Enhancing Neural Data-To-Text Generation Models with External Background Knowledge

Shuang Chen¹, Jinpeng Wang², Xiaocheng Feng¹, Feng Jiang¹,³, Bing Qin¹, Chin-Yew Lin²
¹ Harbin Institute of Technology, Harbin, China ² Microsoft Research Asia ³ Peng Cheng Laboratory
hitercs@gmail.com, {jinpwa, cyl}@microsoft.com, {xcfeng, qinb}@ir.hit.edu.cn, fjiang@hit.edu.cn

Abstract
Recent neural models for data-to-text generation rely on massive parallel pairs of data and text to learn the writing knowledge. They often assume that writing knowledge can be acquired from the training data alone. However, when people are writing, they not only rely on the data but also consider related knowledge. In this paper, we enhance neural data-to-text models with external knowledge in a simple but effective way to improve the fidelity of generated text. Besides relying on parallel data and text as in previous work, our model attends to relevant external knowledge, encoded as a temporary memory, and combines this knowledge with the context representation of data before generating words. This allows the model to infer relevant facts which are not explicitly stated in the data table from an external knowledge source. Experimental results on twenty-one Wikipedia infobox-to-text datasets show our model, KBAtt, consistently improves a state-of-the-art model on most of the datasets. In addition, to quantify when and why external knowledge is effective, we design a metric, KBGain, which shows a strong correlation with the observed performance boost. This result demonstrates the relevance of external knowledge and sparseness of original data are the main factors affecting system performance.

1 Introduction
Automatic text generation from structured data (data-to-text) is a classic task in natural language generation which aims to automatically generate fluent, truthful and informative texts based on structured data (Kukich, 1983; Holmes-Higgins, 1994; Reiter and Dale, 1997). 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 generate the text based on selected contents. Traditionally, these two subproblems have been tackled separately. In recent years, neural generation models, especially the encoder-decoder model, solve these two subproblems jointly and have achieved remarkable successes in several benchmarks (Mei et al., 2016; Lebret et al., 2016; Wiseman et al., 2017; Dušek et al., 2018; Nie et al., 2018).

Such end-to-end data-to-text models rely on massive parallel pairs of data and text to learn the writing knowledge. They often assume that all writing knowledge can be learned from the training data. However, when people are writing, they will not only rely on the data contents themselves but also consider related knowledge, which is neglected by previous methods. For example, as shown in Fig. 1, an infobox about a person called Nacer Hammami is paired with its corresponding biography description from the Wikipedia. However, the information in the infobox is not enough to cover all the facts mentioned in the description. To generate this description from the in-
foinbox, we need to expand information based on external background knowledge from its related entities. For example, in the description: 1) “is an Algerian football player” indicates the nationality of Nacer Hammami which is not explicitly stated in the infobox. However, the place of birth, Guelma, of Nacer Hammami is given, therefore the nationality can be inferred from the knowledge that Guelma is a place in Algeria. 2) “playing for MC El Eulma in the Algerian Ligue Professionnelle 1” describes the fact that MC El Eulma is a club in the Algerian Ligue Professionnelle 1 which is also not explicitly stated in the infobox which can be expanded from external knowledge base.

One may argue that neural models can learn such knowledge when enough parallel co-occurrence pairs such as (Guelma, Algerian) and (MC El Eulma, Algerian Ligue Professionnelle 1) are available. However even in such case, neural models still tend to make mistakes for sparse co-occurrence pairs as we will show in the experiments section.

In this paper we enhance neural-network-based data-to-text generation models with external knowledge in a simple but effective way. Besides learning the association between data and text from parallel data-text pairs as in previous work, our model attends to relevant external knowledge, encoded as a temporary memory, and combines this knowledge with the context representation of data before generating words. Specifically, both infobox and background knowledge facts are encoded and a dual-attention mechanism is proposed to guide the decoder to generate text.

To verify the effectiveness of our proposed model, Knowledge Base enhanced Attention-based sequence-to-sequence network (KBAtt), we conduct experiments on multiple Wikipedia infobox-to-text datasets including WikiBio (Lebret et al., 2016) and 20 new datasets\(^1\). Our experiment results show that KBAtt consistently improves a state-of-the-art neural data-to-text model to achieve higher performances on most of the datasets. To quantify when and why external knowledge is effective, we design a metric which could be used to enhance the performance of neural data-to-text models.

