq2Seq Models Perform on End-to-End Data-to-Text Generation?

Anonymous ACL submission

Abstract

With the rapid development of deep learning, Seq2Seq paradigm has become prevalent for end-to-end data-to-text generation, and the BLEU scores have been increasing in recent years. However, it is widely recognized that there is still a gap between the quality of the texts generated by models and the texts written by human. In order to better understand the ability of Seq2Seq models, evaluate their performance and analyze the results, we choose to use Multidimensional Quality Metric (MQM) to evaluate several representative Seq2Seq models on end-to-end data-to-text generation. We annotate the outputs of five models on four datasets with eight error types and find that 1) copy mechanism is helpful for the improvement in Omission and Inaccuracy Extrinsic errors but it increases other types of errors such as Addition; 2) pre-training techniques are highly effective, and pre-training strategy and model size are very significant; 3) the structure of the dataset also influences the model’s performance greatly; 4) some specific types of errors are generally challenging for seq2seq models.

1 Introduction

Data-to-text generation is a task of automatically producing text from non-linguistic input (Gatt and Krahmer, 2018). The input can be in various forms such as databases of records, spreadsheets, knowledge bases, simulations of physical systems.

Traditional methods for data-to-text generation (Kukich, 1983; Mei et al., 2015) implement a pipeline of modules including content planning, sentence planning and surface realization. Recent neural generation systems (Lebret et al., 2016; Wiseman et al., 2017a) are trained in an end-to-end fashion using the very successful encoder-decoder architecture (Bahdanau et al., 2014) as their backbone. Ferreira et al. (2019) introduce a systematic and comprehensive comparison between pipeline and end-to-end architectures for this task and conclude that the pipeline models can generate better texts and generalize better to unseen inputs than end-to-end models.

However, with the rapid development of the Seq2Seq models especially pre-trained models, more and more end-to-end architectures based on Seq2Seq paradigm get state-of-the-art results on data-to-text benchmarks nowadays. Although BLEU score (Papineni et al., 2002), which is based on precision, has been improved dramatically on standard data-to-text benchmarks such as WebNLG (Gardent et al., 2017), ToTTo (Parikh et al., 2020) and RotoWire (Wiseman et al., 2017b) over the recent years, it is commonly accepted that, compared with human evaluation, BLEU score cannot evaluate the models very well. It is too coarse-grained to reflect the different dimensions of the models’ performance and not always consistent with human judgment (Novikova et al., 2017a; Reiter, 2018; Sulem et al., 2018). Moreover, existing human evaluations on data-to-text generation are usually limited in size of samples, numbers of datasets and models, or dimensions of evaluation.

In this study, we aim to conduct a thorough and reliable manual evaluation on Seq2Seq-based end-to-end data-to-text generation based on multiple datasets and evaluation dimensions. We want to know the pros and cons of different Seq2Seq models on this task, and the factors influencing the generation performance. Particularly, following MQM (Mariana, 2014), similar to the job on summarization evaluation (Huang et al., 2020), we use 8 metrics on the Accuracy and Fluency aspects to count errors, respectively. Therefore, compared with existing manual evaluation reports, it is more informative and objective.

Using this method, we manually evaluate several representative models, including Transformer (Vaswani et al., 2017), Transformer with Pointer Generator (See et al., 2017), T5(small&base) (Raf-
fe
de

1. Pre-training: Pre-training is powerful and effective which highly increases the ability of the Seq2Seq paradigm on the data-to-text task.

2. Size: The size of the model makes difference to the results. Particularly, T5-base achieves the best scores on both automatic and human evaluations.

3. Essential Techniques: The copy mechanism can make noticeable improvements for the basic Seq2Seq model, decreasing word-level errors such as Omission and Inaccuracy Extrinsic. But it also introduces more Addition errors sightly.

4. Dataset Structure: The structure of the dataset also influences the model’s understanding of the sequence greatly. Content-controlled generation is still a little hard for the Seq2Seq models.

5. Error Type: The most common mistakes of Seq2Seq models on data-to-text task are Omission, Inaccuracy Intrinsic and Inaccuracy Extrinsic, indicating the direction we need to improve the effectiveness of the model. On the other hand, models perform well in fluency.

The code and annotated data will be released to the community.

