Beyond Word for Word: Fact Guided Training for Neural Data-to-Document Generation

Feng Nie[1], Hailin Chen[2⋆], Jinpeng Wang[3], Rong Pan[1], and Chin-Yew Lin[3]

[1] Sun Yat-Sen University
fengniesysu@gmail.com, panr@sysu.edu.cn
[2] Nanyang Technological University
chen1039@e.ntu.edu.sg
[3] Microsoft Research Asia
{jinpwa, cyl}@microsoft.com

Abstract. Recent end-to-end encoder-decoder neural models for data-to-text generation can produce fluent and seemingly informative texts despite these models disregard the traditional content selection and surface realization architecture. However, texts generated by such neural models are often missing important facts and contradict the input data, particularly in generation of long texts. To address these issues, we propose a **F**act **G**uided **T**raining (FGT) model to improve both content selection and surface realization by leveraging an information extraction (IE) system. The IE system extracts facts mentioned in reference data and generates texts which provide fact-guided signals. First, a content selection loss is designed to penalize content deviation between generated texts and their references. Moreover, with the selection of proper content for generation, a consistency verification mechanism is designed to inspect fact discrepancy between generated texts and their corresponding input data. The consistency signal is non-differentiable and is optimized via reinforcement learning. Experimental results on a recent challenging dataset ROTOWIRE show our proposed model outperforms neural encoder-decoder models in both automatic and human evaluations.

Keywords: generation · information extraction · reinforcement learning.

1 Introduction

Data-to-text generation, a classic task of natural language generation, aims to generate descriptions that describe structured input data (e.g., tables) adequately and fluently [12, 19, 3, 1, 11]. Data-to-document generation is a more challenging setting in which a system generates multi-sentence summaries based on input data [26]. Traditionally, it is divided into content selection (i.e., *what to say*) and the surface realization (i.e., *how to say*) [19, 9]. Recent neural generation systems ignore the distinction of these two subtasks using a encoder-decoder model [23] with attention mechanism [2, 16, 8].

Although neural network models are capable of generating fluent text [26], they tend to generate irrelevant descriptions (e.g., missing essential contents in generated texts) and hallucinated content (e.g., text that contradicts the input structured data). As shown

⋆ Equal Contribution, work was done when the first and second author internships at Microsoft.

Input Data					
Name	PTS	AST	REB	FGM	FGA
E. Mudiay	25	9	6	10	17
Kyle Lowry	18	13	6	6	15
Generated: Kyle Lowry went 10 - for - 17 from the field to score 18 points while also adding 13 assists ...					
Reference: E. Mudiay had one of his best games of the season, as he tallied 25 points, six rebounds ...					

Table 1: A generated description from baseline model based on its paired input data. The underlined texts are words contradicted with input data and waved texts highlights the missing informative content in the reference data.

in Table 1, the generated text by a neural model does not mention the facts about one of the point leader "Emmanuel Mudiay" (e.g., "Emmanuel Mudiay tallied 25 points"). Such mistakes happen as most of current neural based methods is optimized word by word which ignores coverage of facts and implicitly model the content selection by solely relying on word level attention. Moreover, the neural methods also produce contradictory fact (e.g., "Kyle Lowry went 10 - for -17 from field"), as it is trained with maximum-likelihood (MLE) objective, which can only measure the generated texts with reference data word by word (i.e., on lexical level).

In this paper, we propose a **F**act **G**uided **T**raining (FGT) framework for data-to-text generation which measures content selection by penalizing content deviation between generated texts and references and measures consistency of generated texts by inspecting fact discrepancy between generated texts and input. In the scenario of data-to-text, the training data consists of loosely aligned structured input facts and unstructured description pairs, which do not have alignments between each token mentioned in the description to its corresponding input facts. To provide fact signals, a simple information extraction (IE) system is applied to collect the facts in the reference and the generated text [26]. E.g., (Emmanuel Mudiay, 25, PTS) is a fact in Figure 1.

To incorporate collected fact signals to improve both content selection and surface realization, we first design a simple yet effective content selection loss to penalize content deviation between generated texts and references, which encourages our model to learn the ability of selecting essential input facts with the fact signals. Moreover, with the selected facts, a consistency model is designed to inspect the contradictions between the generated text and its input data and between the generated text and its reference. Specifically, we apply the above IE system to extract facts from the generated texts, then compare the facts with its reference and its input data to produce reward signals. The non-differentiable consistency reward signals are incorporated into the training procedure via a reinforcement learning approach. In this way, the fact inconsistency can be treated as negative signals to guide a encoder-decoder network.