The contributions of our work can be summarized as follows:

- We propose a simple yet effective model, KBAtt, to integrate external knowledge using a dual-attention mechanism.
- We design a metric, KBGain, to quantify when and why external knowledge is effective.
- We contribute twenty infobox-to-text datasets from a variety of domains.

2 The Proposed Model

Our model takes a data table \(D\) (e.g., a Wikipedia infobox) and a relevant external knowledge base (KB) containing a set of facts \(F\) as input and generates a natural language text \(y = y_1, \ldots, y_T\) consisting of \(T\) words. To augment the infobox with external knowledge, we preserve the Wikipedia internal hyperlink information in the field values of infobox, and track these hyperlinks to get their corresponding entities from Wikidata\(^2\) where we retrieve only one-hop facts. The backbone of our model is an attention based sequence-to-sequence model (Bahdanau et al., 2014) equipped with copy mechanism (See et al., 2017). As shown in Fig. 2, the model consists of four main components: a table encoder, a KB encoder, the dual attention mechanism and a decoder. We describe each component in the following sections.

2.1 Table Encoder

In Fig. 2, the input data table \(D\) consists of several field name and field value pairs. We follow (Sha et al., 2017; Liu et al., 2017) to tokenize the field values and transform the input table into a flattened sequence \(\{(n_i, v_i)\}_{i=1}^N\), where each element is a token \(v_i\) from a field value paired with its corresponding field name \(n_i\). To encode the flattened table, we map each \((n_i, v_i)\) to vector \(x_i = [e_n^i; e_v^i]\), where \(e_n^i\) and \(e_v^i\) are trainable

\(^1\)Available at https://github.com/hitercs/WikiInfo2Text

\(^2\)We adopt Wikidata (dumps version: 20150831) as the external knowledge base. Although we extend the infobox with external knowledge by using the Wikipedia hyperlink, we can also apply entity linking to link input data to the knowledge base in practice.

3014
Figure 2: A diagram of the knowledge base enhanced neural data-to-text generation model. First, we transform the table into a flattened sequence, extract entities mentioned in the field value of the infobox and link them to Wikidata where we can retrieve relevant facts. Then, the table contents and external knowledge base facts are carefully encoded. Finally, a single layer GRU decoder with a dual attention mechanism decides which part of information should be used for generation.

word embeddings of \( n_j \) and \( v_i \), and \([; ;] \) is the concatenation of vectors. Then each \( x_i \) is encoded into a hidden vector \( h_i \) using a bi-directional GRU (Cho et al., 2014).

2.2 Knowledge Base Encoder

As shown in Fig. 2, we extract entities mentioned in the field value of infobox and link them to Wikidata. Then we can retrieve relevant facts whose subject is the linked entity from Wikidata. These facts contain important background knowledge related to the infobox which is helpful for generation. The KB fact set is denoted by \( F = \{ (n_j, s_j, r_j, o_j) \} \) \(|F| \), where \( s_j, r_j, o_j \) is the subject, relation and object of the fact respectively, and field name \( n_j \) indicates the current fact is linked by the field value of \( n_j \). In order to integrate such KB facts into the neural model, we apply Multi-Layer Perceptron (MLP) to encode each fact into its representation:

\[
f_j = \tanh( W_f [e^{n_j}; e^{s_j}; e^{r_j}; e^{o_j}] + b_f) \tag{1}
\]

where \( W_f \) and \( b_f \) are trainable weights and bias, while \( e^{n_j}, e^{s_j}, e^{r_j} \) and \( e^{o_j} \) is the embedding of field name \( n_j \), subject \( s_j \), relation \( r_j \) and object \( o_j \) respectively. To accommodate generation steps where no information from the external knowledge is needed, such as generating name field which is already stated in the table, we apply a simple strategy by padding a none fact in the knowledge base.

2.3 Dual Attention Mechanism

After encoding the table and background knowledge base facts, we apply a RNN-based decoder to generate words conditioned on both table information and background knowledge fact information. In general, given a decoder hidden state \( d_t \) at timestep \( t \), we apply dual attention mechanism including table attention and KB attention to determine which parts that it should pay attention to. Next we will introduce table attention and KB attention briefly.