2 Related Work

Data-to-Text Generation Traditional methods for data-to-text generation (Kukich, 1983; Mei et al., 2015) implement a pipeline of modules including content planning, sentence planning and surface realization. Recent neural generation systems (Lebret et al., 2016; Wiseman et al., 2017a) are trained in an end-to-end fashion using the very successful encoder-decoder architecture (Bahdanau et al., 2014) as their backbone. Many Seq2Seq models have demonstrated their effectiveness on data-to-text tasks. Since we want to make a general comparison on Seq2Seq models, we will focus on this method. Moreover, with the development of pre-training methods, more and more work (Kale, 2020; Wang et al., 2021; Kale and Rastogi, 2020) began to introduce pre-training model for data-to-text generation.

There is some work evaluating and analyzing the data-to-text generation task. Perez-Beltrachini and Gardent (2017) propose a methodology to analyze the data-to-text benchmarks and apply their method to WikiBio, RNNLG (Wen et al., 2016) and IMAGEDESC (Novikova and Rieser, 2016) datasets. Ferreira et al. (2019) introduce a systematic comparison between pipeline and end-to-end architectures for neural data-to-text generation. Thomson and Reiter (2020) propose a methodology for human to evaluate the accuracy of the generated texts.

Sequence-to-Sequence Seq2Seq paradigm is a general and flexible paradigm that is typically implemented by an encoder-decoder framework. Sutskever et al. (2014) discuss sequence to sequence learning with neural networks. Furthermore, there are some representative architectures that have been proposed such as recurrent neural network (Zaremba et al., 2014) and Transformer (Vaswani et al., 2017). Seq2Seq paradigm can naturally handle any task, as long as their input and output can be represented as sequences. Therefore, there have been many attempts to apply Seq2Seq to different tasks. More recently, pre-trained models based on Seq2Seq paradigm (Lewis et al., 2019; Raffel et al., 2019) have proved their power on lots of tasks (McCann et al., 2018; Yan et al., 2021). There has been much work analyzing Seq2Seq models which is always task-specific and based on automatic or human evaluation. For example, Huang et al. (2020) analyze the common models’ performance on summarization.

To our knowledge, there is few work done to comprehensively evaluate the performance of Seq2Seq models on data-to-text generation. And much work is based on automatic metrics such as ROUGE or BLEU which can be different from human evaluation as some work (Novikova et al., 2017a; Reiter, 2018; Sulem et al., 2018) shows. Therefore it is meaningful to manually evaluate representative Seq2Seq models on the data-to-text
We choose to explore and compare Transformer, Pointer Generator, BART and T5’s performance on data-to-text generation and explore the role of copy mechanism by comparing Transformer and Pointer Generator, the benefits brought by the pre-training technique by comparing Transformer with T5 and BART, the influence of the different pre-training methods by comparing BART and T5, the power of parameter size by comparing T5-base and T5-small.

**Transformer**  Transformer (Vaswani et al., 2017) is widely used in natural language processing and has shown its potential on many tasks. It uses self-attention and multi-head attention which let a model draw from the state at any preceding point along the sequence. The attention layer can access all previous states and weigh them according to a learned measure of relevancy, providing relevant information about far-away tokens. There are also some experiments with Transformer as the baseline model (Zhao et al., 2020) for data-to-text generation. Moreover, many improved models for data-to-text generation are also based on Transformer (Wang et al., 2020; Zhu et al., 2019). Therefore, it is worth and reasonable to explore the performance of Transformer on the data-to-text task.

**3.1 Models**

We conduct experiments using five representative Seq2Seq models on four commonly used data-to-text datasets and evaluate the generated texts accordingly. Note that we do not use models that are designed for specific data sets or data structures (Moryossef et al., 2019; Rebuffel et al., 2020; Puduppully and Lapata, 2021), but adopt models that allow inputs of different formats and structures, which brings convenience to comparison on different data sets. Besides, most specific models for data-to-text generation are actually based on these typical Seq2Seq models (Ferreira et al., 2019; Rebuffel et al., 2020), which also proves the rationality of our selection of these models.

**3.2 Datasets**

We use the datasets commonly used in data-to-text task in the experiments, including E2E, WebNLG, WikiBio and ToTTo. They have different forms and characteristics, which can give a comprehensive comparison of models. The summary of these data-to-text datasets is shown in Table 1.

**E2E** The input of E2E dataset (Novikova et al., 2017b) is the information about the restaurant, and
the output is its natural language description. It consists of more than 50K combinations and the average length of the output text is 8.1 words.

**WikiBio** WikiBio (Lebret et al., 2016) is a personal biography dataset containing more than 70K examples. The input is the infobox from Wikipedia, and the output is the first sentence of the biography. The average length of the output text is 26.1 words.