We evaluate the proposed method, FGT, on the ROTOWIRE dataset [26], which targets at generating multi-sentence game summaries. The experimental results show that FGT outperforms a encoder-decoder neural generation baseline in terms of BLEU and extractive metrics proposed by [26].

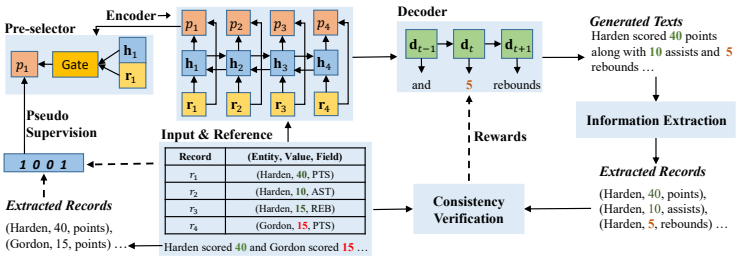

Fig. 1: Neural generation model with fact guided content selection and consistency verification .

2 Background

In this section, we briefly introduce the architecture of the attention-based sequence-to-sequence (Seq2Seq) [6, 2] model with copy mechanism [21], which is the basis of our proposed model.

The goal of data-to-text generation is to generate a natural language description $y = y_1, ..., y_T$ consists of T words for a given set of records $S = \{r_j\}_{j=1}^{T}$. Firstly, each input record r_j is encoded into a hidden vector $\mathbf{h}_j$ with $j \in \{1, ..., T\}$ using a bidirectional RNN. Then the decoder generates the description y by maximizing the conditional probability as:

$$P(y|S) = \prod_{t=1}^{T} P(y_t|y_{<t}, S) \tag{1}$$

where y_t is the t-th word in the description and T is the length of the description. The conditional probability $P(y_t|y_{<t}, S)$ is computed as:

$$P(y_t|y_{<t}, S) = \text{softmax}(f(\mathbf{d}_t, y_{t-1}, \mathbf{c}_t)) \tag{2}$$

where $f(.)$ is a non-linear function, $\mathbf{d}_t = LSTM(\mathbf{d}_{t-1}, y_{t-1}, c_{t-1})$ is the hidden state in the decoder at time step t, and $\mathbf{c}_t = \sum_{j=1}^{T} \alpha_{t,j}\mathbf{h}_j$ is the context vector at time step t, $\alpha_{t,j}$ is computed by the attention model [2]. We also adapt the conditional copy mechanism [10, 21] into the Seq2Seq models.

3 Our Approach

As shown in Figure 1, our model contains two parts, an encoder plugged with a pre-selector module, where a subset of the input records are selected for decoding, and an attention-equipped decoder. To ensure that generated texts describe the same set of records with its corresponding reference, we collect factual information by applying an information extraction (IE) system, where the information acts as a pseudo content selection supervision to guide the pre-selector to choose relevant input information for generation. Moreover, to avoid the contradictions between the generated texts and the input information, a consistency verification procedure is applied to inspect factual information overlap between the generated texts and its paired input table and the corresponding reference via a reinforcement learning approach.

3.1 Record Encoder

Given a set of input records $S = \{r_i\}_{i=1}^{K}$, each record r is a triple (r^e, r^f, r^v), where r^e, r^f and r^v refer to the entity (e.g. Harden), the field name (e.g. column PTS) and value (e.g. cell value 40), respectively. We map each record $r \in S$ into a vector $\mathbf{r}$ by concatenating the embedding of r^e, r^f and r^v, denoted as $\mathbf{r} = [\mathbf{e}^e, \mathbf{e}^f, \mathbf{e}^v]^\top$, where $\mathbf{e}^e$, $\mathbf{e}^f$, $\mathbf{e}^v$ are trainable word embeddings of r^e, r^f and r^v, similar to [27]. We feed a set of record vectors $\mathbf{r}_1, ..., \mathbf{r}_K$ to a bidirectional LSTM and yield the final record representations $\mathbf{h}_1, ..., \mathbf{h}_K$ as introduced in Section 2.