2.3.1 Table Attention

The table attention is based on similarity between decoder hidden state \( d_t \) and table contents representation \( \{h_i\}_{i=1}^N \). Specially we apply attention mechanism from (Bahdanau et al., 2014) to calculate the table context representation \( c_{t,table} \).

\[
e_{t,i} = v_a^\top \tanh(W_a d_t + U_a h_i) \tag{2}
\]

\[
\alpha_{t,i} = \frac{\exp(e_{t,i})}{\sum_{i=1}^N \exp(e_{t,i})} \tag{3}
\]

\[
c_{t,table} = \sum_{i=1}^N \alpha_{t,i} h_i \tag{4}
\]

where \( W_a, U_a \) and \( v_a \) are trainable parameters. \( \alpha_{t,i} \) is the table attention weight.

2.3.2 KB Attention

Besides utilizing table information, the words generated may contain facts which are not directly mentioned in the table but could be inferred from the background knowledge \( F \). In order to integrate such knowledge into the decoder, we apply KB attention over \( \{f_j\}_{j=1}^{|F|} \). Similar to the table attention, we can get KB context representation \( c_{t}^{kb} \).
2.4 Decoder

The decoder is a single layer GRU equipped with copy mechanism. As for the generation mode, given a decoder hidden state $d_t$, the decoder attends both table and knowledge base using mechanism described above, and get table context representation $c^{table}_t$ and KB context representation $c^{kb}_t$. So the context representation at time step $t$ is given by:

$$c_t = [c^{table}_t; c^{kb}_t]$$ (5)

Then given table $D$ and knowledge base facts set $F$, the probability of word $y_t$ generated from a fixed vocabulary at time step $t$ is defined as follows:

$$P_{vocab} = \text{softmax}(f(d_t, y_{t-1}, c_t))$$ (6)

where $f(\cdot)$ is a non-linear function applying on the decoder hidden state $d_t$, previous word embedding $y_{t-1}$ and the current context vector $c_t$.

To tackle the rare and unknown words problem, we adopt the copy mechanism (See et al., 2017). Specifically, a gate $p_{gen} \in [0, 1]$ is introduced to switch between copy mode and generation mode. The generation probability $p_{gen}$ is defined as:

$$p_{gen} = \sigma(w^c_t c^{table}_t + w^d_t d_t + w^y_t y_{t-1} + b)$$ (7)

where $\sigma$ stands for the sigmoid function while vectors $w_c$, $w_d$, $w_y$ and scalar $b$ are trainable parameters. The joint probability for generating $y_t$ at time step $t$ is formulated as follows:

$$P(y_t|y_{<t}, D, F) = p_{gen} P_{vocab}(y_t)$$
$$+ (1 - p_{gen}) \sum_{i:t_i=y_t} \alpha_{t,i}$$ (8)

where $\alpha_{t,i}$ is the table context weight defined in Equation 3.

2.5 Training

Given training dataset $\{(D^k, F^k, y^k)\}^S_{k=1}$ consisting of $S$ samples, our goal is to maximize the probability of target description $y = y_1, \ldots y_T$ given the input table $D$ and the background knowledge facts $F$. So the objective function is to minimize the negative log-likelihood:

$$J(\theta) = -\frac{1}{S} \sum_{k=1}^{S} \sum_{t=1}^{T_k} \log P(y^k_t|y^k_{<t}, D^k, F^k)$$ (9)

The objective function is fully differentiable, so the entire model can be trained end-to-end through backpropagation. Adam (Kingma and Ba, 2014) is used to optimize our model.

3 When and Why External Knowledge is Beneficial

In this section, we introduce KBGain to quantify when and why external knowledge (KB) is effective, and then show how KBGain correlates with the performance boost of KBAtt over Seq2Seq+Copy in 21 datasets in Section 4.4. Intuitively, incorporating an external KB should improve data-to-text generation performance. However, to pinpoint the effect of the additional knowledge is not trivial since we know that (a) not all external knowledge is relevant and (b) neural models may memorize certain inference patterns when parallel data is big enough. We assume (1) matching tokens between the external KB and the references indicate relevance of the tokens in the external KB; (2) high frequency of co-occurrence of matching tokens between the infobox and the references indicate good potential for neural models to learn generation patterns which leads to less effectiveness of an external KB, i.e. no data sparsity in the original data. To characterize these factors, we introduce KBGain.