**WebNLG** The WebNLG challenge (Gardent et al., 2017) consists of mapping sets of RDF triples to text. The latest WebNLG dataset contains more than 40K data-text pairs. The average length of the output text is 22.3 words.

**ToTTo** ToTTo (Parikh et al., 2020) is an open-domain English table-to-text dataset with over 120,000 training examples that proposes a controlled generation task: given a Wikipedia table and a set of highlighted table cells, produce a one-sentence description.

### 4 Evaluation Method

We first evaluate models’ performance using automatic metric BLEU (Papineni et al., 2002), and the BLEU scores are comparable to the mainstream research. Then, we use human evaluation similar to PolyTope (Huang et al., 2020) to further analyze and evaluate the performance of the models on different datasets.

BLEU is a precision-based metric for evaluating the quality of generated text and it is widely used by work on data-to-text generation.

Multidimensional Quality Metric (MQM) (Mari-ana, 2014) is a framework for describing and defining custom translation quality metrics. It defines flexible issue types and a method to generate quality scores. Based on MQM, Huang et al. (2020) introduce an error-oriented fine-grained human evaluation method PolyTope. It defines five issue types about accuracy, three issue types about fluency and three error severity rules. After annotating every generated sentence with these error types and severity, we finally calculate an overall score to evaluate the model’s performance.

### 4.1 Issue Type

According to the MQM principle, we define error types in two aspects: accuracy and fluency. Errors related to accuracy mean the generated text is not faithful to the original data or does not reflect the critical information totally from the original data. This type consists of five sub-types:

- **Addition** The generated text contains unnecessary and irrelevant fragments from the source data.
- **Omission** The key point does not exist in the output.
- **Inaccuracy Intrinsic** Terms or concepts appearing in the original data are distorted in the output.
- **Inaccuracy Extrinsic** The generated text shows the content which does not exist in the source data.
- **Positive-Negative Aspect** The generated text is positive, whereas the source data represents a negative statement and vice versa.

Fluency aspect evaluates the linguistic quality of the generated text, which is a primary natural language requirement. It consists of three sub-types:

- **Duplication** Unnecessarily repeat a word or longer part of the text.
- **Word Form** Problems related to the form of words, including consistency, part of speech, tense and so on.
- **Word Order** Problems about the order of words in outputs.

One example output with errors on WebNLG dataset is shown in Table 2.

### 4.2 Severity

Severity describes how severe a particular error is. There are three levels: Minor, Major and Critical. Each specific error in the sentence will be allocated a severity. It is decided by the annotator and will be considered as a weight to score the quality of the annotated sentence automatically.
### 4.4 Human Annotation

After training and testing, we hire five annotators with satisfactory levels in reading from eight candidates. They are all highly educated enough to understand structured data and tables, and their English level is also very high to understand the text. Before formal annotation, we conduct detailed training to make them have a clear understanding of various errors and the severity of PolyTope framework. Examples used in training do not appear in the final annotation. In order to ensure objectivity and impartiality, they know nothing about the name, architecture, BLEU score of the model and dataset in the process of annotation.

During testing, annotators are asked to locate every error’s position, point out the type of the error, choose the severity of the error and explain the reason. We check their answers and score them. Through the overall performance in the test, we select the best five annotators and ensure all of them really understand our evaluation method and have the ability to do the annotation work.

For each dataset, we select 80 data-text pairs and input them into each model respectively. There are four datasets and five models, so we have 1600 texts to annotate. Each text is annotated by two annotators.

Note that we set the length of an Omission error to 1.
Table 3: Human evaluation scores of each model on each dataset (higher means better).

<table>
<thead>
<tr>
<th>Model</th>
<th>E2E</th>
<th>WikiBio</th>
<th>WebNLG</th>
<th>ToTTo</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transformer</td>
<td>76.88</td>
<td>81.31</td>
<td>76.32</td>
<td>45.41</td>
<td>69.98</td>
</tr>
<tr>
<td>Pointer-GEN</td>
<td>86.97</td>
<td>82.98</td>
<td>78.76</td>
<td>54.57</td>
<td>75.82</td>
</tr>
<tr>
<td>T5-small</td>
<td>86.04</td>
<td>86.28</td>
<td>93.92</td>
<td>85.44</td>
<td>87.92</td>
</tr>
<tr>
<td>T5-base</td>
<td>96.36</td>
<td>91.38</td>
<td>94.10</td>
<td>88.59</td>
<td>92.61</td>
</tr>
<tr>
<td>BART-base</td>
<td>91.55</td>
<td>86.37</td>
<td>93.43</td>
<td>90.71</td>
<td>90.52</td>
</tr>
<tr>
<td>Average</td>
<td>87.56</td>
<td>85.66</td>
<td>87.31</td>
<td>72.94</td>
<td></td>
</tr>
</tbody>
</table>

Table 4: BLEU Scores of each model on every dataset (higher means better).