3.2 Information Extraction

To enable fact measurement for content selection and surface realization in data-to-text generation, we employ an information extraction (IE) system to extract relevant input information from the description.

We build a simple IE system based on input and description pairs similar to [26]. Given a generated text $\hat{y}_{1:T}$, we first extract all possible candidate entity e (team, player and city) and value r (number) pairs from the text, and then predict the field name r^f of each candidate pair. For the example in Figure 1, ("Harden", "40") is a possible (entity, value) pair in the generated texts, and its corresponding field is "PTS". In this way, the relation extraction is simplified to multi-class classification , formulated as follows:

$$p(r^f|e, v, x) \propto \mathbf{s}_x^\top [\mathbf{W}^{class}]_{r^f} \tag{3}$$

where x is the sentence which entity e and value v lie in, $\mathbf{s}_\mathrm{x}$ is the learned sentence representation, and $\mathbf{W}^{class}$ refers to classification embedding matrix, and $[\mathbf{W}^{\mathrm{class}}]_{\mathrm{r^f}}$ is the column vector that contains the embedding of class r^f. Note that $r^f = \epsilon$ indicates unrelated (entity, value) pair. Given an input and description pair (S, y), e extract a set of records $\hat{U} = \{\hat{u}_j\}_{j=0}^{|\hat{u}|}$ from the generated text using the trained IE system. For the records set mentioned in reference $U = \{u_i\}_{i=0}^{|U|}$, we use the pseudo label which is constructed for training the IE system, instead of extracting that from the reference.

3.3 Content Selection

Given a set of input records, one core step in data-to-text generation is to decide what to say by selecting a small subset of salient records that are relevant to the output description. Most of the neural methods rely on the attention mechanism to select input content by scanning the entire input records during decoding at each time step t, while the search space for attention mechanism is large. Following [16], it is reasonable to use a content selection model to first capture the prior p_j for each record r_j and re-weight the attention probability $\alpha_{t,j}$ to recalculate the context vector $\mathbf{c}_t$ as follows:

$$p_j = \sigma(\mathbf{q}^\top \tanh(\mathbf{P}[\mathbf{h}_j, \mathbf{r}_j]^\top)) \tag{4}$$

$$\alpha_{t,j} = \mathrm{softmax}(\mathbf{v}^\top \tanh(\mathbf{W}\mathbf{d}_{t-1} + \mathbf{H}\mathbf{h}_j)) \tag{5}$$

$$\beta_{t,j} = p_j \alpha_{t,j} / \sum_j p_j \alpha_{t,j} \tag{6}$$

$$\mathbf{c}_t = \beta_{t,j} \mathbf{h}_j \tag{7}$$

where $\mathbf{r}_j$ and $\mathbf{h}_j$ represent the record embedding and the hidden units in RNN layer for record r_j respectively, **P**, **H**, **W**, **q** and **v** are learned parameters. In this way, the attention mechanism is affected by the prior probability p_j, where a large p_j represents the current record is salient.

In data-to-text scenario, the given references are unstructured text where the alignments of each token to its corresponding input record is not provided. Learning the prior probability p_j automatically from such loosely aligned input and description pairs is difficult. To derive direct training signals for content selection, we collect an approximate supervision by taking the advantage of the IE systems. Specifically, an additional loss based on the records extracted from the reference (i.e., U) is constructed to guide the content selection:

$$L_{cs} = -\sum_{i} \Big(\mathbb{1}_{cs}(r_i) \log p_i + (1 - \mathbb{1}_{cs}(r_i)) \log \min(1 + \eta - p_i, 1) \Big) \tag{8}$$

where $\mathbb{1}_{cs}(r_i)$ is the indicator function which produces 1 when the input record r_i appears in U, otherwise 0. η is a hyper-parameter to control the tolerance on negative labels, as the pseudo label constructed for training the IE system may contains mistakes in which some records that are mentioned in the reference can not be extracted. We set η to 0.5 according to the validation set.

3.4 Consistency Verification

The approximate content selection supervision enforces the model to choose relevant input information for generation. However, a more critical problem for neural generation models is producing facts contradict its paired input table. We therefore propose a novel verification mechanism to inspect the discrepancy between generated texts and its paired input data to guide the training.