KBGain measures the portion of learnable tokens in the references co-occurred with their corresponding external KB entries but filter out those tokens which could also co-occurred with the infobox. We say a token is learnable from one source (infobox or KB) when the co-occurrence frequency between them is higher than a threshold $\gamma$ which is the minimum size of co-occurrence above which learning will be effective. The top table in Fig. 3 summarizes learnable tokens in 4 possible cases. Among them, case C where the frequency of infobox and reference (Infobox-Ref) token co-occurrence does not exceed this threshold but its KB and reference (KB-Ref) token co-occurrence does not exceed this threshold but its KB and reference (KB-Ref) token co-occurrence does not exceed this threshold but its KB

---

Figure 3: An illustration of KBGain metric.

| Field Name | Field Value |
|------------|------------|
| Place of birth | Guelma |
| Current team | MC El Eulma |
| Subject | Current team |
| Relation | Place of birth |
| Object | Guelma |

---

3 Here, $\gamma$ is set 25 by tuning on the development datasets, please see Appendix A.3 for more details.
occurrence does, then this is where the KB will be effective. KBGain is defined as the average ratio between count of the tokens falling into category \( C \) and the length of their corresponding reference on the test set.

Specifically, for all tokens in the reference except stop words and punctuation, we select those matched on string with the object tokens of KB but not with the infobox. For example, the token \( \text{Algerian} \) in the reference is matched with the object of the last KB tuple. The KB-Ref token pair is simply acquired by string matching while Infobox-Ref token pair can be tracked by firstly finding the corresponding field based on the matched tuple, and then selecting the token in its field value with the highest pointwise mutual information (PMI).

### 4 Experiments

#### 4.1 Datasets

In the experiments, we adopt the WikiBio (Lebret et al., 2016) along with twenty new infobox-to-text datasets collected from Wikipedia. The full statistics of these datasets could be found in the Appendix A.2. Table 1 shows the statistics of two sample datasets. Each dataset consists of infoboxes as input data and the first sentences of their corresponding Wikipedia articles as references. For example, on datasets WikiBio and Album, we extract 5.9 and 5.5 entities from each table, and each entity has 19.0 and 7.2 extended facts tuples, on average. WikiBio and the other 20 datasets consist of 728,321 and 668,796 instances respectively, and they together cover 84.34% infoboxes of the English Wikipedia.

#### 4.2 Experiment Setup

We conduct experiments on the datasets as introduced in Section 4.1. Seq2Seq+Copy is the main baseline: a sequence-to-sequence (Seq2Seq) model equipped with copy mechanism from See et al. (2017) which is one of the state of art methods. We also compare our results with other published results using the WikiBio dataset. The model structure of our baseline model is most similar to Sha et al. (2017) by removing their specialized design on order planning which is not the focus of this paper. Since our aim is to verify the effectiveness of external knowledge for data-to-text task, we keep our baseline model as general as possible without other specialized design. Our primary model is KBAtt: a model which integrates the background knowledge into baseline model through a KB encoder and KB attention mechanism. We employ BLEU (Papineni et al., 2002) as the automatic evaluation metric. In addition to BLEU, we conduct human evaluation to assess the factual accuracy of generated sentences.

---

Table 1: Dataset statistics for two sample datasets.

| Dataset | WikiBio | Album |
|---------|---------|-------|
| Avg. # tokens per sentence | 26.1 | 21.0 |
| Avg. # tokens per table | 53.0 | 57.1 |
| Avg. # fields per table | 19.7 | 15.9 |
| Avg. # table tokens per sent. | 11.6 | 9.7 |
| Avg. # entities per table | 5.9 | 5.5 |
| Avg. # fact tuples per entity | 19.0 | 7.2 |
| Dataset size | 728,321 | 76,105 |

Table 2: BLEU-4 score with two generation models on 21 datasets.

| Dataset | Seq2Seq+Copy | KBAtt | ∆ |
|---------|--------------|-------|---|
| Single | 43.60 | 47.70 | 4.10 |
| Station | 54.30 | 56.77 | 2.47 |
| Australian_place | 43.73 | 45.73 | 2.00 |
| Album | 41.04 | 42.77 | 1.73 |
| NRHP | 48.97 | 50.43 | 1.46 |
| Airport | 45.17 | 46.61 | 1.44 |
| Book | 36.07 | 37.49 | 1.42 |
| Automobile | 18.64 | 19.95 | 1.31 |
| Building | 24.13 | 25.21 | 1.08 |
| UK_school | 33.64 | 34.72 | 1.08 |
| School | 37.30 | 38.33 | 1.03 |
| Football_club_season | 46.05 | 47.02 | 0.97 |
| UK_place | 41.46 | 42.38 | 0.92 |
| Military_unit | 37.74 | 38.58 | 0.84 |
| Military_conflict | 18.58 | 19.21 | 0.63 |
| WikiBio | 44.28 | 44.59 | 0.31 |
| NCAA_team_season | 73.59 | 73.87 | 0.28 |
| French_commune | 33.64 | 34.72 | 1.08 |
| Settlement | 77.59 | 77.68 | 0.09 |
| Video_game | 29.54 | 29.48 | -0.06 |