<table>
<thead>
<tr>
<th>Model</th>
<th>E2E</th>
<th>WikiBio</th>
<th>WebNLG</th>
<th>ToTTo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transformer</td>
<td>56.74</td>
<td>43.39</td>
<td>27.95</td>
<td>33.49</td>
</tr>
<tr>
<td>Pointer-GEN</td>
<td>61.37</td>
<td>49.39</td>
<td>27.54</td>
<td>35.28</td>
</tr>
<tr>
<td>T5-small</td>
<td>62.88</td>
<td>49.45</td>
<td>55.66</td>
<td>45.35</td>
</tr>
<tr>
<td>T5-base</td>
<td>59.96</td>
<td>49.12</td>
<td>59.48</td>
<td>48.91</td>
</tr>
<tr>
<td>BART-base</td>
<td>62.66</td>
<td>53.25</td>
<td>52.84</td>
<td>48.22</td>
</tr>
</tbody>
</table>

5 Result Analysis

We evaluate the five models mentioned above on four datasets using the above metrics. The overall human evaluation score and BLEU score of each model on each dataset are shown in Table 3 and Table 4, respectively. The detailed error scores of different error types are shown in Table 5. We can compare the performance of the models to see the influence of the pre-training technique, the copy technique and the mode size. Comparing the results on different datasets using the same model, we can discover how the structured data input influences the performance of the Seq2Seq models. Moreover, we can also analyze the detailed error scores to find out the weakness and advantages of specific models.

5.1 Copy Mechanism

Through comparing the results of Pointer Generator and Transformer on all datasets, we can see that the copy mechanism has a noticeable effect on the improvement of the results. It improves the generation performance on all the datasets. Particularly, it reduces the Inaccuracy Intrinsic error score by about 3 or 4 points on three datasets (E2E, WebNLG and ToTTo), as shown in Table 5. It is easy to understand because using copy mechanism, the model can generate words from the vocabulary through the generator or copy content from the source through the pointer. Pointer Generator with copy mechanism reduces almost all types of errors compared with vanilla Transformer such as Duplication error. The reason may be that the copy mechanism can interpolate vocabulary level probability with copy probability, reducing reliance on previous outputs.

We can observe that the improvement of Pointer Generator over Transformer is the largest on ToTTo dataset. This may be related to ToTTo’s need to pay more attention to the highlighted part of the input sequence, which emphasizes controllability.

Nevertheless, it is interesting that Addition error is increased slightly compared with Transformer. The likely reason may be that the auto-regressive decoder tends to copy longer sequences from the source and it is hard to interrupt the copy action.

5.2 Pre-training

In Table 3, we can see almost all the pre-training models outperform the non-pre-training models by a large margin among all the datasets except E2E dataset which may be too simple to evaluate the ability of models. The reason why the pre-training models can achieve better scores may be that they have learned helpful knowledge from lots of raw texts. And the pre-training method also helps the models become more powerful. BART and T5 are both pre-trained on tasks where spans of text are replaced by masked tokens. The models must learn to reconstruct the original document. According to the average scores of all the datasets, we can
Table 5: Error score of each error type for each model on 80 data-text pairs of every dataset. The results are scored based on manual evaluation and retained to two decimal places. Lower means better. Errors may be approximated to 0 because there are too few errors.

say that T5-base may be the best Seq2Seq model among our experimented models and BART-base is not far behind. And the models achieve the highest score on different datasets: BART-base is the best on ToTTo and T5-base is the best on the other datasets relatively.

5.4 Dataset
We can compare the difficulty level of the datasets by the average and the highest scores of all models. In Table 3, the ToTTo dataset has the lowest average score of 72.9. And the highest score on it achieved by BART-base is 90.7 which is also the lowest among all the datasets. ToTTo is made as a controlled generation task that given a Wikipedia table and a set of highlighted table cells, the model needs produce a one-sentence description of the highlighted part. It is much more complicated than other datasets describing all the given structured data. Maybe it is a bit confusing for models to find out what actually should be noticed, although the scores of the pre-training models are still very high. And the gap between pre-training models and non-pre-training models is the biggest on ToTTo among all datasets which indicates that the simple non-pre-training models can not handle the complex controlled generation very well. Of course the quantity of the data-text pairs and the length of the input and output sequence also influence the models’ performance.