Specifically, we first collect the facts from the generated texts by using the IE system introduced above, and then examine the overlap with its paired input records and its reference. Since only a subset of words in the generated text are describing facts, we design two word-level rewards to encourage words that are consistent with the input table and penalize those containing mistakes.

Consistency Rewards To measure the consistency of the generated texts, we design two rewards based on the reference and the input data respectively. Note that the consistency is designed on the fact level, and we will make use of the record set $U = \{u_i\}_{i=0}^{|U|}$ and $\hat{U} = \{\hat{u}_j\}_{j=0}^{|\hat{u}|}$ which extracted from the reference $y_{1:N}$ and the generated text $\hat{y}_{1:T}$.

We define the first reward to check whether the records extracted from the generated text match those from the input data. Specifically, reward for each word $\hat{y}_t$ in $\hat{y}_{1:T}$:

$$R^S(\hat{y}_t, S) = \sum_{\hat{u}_i \in Sub(\hat{U}, \hat{y}_t)} \Big(\mathbb{1}_S(\hat{u}_i) - b_s \Big) \tag{9}$$

where $Sub(\hat{U}, \hat{y}_t)$ returns a subset of $\hat{U}$ in which the word $\hat{y}_t$ equals to one of the elements in each record, b_s is set to 0.5, and $\mathbb{1}$ is the indicator function defined as:

$$\mathbb{1}_S(\hat{u}_i) = \begin{cases} 1, & \text{if } \hat{u}_i \in S \\ 0, & \text{if } \hat{u}_i \notin S \end{cases}$$

Similarly, we define the second reward for each word $\hat{y}_t$ to inspect the consistency between the generated text $\hat{U}$ and its corresponding reference data U:

$$R^U(\hat{y}_t, S) = \sum_{\hat{u}_i \in Sub(\hat{U}, \hat{y}_t)} \Big(\mathbb{1}_U(\hat{u}_i) - b_u \Big) \tag{10}$$

where b_u is set to 0.5, and

$$\mathbb{1}_U(\hat{u}_i) = \begin{cases} 1, & \text{if } \hat{u}_i \in U \\ 0, & \text{if } \hat{u}_i \notin U \end{cases}$$

To integrate the consistency measurement from both input and reference data, the final *consistency reward* $R(\hat{y}_t|S)$ is calculated by combining these two rewards as follows:

$$R(\hat{y}_t, S) = \lambda_1 R^U(\hat{y}_t, S) + \lambda_2 R^S(\hat{y}_t, S) \tag{11}$$

where λ_1 and λ_2 are hyper parameters to control the scale for each reward. We set both λ_1 and λ_2 to 0.5 according to the validation set.

Policy Gradient Reinforce The consistency reward introduced above is non-differentiable for end-to-end training. One way to remedy this is to learn a policy that maximizes the consistency reward instead of minimizing the maximum-likelihood loss, which is made possible with reinforcement learning. We use the REINFORCE algorithm [25, 28] to learn a policy p_θ, where p_θ refers to the distribution produced by the encoder-decoder model introduced in Eq. 1. The training objective is formulated as follow:

$$J(\theta) = \mathbb{E}_{(\hat{y}_{1:T}) \sim p_\theta(.|S)} \, R(\hat{y}_{1:T}, S) \tag{12}$$

where $R(\hat{y}_{1:T}, S)$ is the reward function of the sequence of words $\hat{Y} = (\hat{y}_1, ..., \hat{y}_T)$ sampled from the policy. Unfortunately, computing the expectation term is prohibitive, since there is an infinite number of possible sequences. In practice, we approximate this expectation with a single sample from the policy distribution p_θ. The gradient of the J_{RL} is:

$$\nabla J_{RL} Z \approx \sum_{t=1}^{T} \nabla_\theta \log p_\theta(\hat{y}_t|\hat{y}_{1:t-1}, S)[R(\hat{y}_{1:T}, S) - b_t] \tag{13}$$

where b_t is a baseline estimator to reduce the variance, and defined as $b_t = \sum_{t=1}^{T} R(\hat{y}_{1:T}, S)/T$. Moreover, our proposed reward can only affect a subset of words related to the input data. Therefore, our word-level reward function can be formulated as $R(\hat{y}_{1:T}, S) =$