Although the English Wikipedia has about 5 million entities, we totally parsed 1,656,458 infoboxes and drop some of them due to data noise which indicates nearly 2/3 entities have no infobox.
4.3 Training Details

The dimensions of all trainable word embeddings are set to 512, and the GRU hidden states sizes are set to 512. To limit the memory of our model, we set the maximum number of facts per table to 500. We initialize all the model parameters randomly using a uniform distribution between -0.08 and 0.08. For the model training, we use Adam (Kingma and Ba, 2014) with initial learning rate of 0.001 as the optimization algorithm. The training batch size is set to 64. We also apply gradient clipping (Pascanu et al., 2013) with range [-1,1] during training. We conduct experiments using single card NVIDIA Tesla V100. The largest 15 datasets with more than 9500 training instances are trained for 20 epochs, while the remaining 6 datasets are trained for 40 epochs. All the models were selected based on BLEU-4 score on the development set. All the experiments use greedy search as the decoding algorithm during testing.

4.4 Main Results

4.4.1 Overall Results

To verify the effectiveness of KBAtt, we conduct experiments on 21 Wikipedia infobox-to-text datasets. Table 2 shows the performances of KBAtt and Seq2Seq+Copy in terms of BLEU-4 score. As we can see, KBAtt consistently outperforms Seq2Seq+Copy on most of the datasets, i.e., more than 0.5 BLEU-4 improvements on 15 out of 21 datasets, and comparable on the remaining 6 categories. To get a better understanding of when and why external knowledge will be effective or not, we correlate the performance gains in terms of BLEU-4 with KBGain metric described in Section 3 across 21 datasets, and plot their values on the scatter plot in Fig. 4. The Pearson correlation coefficient ($\rho$) between BLEU-4 improvements and KBGain is 0.716 which indicates a strong positive correlation between them. This confirms our analysis in Section 3 that KBAtt is effective when the external knowledge is relevant and the original data is sparse.

4.4.2 Results on WikiBio

To compare with state-of-the-art models, Table 3 shows the results on the WikiBio dataset. Among them, our baseline Seq2Seq+Copy gains a performance of 44.28 which is comparable to Sha et al. (2017) and Liu et al. (2017) and significantly better than Lebret et al. (2016) and Bao et al. (2018), and this baseline model is general without specific design, e.g., order planning from Sha et al. (2017) or field-gating encoder from Liu et al. (2017). Our proposed model, namely KBAtt, obtains a BLEU-4 score of 44.59. Although the absolute improvement (+0.31 BLEU-4 points) of KBAtt over Seq2Seq+Copy is relative small, the difference between them is statistically significant under the one-tailed paired t-test at the 99% significance level. The reason why the absolute improvement is relative small is that the full WikiBio dataset consists of 728,321 parallel data-to-text pairs which are enough for neural models to memorize certain inference patterns for high frequency pairs. However, as will be shown in the Section 4.5, the baseline fails on low co-occurrence frequency pairs, but the KBAtt avoids this problem with the contributions from the external knowledge.

4.4.3 Human Evaluation

We conduct human evaluation to assess the factual accuracy of the generated sentences. Manually evaluating the generated results of all datasets is labour intensive, so we choose to evaluate two sample datasets for case study purposes. Specifically, we sample 50 instances each from the WikiBio and the Album datasets, and ask two annotators to extract facts tuples (subject, relation, object) from the references and the generated sentences. In table 4, precision $P_1$, recall $R_1$ and their overall score $F_1$ measure the extent that the facts extracted from the generated sentences conform to those from references. As we can see, KBAtt achieves 1.21%, 7.42% improvements in terms of $F_1$ on WikiBio and Album respectively, which shows that KBAtt can generate more relevant facts with respect to the reference than the

---

The generated sentences are shuffled to avoid model preferences. The Cohen’s kappa between two annotators is 0.776.
Table 3: BLEU-4 scores (%) on WikiBio dataset.