5.3 Model Size
It is evident that the parameter quantity is the critical factor to the pre-trained model’s performance. BART-base has 139M parameters, T5-base has 220M parameters and T5-small has 60M parameters only. With the same architecture and same pre-training method, T5-base totally outperforms T5-small. Due to pre-training methods and other factors, T5-base and BART-base achieve the best results on different datasets. But on average, T5-base is the best. The relation between model size and the performance on different datasets is shown in Figure 1. The only exception mentioned above is ToTTo, where BART-base achieves the best results. One of the likely reasons is the pre-training strategy of BART which helps it have better denoising and reconstruction ability. Another reason will be mentioned below.
We can see that the overall trend of the BLEU Word Form and Word Order are very sporadic. This with the structured input. The errors of Duplication, Inaccuracy Intrinsic Error also can not be ignored which are 5.25 and 2.31, respectively. From the perspective of the pre-training model, it may be because they learn too much from the raw texts on pre-training stage and the knowledge lets them tend to generate inaccurate texts.

It is excited that all the models perform very well in terms of fluency. The errors of Duplication, Word Form and Word Order are very sporadic. This shows the Seq2Seq models can generate fluent text with the structured input.

### 5.5 Error Types

Table 6 shows the average error scores of each error type across all models and datasets. From Table 5 and Table 6, we can find that different types of errors have different effects on the performance of the models. We can find that Omission Error is the most frequent and severe error and its error score is almost up to 6. The likely reason is that the input sequence is too long, so it is hard to encode all its meaning. So the models tend to omit some information from the input. And Inaccuracy Intrinsic Error and Inaccuracy Extrinsic Error also can not be ignored which are 5.25 and 2.31, respectively.

From the perspective of the pre-training model, it may be because they learn too much from the raw texts on pre-training stage and the knowledge lets them tend to generate inaccurate texts.

<table>
<thead>
<tr>
<th>Addition</th>
<th>Duplication</th>
<th>Extrinsic</th>
<th>Intrinsic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Scores</td>
<td>1.48</td>
<td>0.63</td>
<td>2.30</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Omission</th>
<th>Positive-Negative Aspect</th>
<th>Word Form</th>
<th>Word Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Scores</td>
<td>5.97</td>
<td>0.17</td>
<td>0.23</td>
</tr>
</tbody>
</table>

Table 6: Average error score of each error type across all models and datasets (lower means better).

### 6 BLEU or Human Evaluation?

We can see that the overall trend of the BLEU score is consistent with human evaluation, which can basically reflect the overall performance of the model. Many conclusions we made above can also be proved by the BLEU score. For example, the biggest pre-training model T5-base achieves the highest BLEU score too among the selected models. Pointer Generator with copy mechanism still performs better than Transformer and ToTTo is still the most difficult dataset.

Although our primary goal is not to promote a human evaluation metric, our dataset with human annotations gives us a testbed to analyze the correlations and differences between automatic and human metrics. There have been a lot of discussions in the community about the unreliability of BLEU metric. Sulem et al. (2018) recommend not using BLEU on text simplification. They found that BLEU scores can neither reflect grammar nor the meaning of preservation. Novikova et al. (2017a)

5.5 Error Types

show that BLEU and some other commonly used indicators are not well consistent with human judgment when evaluating NLG tasks.

We compute the Pearson correlation coefficients between BLEU score and manual evaluation in terms of Accuracy and Fluency. We categorize the error types into accuracy and fluency aspects according to the definition in Section 4.1, and use Equation 2 to calculate Accuracy score and Fluency score respectively. The Pearson correlation coefficient between BLEU score and Accuracy is 0.61 and in Fluency aspect is 0.08. There is a huge gap between them and we can see that BLEU can evaluate Accuracy to a certain extent and it is poor at Fluency. Moreover, the BLEU metric is too coarse-grained to reveal the model’s specific problems, which enlighten us on how to improve the model. Our result is consistent with views of other work.

### 7 Conclusion

We empirically compared five representative Seq2Seq models on the data-to-text task using a fine-grained set of human evaluation metrics based on MQM. We aim to make a systematic and comprehensive evaluation and analysis on end-to-end Seq2Seq models for the data-to-text task. We analyze the effect of milestone techniques such as copy and pre-training, the influence of the dataset and model size and the models’ performance in terms of different types of errors. Our evaluation shows that pre-trained models can generate quite good texts. But there is still much room for improvement in this task. Furthermore, the improvement of specific errors is also worth exploring in the future.

### References


8


