KG-to-Text Generation with Slot-Attention and Link-Attention

Yashen Wang[1] ⋆, Huanhuan Zhang[1], Yifeng Liu[1], and Haiyong Xie[1,2]

[1]China Academy of Electronics and Information Technology of CETC, Beijing, China
[2]University of Science and Technology of China, Hefei, Anhui, China
yashen_wang@126.com;huanhuanz_bit@139.com;
yliu@csdslab.net;haiyong.xie@ieee.org

Abstract. Knowledge Graph (KG)-to-Text generation task aims to generate a text description for a structured knowledge which can be viewed as a series of slot-value records. The previous seq2seq models for this task fail to capture the connections between the slot type and its slot value and the connections among multiple slots, and fail to deal with the out-of-vocabulary (OOV) words. To overcome these problems, this paper proposes a novel KG-to-text generation model with hybrid of slot-attention and link-attention. To evaluate the proposed model, we conduct experiments on the real-world dataset, and the experimental results demonstrate that our model could achieve significantly higher performance than previous models in terms of BLEU and ROUGE scores.

Keywords: Text Generation · Knowledge Graph · Attention Mechanism.

1 Introduction

Generation of natural language description from structured Knowledge Graph (KG) is essential for various Natural Language Processing (NLP) tasks such as question answering and dialog systems [29,24,26,22]. As shown in Figure 1, for example, a biographic infobox is a fixed-format table that describes a person with many "slot-value" records like (*Name*, *Charles John Huffam Dickens*), etc. We aim at filling in this knowledge gap by developing a model that can take a KG (consisted of a set of slot types and their values) about an entity as input, and automatically generate a natural language description.

As discussed in [37], recently text generation task is usually accomplished by human-designed rules and templates [43][23][3][9][8]. Generally speaking, high-quality descriptions could be released from these models, however the results heavily rely on information redundancy to create templates and hence the generated texts are not flexible. In early years, researchers apply language model (LM) [4,20,21,33] and neural networks (NNs) [41,18,34,40] to generate texts from structured data [26,37], where a neural encoder captures table-formed information and, a recurrent neural network (RNN) decodes these information to a natural language sentence [25,22,49]. The previous work usually considers the slot type and slot value as two sequences and applies

⋆ The corresponding author.

Knowledge:

Row	Slot Type	Slot Value
1	Name	Charles John Huffam Dickens
2	Born	7 February 1812 Landport, Hampshire, England
3	Dicd	9 June 1870 (aged 58) Higham, Kent, England
4	Resting place	Poets' Corner, Westminster Abbey
5	Occupation	Writer
6	Nationality	British
7	Genre	Fiction
8	Notable work	The Pickwick Papers

Text:

Charles John Huffam Dickens (7 Feb 1812 – 9 Jun 1870) was a British writer best known for his fiction The Pickwick Papers.

Fig. 1: Wikipedia infobox about Charles Dickens and its corresponding generated description.

a sequence-to-sequence (seq2seq) framework [29][42][25][46][37][29] for generation. However, in the task of describing structured KG, we need to cover the knowledge elements contained in the input KG (but also attend to the out-of-vocabulary (OOV) words), and generally speaking generating natural language description mainly aims at clearly describing the semantic connections among these knowledge elements in an accurate and coherent way. Therefore, the previous seq2seq models: (i) fail to capture such correlations and hence are apt to release wrong description; (ii) fail to deal with OOV words.

To address this challenge of considering OOV words, we choose a pointer network [35][44][30][35] to copy slot values directly from the input KG, which is designed to automatically capture the particular source words and directly copy them into the target sequence[10][35]. By leveraging attention mechanism [14,47] for booting the performance of traditional pointer network, we introduce a *Slot-Attention* mechanism to model slot type attention and slot value attention simultaneously and capture their correlation. In parallel, attention model has gained popularity recently in neural NLP research, which allows the models to learn the alignments between different modalities [2,50,27,32,51,34], and has been used to improve many neural NLP studies by selectively focusing on parts of the source data [31,14,47,11]. Besides, we could show by inspection that, multiple slots in the structured knowledge are often inter-dependent [19][48]. For example, an actor (or actress) player may join multiple movies, with each movie associated with a certain number of “premiere time”s, “reward”s, and so on. Hence, we also design a novel *Link-Attention* mechanism to capture correlations among multiple inter-dependent slots [28][38][39].

Moreover, the structured slots from Wikipedia Infoboxes and Wikidata [45] and the corresponding sentences describing these slots in Wikipedia articles, which has been proved to be available for diversified characteristics on expression, are leveraged for training the proposed model. Especially, a biographic infobox is a fixed-format table that describes a person with many <slot type, slot value> records like (*Name*, *Charles John Huffam Dickens*), (*Nationality*, *British*), (*Occupation*, *Writer*), etc, as shown in Figure 1 . Finally, we evaluated our method on the widely-used `WIKBIO` dataset [25]. Experimental results show that the proposed approach significantly outperforms previous state-of-the-art results in terms of BLEU and ROUGE metrics.