| Model                          | BLEU-4 |
|--------------------------------|--------|
| Table NLM (Lebret et al., 2016)| 34.70  |
| Table2Seq (Bao et al., 2018)    | 40.26  |
| Order Planning, full model (Sha et al., 2017) | 43.91  |
| Field-gating Seq2Seq, full model (Liu et al., 2017) | 44.71  |
| Seq2Seq+Copy                    | 44.28  |
| KBAtt                           | 44.59  |

Table 4: Human based evaluation results.

| Dataset | Model          | $P_1$ | $R_1$ | $F_1$ | $P_2$ |
|---------|----------------|-------|-------|-------|-------|
| WikiBio | Seq2Seq+Copy   | 69.07%| 62.45%| 65.59%| 88.35%|
| WikiBio | KBAtt          | 71.81%| 62.45%| 66.80%| 91.85%|
| Album   | Seq2Seq+Copy   | 73.33%| 71.84%| 72.58%| 84.38%|
| Album   | KBAtt          | 81.70%| 78.37%| 80.00%| 90.64%|

4.5 Analysis of Few-Shot Learning Ability

To examine the ability of learning writing knowledge from few examples, we design an experiment to compare the performance under different number of training samples for the baseline and our model. In the training set of WikiBio, about 78.5% tables contain place of birth field but only 19.0% tables include the nationality field. However, in the references, nationality of a person is frequently mentioned. This means that the ability of inferring nationality based on place of birth is important. We collect the cases from the test data with the following conditions: (a) only city level information is given in the field of place of birth; (b) nationality is not specified in the table; (c) the reference mentions country name or nationality. From these cases, we get over 400 unique places of birth to nationality inference pairs and split them into two intervals $[1, 25)$ and $[25, \infty)$ according to their co-occurrence frequency in the training set. For each interval, we randomly sample 50 test cases and manually assess the accuracy of nationality information mentioned in the generated sentences.

Fig. 5 shows the accuracy of nationality information in the generated text with respect to different co-occurrence frequency intervals: Firstly, we found that the baseline model struggles to learn the inference from place of birth to nationality when their co-occurrence frequency in training set is less than 25, which shows the difficulty of this task; Secondly, in interval $[1, 25)$, the baseline model only gets 42.0% accuracy, but our model achieves 80.0% accuracy, a 38.0% absolute gain. This confirms our motivation that incorporating external knowledge into neural models can improve model performances especially when the original data is sparse; Finally, the accuracy of both models increases as the frequency goes from $[1, 25)$ to $[25, \infty)$, and the improvement of our model over the baseline model gradually narrows, from 38.0% to 6.0%. This shows that the baseline model could learn part of inference patterns when enough parallel data is available.

4.6 Case Study

Fig. 6 shows three examples from the development set of WikiBio dataset which can demonstrate how KBAtt succeeds or fails. Fig. 6 (a) and (b) illustrate KBAtt is helpful when original data is sparse. However when original data is dense, the baseline model still can learn it. As shown in Fig. 6 (a), the baseline model struggles to learn direct association between “german” and “gummersbach” (birth place in the table), since they only co-occur 12 times in the training data. But it is easy for KBAtt to learn since “german” co-occurs with “germany” in KB 12,437 times in the training data. As shown in Fig. 6 (b), although the national-
5 Related Work

Data-to-text generation is an important task in natural language generation which has been studied for decades (Kukich, 1983; Holmes-Higgin, 1994; Reiter and Dale, 1997). This task is broadly divided into two subproblems: content selection (Kukich, 1983; Reiter and Dale, 1997; Duboue and McKeown, 2003; Barzilay and Lapata, 2005) and surface realization (Goldberg et al., 1994; Reiter et al., 2005).

With the advent of neural text generation, the distinction between content selection and surface realization becomes blurred. For example, Mei et al. (2016) proposed an end-to-end encoder-aligner-decoder model to learn both content selection and surface realization jointly which shows good results on WeatherGov and RoboCub datasets. Wiseman et al. (2017) generate long descriptive game summaries from a database of basketball games where they show the current state-of-the-art neural models are quite good at generating fluent outputs, but perform poorly in content selection and capturing long-term structure. Our work falls into the task of single sentence generation from Wikipedia infoboxes. The model structure ranges from feed-forward networks work (Lebret et al., 2016) to encoder-decoder models (Sha et al., 2017; Liu et al., 2017; Bao et al., 2018; Nema et al., 2018). Recently, Perez-Beltrachini and Lapata (2018) generalize this task to multi-sentence text generation, where they focus on bootstrapping generators from loosely aligned data. However, most of the work mentioned above assume all the writing knowledge can be learned from massive parallel pairs of training data. Different from the previous work, we exploit incorporating external
knowledge into this task to improve the fidelity of generated text.