$\sum_{t=1}^{T} R_t(\hat{y_t}|\hat{y}_{1:t-1}, S)$. Therefore, we can have word level feedback as [24]:

$$\begin{aligned} \nabla J_{RL} &\approx \sum_{t=1}^{T} \nabla_\theta \log p_\theta(\hat{y_t}|\hat{y}_{1:t-1}, S)(Q_t - b_t) \\ \text{where } Q_t &= \sum_{k=t}^{T} \gamma^{k-t} R_t(\hat{y_k}|\hat{y}_{1:k-1}, S) \end{aligned} \tag{14}$$

with γ denoting a discount factor $\in [0, 1]$. The original REINFORCE algorithm starts learning with a random policy, which can make the model training for generation tasks with large vocabularies a challenge. We therefore conduct pre-training on our policy with the maximum likelihood (MLE) objective prior to REINFORCE training.

4 Experiments

4.1 Datasets & Evaluation

Data: We use ROTOWIRE dataset [26], which is a collection of articles summarizing NBA basketball games, paired with their corresponding box- and line-score tables. The average number of input records and article length is 628 and 337 respectively. It consists of 3,398, 727, and 728 summaries for training, validation and testing respectively.
Evaluation: For automatic evaluation metrics, we use BLEU-4 [17] and the extractive evaluation metrics proposed by [26] for evaluation. The extractive evaluation metrics are based on relationship classification techniques introduced in Section 3.2. Following [26], we evaluate our proposed method on these three criteria: a) Relation Generation (RG): precision (P%) and number (#) of unique records correctly reflected in the generated text; b) Content Selection (CS): precision (P%) and recall (R%) of unique records correctly reflected in the generated text that are also appear in its paired reference; c) Content Ordering (CO): normalized Damerau-Levenshtein Distance (DLD%) between the sequence records extracted from generated text G and reference text R. Among

	Dev						Test					
	RG		CS		CO	BLEU	RG		CS		CO	BLEU
	P%	#	F1%	P%/R%	DLD%		P%	#	F1%	P%/R%	DLD%	
`Ref`	95.98	16.93	100	100/100	100	100	96.11	17.31	100	100/100	100	100
`Template`	99.93	54.21	35.42	23.42/72.62	11.30	8.97	99.95	54.15	35.75	23.74/72.36	11.68	8.93
`Wiseman`	75.74	16.93	34.64	31.20/38.94	14.98	14.57	75.62	16.83	36.02	**32.80**/39.93	15.62	14.19
`Seq2Seq`	74.80	19.62	34.47	28.90/42.71	15.18	14.19	74.18	19.75	33.92	28.44/42.03	14.71	14.55
`PreSel`	77.15	17.97	35.22	31.10/40.62	15.59	14.40	77.03	18.45	34.65	30.51/40.10	15.68	14.27
`+ CS`	78.75	19.16	36.73	**31.83**/43.43	**15.70**	15.19	79.17	19.65	36.32	31.50/42.88	**16.41**	14.95
`+ CV`	78.33	19.59	36.53	31.21/44.05	15.39	15.49	77.46	19.62	35.67	30.53/42.90	15.28	14.94
`FGT`	**82.22**	**22.36**	**37.90**	31.30/**48.04**	15.46	**15.62**	**82.99**	**23.17**	**38.09**	31.19/**48.90**	15.58	**15.73**

Table 2: Results of different methods on ROTOWIRE dataset, where the best performance of neural based methods on each metric is in **bold**.

these three criteria, the RG metric directly evaluates the data fidelity of the system and

thus is the most crucial evaluation metric, and we argue that CO metric does not really reflect the quality of generation, as there are different ways to describe the same information of a game.

4.2 Experimental Setup

In the main experiments, we compare our model with : (a) `Template`: a problem-specific, template-based generator similar to [26][4], (b) `Wiseman`: an encoder-decoder neural method with conditional copy mechanism (c) `Seq2Seq`: Seq2Seq model with pointer network copy mechanism introduced in the background section. It is one of the state-of-the-art neural systems, (d) `PreSel`: `Seq2Seq` method plus the content selection introduced in Eq.4-7. For ablation study, we provide the results of (e) `PreSel+CS`: `PreSel` when adding our proposed content selection loss for training and (f) `PreSel+CV`: `PreSel` with consistency verification. All the experiments use beam size of 5 in decoding. **Training:** For MLE training, we use the SGD optimizer with starting learning rate as 1. For REINFORCE training, we continue from MLE training with the same optimizer and learning rate. The dimension of trainable word embeddings and hidden units in LSTMs are all set to 600 and both encoder and decoder share the same word embedding. As the length of generated text is more than 300 words on average, we apply the truncated back propagation with window size 100. For REINFORCE training, we set the sample size to 1, γ to 0 according to the validation set[5], and limit the consistency reward for each word to be within the range [-1, 1].