Our work is also relevant to recent works on integrating external knowledge into neural models for other NLP tasks. The motivations of incorporating external knowledge range from enriching the context information (Mihaylov and Frank, 2018) in reading comprehension, improving the inference ability of models (Chen et al., 2018) in natural language inference, to providing the model a knowledge source to copy from in language modelling (Ahn et al., 2016). Our model, KBAtt, is most relevant to Mihaylov and Frank (2018), where they focus on similarity calculation but we focus on generation in this paper. Moreover, in addition to demonstrating the positive effect of incorporating external knowledge as previous work, we also design a new metric to quantify the potential gains of external knowledge for a specific dataset which can explain when and why our model is effective.

6 Conclusion

In this paper, we propose a neural data-to-text generation model, KBAtt, that incorporates external background knowledge in a simple but effective way to improve fidelity of the generated text. Experiments on 21 Wikipedia infobox-to-text datasets show KBAtt consistently achieves better performance in BLEU than a state-of-the-art baseline on most of datasets. Meanwhile, to quantify when and why external knowledge is effective, we design a metric, KBGain, which shows a strong correlation with observed performance boost. This result indicates the relevance of external knowledge and sparseness of original data are the main factors affecting the effectiveness of KBAtt.

In the future, we plan to investigate integrating multi-hops knowledge graph behind the data which has potential to further improve the inference ability of neural models. It will be worthwhile especially when we extend the task to multiple sentence generation. The main challenge in integrating multi-hops knowledge graph is the large search space. We plan to employ reinforcement learning based techniques to allow the model to search the optimal inference paths by trial and error. Besides, we are also interested in integrating external knowledge into other types of datasets beyond Wikipedia infobox-to-text datasets.

7 Acknowledgement

We would like to thank the anonymous reviewers for their helpful comments. The work of this paper is funded by the project of National Key Research and Development Program of China (No. 2018YFC0832105).

References

Sungjin Ahn, Heeyoul Choi, Tanel Pärnamaa, and Yoshua Bengio. 2016. A neural knowledge language model. arXiv preprint arXiv:1608.00318.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. International Conference on Learning Representations.

Junwei Bao, Duyu Tang, Nan Duan, Zhao Yan, Yuanhua Lv, Ming Zhou, and Tiejun Zhao. 2018. Table-to-text: Describing table region with natural language. AAAI.

Regina Barzilay and Mirella Lapata. 2005. Collective content selection for concept-to-text generation. In Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, pages 331–338. ACL.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, and Diana Inkpen. 2018. Natural language inference with external knowledge. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics.

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. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing.

Pablo A Duboue and Kathleen R McKeown. 2003. Statistical acquisition of content selection rules for natural language generation. In Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, pages 121–128. ACL.

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2018. Findings of the E2E NLG Challenge. In Proceedings of the 11th International Conference on Natural Language Generation, Tilburg, The Netherlands. ArXiv:1810.01170.

Eli Goldberg, Norbert Driedger, and Richard I Kittridge. 1994. Using natural-language processing to produce weather forecasts. IEEE Intelligent Systems, (2):45–53.

Paul Holmes-Higgin. 1994. Text generation: using discourse strategies and focus constraints to generate natural language text. The Knowledge Engineering Review, 9(4):421–422.
Our dataset is an extension of WikiBio dataset (Lebret et al., 2016) where we extend previous single Biography domain to other twenty domains (e.g. Album, Book etc.) and augment the infoboxes with external knowledge from Wikidata. Our dataset consists of infoboxes as input data and the first sentence of their corresponding Wikipedia article as reference. The infoboxes are parsed from English Wikipedia dump (Sep 2015) while the reference sentences are tokenized with Stanford CoreNLP (Manning et al., 2014). Different from (Lebret et al., 2016), we keep the capitalization information which we believe is important for real-world application. To augment the infoboxes with external knowledge, we preserve the Wikipedia internal hyperlink information in the field values of infoboxes, and track these hyperlinks to get their corresponding entities from Wikidata. Finally we retrieve one-hop facts from Wikidata.