4.3 Main Results

Experimental results with comparisons to the previous work on this dataset are shown in Table 2[6]. We apply MLE training on our baseline model and achieve comparable results on ROTOWIRE dataset w.r.t. the previous work [26]. The differences between our method and [26] is that we adopt a LSTM for the encoder, while [26] uses a table encoder similar to [27]. Template based method performs poorly than all neural based method in terms of BLEU score, but it performs quite well on the extractive metrics, as input data is directly feed into placeholders of template by rules, which provides the upper-bound for how domain knowledge could help content selection and consistency for generation. For neural based methods, the `PreSel` shows improvement over `Seq2Seq` method in the precision of RG and CS metrics, as well as achieves comparable performance in terms of BLEU score, which indicates the importance of content selection for generation. Our proposed method `FGT` which incorporates fact-guided content selection loss and the consistency verification into training outperforms `PreSel` in terms of both BLEU and extractive metrics. Notably, for the recall of CS metric which directly measures the content overlap with reference texts, we observe

[4] A template example, where the players and scores are emitted in the sentence. `<player>` scored `<pts>` points (`<fgm>-<fga>` FG, `<tpm>-<tpa>` 3PT, `<ftm>-<fta>` FT)

[5] We do not apply dropout in RL training

[6] Wiseman17 have recently updated the dataset to fix some mistakes. We cannot directly use the results which is reported in their paper and rerun the author's code.

				RG		CS		CO	BLEU
				Acc%	#	F1%	P%/R%	DLD	
Linear	Precision	Recall	PreSel+CV	78.12	12.64	35.27	37.34/33.41	16.72	12.03
Classifier	0.460	0.322	FGT	75.76	11.76	34.52	37.43/32.03	17.01	12.04
CNN+LSTM	Precision	Recall	PreSel+CV	79.17	19.65	36.32	31.50/42.88	16.41	14.95
Classifier	0.947	0.753	FGT	82.99	23.17	38.09	31.19/48.90	15.58	15.73

Table 3: Performance of our framework over different RE models in ROTOWIRE test dataset.

6.87% improvements over PreSel, and the result shows that our proposed method is able to generate more relevant information which is also selected by the reference. Moreover, the precision and average number of relations in RG metric increases 5.96% and 4.72 respectively which proves that FGT produces less contradicted facts than baseline methods. The result confirms that our proposed method is helpful for both content selection and fidelity of generation when incorporating fact-level training objectives.

4.4 Ablations

To investigate the effect of content selection training objective and the consistency verification individually, we report the results of ablations of our model in Table 2 by disabling some components in our proposed method. The results show that incorporating content selection loss is helpful for the recall of CS metric. This suggests that injecting an additional content selection loss for content selection enables the model to generate more input records which also selected by the reference. Interestingly, we also observe the improvement on RG, which explains the necessity of content selection to reduce the influence of irrelevant information for neural generation models. Similarly, our proposed method yields performance boost in precision of RG and CS metric by incorporating consistency verification, as the fidelity of generation is guaranteed by using consistency constraints to guide the training. The results illustrate the effectiveness of using fact-guided training objectives for data-to-text generation.

4.5 Effect of Information Extraction

As the IE system is the core component to improve both content selection and surface realization from fact aspect, we investigate the affect brought by different IE models. Table 3 shows the performance on two relation classifiers with different methods to learn the sentence representation $\mathbf{s}_x$ introduced in Eq. 3. The Linear Classifier refers to use a simple linear layer with average pooling method to learn the sentence vector, and the CNN+LSTM Classifier refers to the ensemble method of using both convolutional neural network and LSTM to represent the sentence. As shown in in Table 3, Linear Classifier has only 46% precision and 32% recall on extraction. This means that it extracts a large portion of incorrect records from the generated texts and misleads the rewards, as performances decrease compared to baseline method PreSel and PreSel+CS. In contrast, a relatively strong relation classifier CNN+LSTM Classifier is helpful for consistency verification and achieves much better performance over Linear Classifier. The results also suggest that potential improvements for our framework are available if better relation classifiers are incorporated.

Seq2Seq: ... The Raptors were led by DeMar DeRozan, who went 12-for-25 from the field and 0-for - 6 from the three-point line to score a game-high of 30 points, ... Kyle Lowry also had a strong showing as well. He went 10-for-17 from the field and 0-for-6 from the three-point line to score 18 points, while also adding 13 assists ...
FGT: ... DeMar DeRozan led the way for Toronto, as he tallied 30 points, five rebounds and four assists on 12-of-25 shooting. Kyle Lowry was second on the team, with his 18 points, six rebounds and 13 assists on 6-of-15 shooting. Jonas Valanciunas was the only other starter in double figures ... Emmanuel Mudiay finished second on the team, totaling 25 points, six rebounds and nine assists.
Ref: ... Emmanuel Mudiay had one of his best games of the season though, as he tallied 25 points, six rebounds and nine assists. Wilson Chandler continues to dominate off the bench, as his 25 points and 10 rebounds add to his averages of 24 points and 9 rebounds over his last three games...

Table 4: Example output from `Seq2Seq`, `FGT` and Reference. Text in red is inconsistent with input, text in blue are consistent with input.

	#Supp.	#Contra.	Error ratio(%)
`Seq2Seq`	3.65	1.15	23.96
`FGT`	5.02	1.22	19.55

Table 5: Average number of supported and contradicted words describing input records in the generated text per sentence.

4.6 Qualitative Analysis

Case Study: We provide an example of generated text by our model, together with the generation result by baseline model `Seq2Seq` and its corresponding reference text in Table 4[7]. It is clear to see that our proposed method `FGT` is able to generate more facts that are also mentioned in the reference, such as one leading player "Emmanuel Mudiay". Moreover, our proposed method is less likely to produce mistakes describing the player scoring points and the number of shooting goals when compared to the baseline method `Seq2Seq` (e.g. a large portion of content describing "Kyle Lowry" is wrong). However, we notice that our method produces mistakes when requiring calculation among the input data (e.g. "Jonas was the only starter in double figures"). Such information cannot be extracted by IE systems, therefore `FGT` made mistakes describing them. The results also suggest the limitation of the simple IE system.

Human Evaluation: We also conduct human evaluation to examine the words describing input records in generated texts. We randomly sampled 50 games from the test set and randomly select one sentence from each game. Each sentence is rated by three annotators who are familiar with NBA games. They are first required to identify text spans which contain facts from generated texts and then check whether the text spans are consistent or contradicted with the input data. Results in Table 5 show that our proposed method generate more facts than vanilla sequence-to-sequence model and make less mistakes in generation (i.e. the error ratio decrease absolute 4.38% compared to the baseline method).

[7] The complete game summary is relatively long, we presents a part of summary for brevity.

5 Related Work

Data-to-text generation is a task of natural language generation (NLG) [9]. Previous research has focused on individual content selection [12, 19, 7] and surface realization [22]. For neural based methods, mei2016 uses a neural encoder-decoder approach with a coarse-to-fine aligner for end-to-end training. Some have focused on conditional language generation based on tables [27], short biographies generation from Wikipedia tables [13, 5, 15]. duvsek16 use a neural encoder-decoder for generation and applies a DA reranker to choose the most appropriate sentence. Chisholm17 uses a table-text and text-table auto-encoder framework for Wikitext generation. Wiseman17 generate game summaries and use the information extraction model as evaluation. Perez-Beltrachini18 model content selection explicitly using multi-instance learning to improve the generation quality. liunian propose a two stage method that first uses neural network to generate template and then rewrite the content for generation. Most recently, ratish18 propose an end-to-end system that incorporate content selection and content planning in generation. The difference of their work and ours lies in that our methods considers fact-level training objectives to improve the content selection and fidelity during generation, while their work explicitly models the content selection and planning using specific neural modules.

Our work is also related to use specialized rewards to improve specific tasks such as dialogue [14], image captioning [20], simplification [29], summarization [18] and recipe generation [4]. Our work first considers the consistency reward in generation by making use of information extraction system.