The dataset is available at https://github.com/hitercs/WikiInfo2Text.
in Wikipedia. However, the size of remaining datasets ranges from 4,559 to 327,863 and more than 60% of them have less than 20,000 instances. So, the abundance of data such as WikiBio is not common among the Wikipedia infobox-to-text datasets. Meanwhile, these datasets vary in sentence length and length of tokens in table. In addition, the number of entities extracted from each table ranges from 0.8 to 10.1 and the average number of fact tuples per entity ranges from 5.2 to 126.1.

### A.3 Threshold Tuning for KBGain metric

![Figure 7](image.png)

Figure 7: Pearson correlation coefficient w.r.t Co-occurrence frequency threshold.

The threshold $\gamma$ in KBGain metric denotes the minimum size of co-occurrence above which learning will be effective. We tune this threshold by finding the maximum Pearson correlation coefficient ($\rho$) between BLEU-4 improvements and the values of KBGain on the 21 development datasets. Fig. 7 shows the curve of Pearson correlation coefficient w.r.t co-occurrence frequency threshold ranging from 5 to 100. As we can see, $\rho$ reaches peak value as 0.753 when the threshold is 25. So $\gamma$ is set 25 throughout this paper.

| Dataset              | Avg. # tokens per sentence | Avg. # tokens per table | Avg. # fields per table | Avg. # table tokens per sent. | Avg. # entities per table | Avg. # fact tuples per entity | Dataset size |
|----------------------|-----------------------------|-------------------------|-------------------------|-------------------------------|--------------------------|--------------------------------|--------------|
| WikiBio              | 26.1                        | 53.0                    | 19.7                    | 11.6                          | 5.9                      | 19.0                           | 728321       |
| Settlement           | 22.1                        | 57.3                    | 50.6                    | 10.3                          | 6.6                      | 14.4                           | 327863       |
| Album                | 21.0                        | 57.1                    | 15.9                    | 9.7                           | 5.5                      | 7.2                            | 76105        |
| Single               | 24.3                        | 70.5                    | 19.2                    | 12.6                          | 7.3                      | 7.5                            | 36937        |
| NRHP                 | 23.0                        | 54.4                    | 22.7                    | 10.8                          | 1.4                      | 10.9                           | 35945        |
| French_commune       | 14.7                        | 31.1                    | 20.3                    | 3.7                           | 0.8                      | 126.1                          | 29218        |
| Book                 | 24.3                        | 41.7                    | 20.1                    | 11.1                          | 4.2                      | 16.7                           | 23859        |
| School               | 24.2                        | 62.6                    | 44.3                    | 11.2                          | 3.7                      | 29.8                           | 18751        |
| Station              | 19.9                        | 51.8                    | 32.6                    | 9.0                           | 3.2                      | 18.0                           | 16625        |
| Video_game           | 25.0                        | 42.8                    | 13.2                    | 9.4                           | 6.3                      | 6.4                            | 16381        |
| UK_place             | 20.5                        | 30.9                    | 19.4                    | 7.0                           | 3.0                      | 5.2                            | 12279        |
| Military_unit        | 25.4                        | 62.5                    | 27.9                    | 11.9                          | 5.2                      | 18.9                           | 11273        |
| Airport              | 22.9                        | 74.4                    | 32.9                    | 10.7                          | 3.0                      | 15.0                           | 10934        |
| Building             | 24.1                        | 47.6                    | 35.8                    | 10.4                          | 2.9                      | 33.6                           | 10052        |
| Military_conflict    | 34.7                        | 96.6                    | 18.0                    | 18.6                          | 8.1                      | 17.3                           | 9592         |
| NCAA_team_season     | 24.2                        | 45.1                    | 23.4                    | 12.7                          | 2.7                      | 5.5                            | 7766         |
| Television_episode   | 26.4                        | 78.8                    | 15.9                    | 12.4                          | 10.1                     | 8.0                            | 5930         |
| UK_school            | 23.8                        | 51.6                    | 48.0                    | 10.1                          | 4.1                      | 10.0                           | 4974         |
| Australian_place     | 21.3                        | 43.8                    | 29.1                    | 8.2                           | 5.6                      | 6.5                            | 4972         |
| Automobile           | 23.3                        | 70.9                    | 17.2                    | 7.8                           | 5.7                      | 18.5                           | 4781         |
| Football_club_season | 24.6                        | 82.8                    | 30.0                    | 9.3                           | 8.8                      | 6.9                            | 4559         |

Table 5: Dataset statistics for all datasets used in this paper.