6 Conclusion and Future Work

In this paper, we propose a new training framework to improve both content selection and surface realization from fact aspect by using information extraction (IE) based methods. After extracting fact-guided signals from reference data, we propose a loss function to directly optimize content selection with these signals. Moreover, to avoid factual contradictions between the generated texts and its pairing input data, a novel IE based verification module is incorporated into the training framework. Experimental results show that our method outperforms the state-of-the-arts neural encoder-decoder models in both automatic and human evaluations. In the future, we will generalize our model to other domains.

References

1. Angeli, G., Liang, P., Klein, D.: A simple domain-independent probabilistic approach to generation. In: EMNLP (2010)
2. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. ICLR (2015)
3. Barzilay, R., Lapata, M.: Collective content selection for concept-to-text generation. In: EMNLP (2005)
4. Bosselut, A., Çelikyilmaz, A., He, X., Gao, J., Huang, P., Choi, Y.: Discourse-aware neural rewards for coherent text generation. In: NAACL (2018)

5. Chisholm, A., Radford, W., Hachey, B.: Learning to generate one-sentence biographies from wikidata. CoRR **abs/1702.06235** (2017)
6. Cho, K., van Merrienboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: ACL (2014)
7. Duboué, P.A., McKeown, K.R.: Statistical acquisition of content selection rules for natural language generation. In: EMNLP (2003)
8. Dušek, O., Jurcicek, F.: Sequence-to-sequence generation for spoken dialogue via deep syntax trees and strings. In: ACL (2016)
9. Gatt, A., Krahmer, E.: Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. J. Artif. Intell. Res. (2018)
10. Gu, J., Lu, Z., Li, H., Li, V.O.K.: Incorporating copying mechanism in sequence-to-sequence learning. In: ACL (2016)
11. Kim, J., Mooney, R.J.: Generative alignment and semantic parsing for learning from ambiguous supervision. In: COLING (2010)
12. Kukich, K.: Design of a knowledge-based report generator. In: ACL. pp. 145–150 (1983)
13. Lebret, R., Grangier, D., Auli, M.: Neural text generation from structured data with application to the biography domain. In: EMNLP. pp. 1203–1213 (2016)
14. Li, J., Monroe, W., Ritter, A., Jurafsky, D., Galley, M., Gao, J.: Deep reinforcement learning for dialogue generation. In: EMNLP (2016)
15. Liu, T., Wang, K., Sha, L., Chang, B., Sui, Z.: Table-to-text generation by structure-aware seq2seq learning. In: AAAI (2018)
16. Mei, H., Bansal, M., Walter, M.R.: What to talk about and how? selective generation using lstms with coarse-to-fine alignment. In: NAACL (2016)
17. Papineni, K., Roukos, S., Ward, T., Zhu, W.: Bleu: a method for automatic evaluation of machine translation. In: ACL. pp. 311–318 (2002)
18. Paulus, R., Xiong, C., Socher, R.: A deep reinforced model for abstractive summarization. CoRR **abs/1705.04304** (2017)
19. Reiter, E., Dale, R.: Building applied natural language generation systems. Natural Language Engineering (1997)
20. Ren, Z., Wang, X., Zhang, N., Lv, X., Li, L.: Deep reinforcement learning-based image captioning with embedding reward. In: CVPR (2017)
21. See, A., Liu, P.J., Manning, C.D.: Get to the point: Summarization with pointer-generator networks. In: ACL. pp. 1073–1083. Association for Computational Linguistics (July 2017)
22. Soricut, R., Marcu, D.: Stochastic language generation using widl-expressions and its application in machine translation and summarization. In: ACL (2006)
23. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: NIPS (2014)
24. Sutton, R.S., Mcallester, D., Singh, S., Mansour, Y.: Policy gradient methods for reinforcement learning with function approximation. In: NIPS. pp. 1057–1063. MIT Press (2000)
25. Williams, R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine Learning (1992)
26. Wiseman, S., Shieber, S.M., Rush, A.M.: Challenges in data-to-document generation. In: EMNLP (2017)
27. Yang, Z., Blunsom, P., Dyer, C., Ling, W.: Reference-aware language models. In: EMNLP. pp. 1850–1859 (2017)
28. Zaremba, W., Sutskever, I.: Reinforcement learning neural turing machines. CoRR **abs/1505.00521** (2015)
29. Zhang, X., Lapata, M.: Sentence simplification with deep reinforcement learning. In: EMNLP (2017)