2 Task Definition

We formulate the input structured KG to the model as a list of triples:

$$\mathcal{X} = [(s_1, v_1,), \ldots, (s_n, v_n,)] \tag{1}$$

Wherein s_i denotes a slot type (e.g., "*Nationality*" or "*Occupation*" in Figure 1), and v_i denotes the corresponding slot value (e.g., "*British*" or "*Writer*" in Figure 1). The outcome of the model is a paragraph: $\mathcal{Y} = [y_1, y_2, \ldots, y_m]$. The training instances for the generator are provided in the form of: $\mathcal{T} = [(\mathcal{X}_1, \mathcal{Y}_1), \ldots, (\mathcal{X}_k, \mathcal{Y}_k)]$.

3 KG-to-Text Generation with Slot-Attention and Link-Attention

3.1 Slot-Attention Mechanism

Following previous research [25][46][37][29], sequence-to-sequence (seq2seq) framework is utilized here for describing structured knowledge.

Encoder: Given a structured KG input $\mathcal{X}$ (defined in Section 2), where $\{s_i, v_i\}$ are randomly embedded as vectors $\{\mathbf{s}_i, \mathbf{v}_i\}$ respectively, we concatenate the vector representations of these slots as $\Phi_i = [\mathbf{s}_i; \mathbf{v}_i]$, and obtain $[\Phi_1, \Phi_2, \ldots, \Phi_n]$. Intuitively, with efforts above, we then utilize a bi-directional Gated Recurrent Unit (GRU) encoder [6][15] on $[\Phi_1, \Phi_2, \ldots, \Phi_n]$ to release the encoder hidden states $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n]$, where $\mathbf{h}_i$ is a hidden state for $\mathbf{I}_i$.

Decoder: Following [46], the forward GRU network with an initial hidden state $\mathbf{h}_n$ is leveraged for the construction of the decoder in the proposed model. In order to capture the correlation between a slot type and its slot value (as shown in Fig. 2), we also design a *slot-attention* similar to the strategy used in [46]. Hence, we could generate the attention distribution over the sequence of the input triples at each step t . For each slot i, we assign it with an slot-attention weight, as follows:

$$\alpha_{t,i}^{slot} = \mathrm{Softmax}(\mathbf{v}_i^\top \tanh(\mathbf{W}_h \mathbf{h}_t + \mathbf{W}_s \mathbf{s}_i + \mathbf{W}_v \mathbf{v}_i + \mathbf{b}_{slot})) \tag{2}$$

where $\mathbf{h}_t$ indicates the decoder hidden state at step t. $\mathbf{s}_i$ and $\mathbf{v}_i$ indicate the vector representations of slot type s_i and slot value v_i, respectively. $\{\mathbf{W}_h, \mathbf{W}_s, \mathbf{W}_v, \mathbf{b}_{slot}\}$

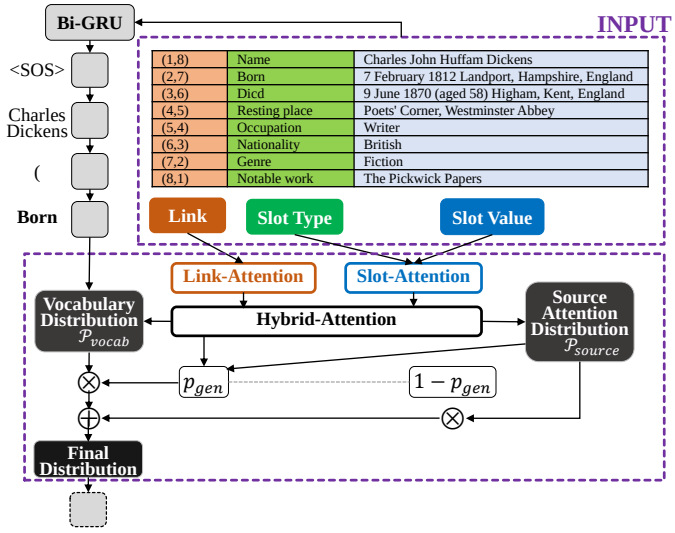

Fig. 2: Overview of the proposed KG-to-Text Generation Model with Slot-Attention and Link-Attention.

is a part of model parameters and learned by backpropagation (details in Section 4). Furthermore, the slot-attention weight distribution $\alpha_{t,i}^{slot}$ is utilized to generate the representation of the slot type $\mathbf{s}^*$ and the representation of the slot value $\mathbf{v}^*$ respectively:

$$\mathbf{s}^* = \sum_{i=1}^{n} \alpha_{t,i}^{slot} \mathbf{s}_i \tag{3}$$

$$\mathbf{v}^* = \sum_{i=1}^{n} \alpha_{t,i}^{slot} \mathbf{v}_i \tag{4}$$

Finally, the loss function is computed as follows:

$$\ell = \sum_{t} \{-\log \mathcal{P}_{vocab}(y_t) + \lambda \sum_{i} \min(\alpha_{t,i}^{slot}, c_{t,i})\} \tag{5}$$

wherein, notation $\mathcal{P}_{vocab}$ in Eq(5) indicates the vocabulary distribution (shown in Fig. 2) , which could be computed with the decoder hidden state $\mathbf{h}_t$ and the context vectors $\{\mathbf{s}^*, \mathbf{v}^*\}$ at step t, as follows:

$$\mathcal{P}_{vocab} = \text{Softmax}(\mathbf{V}_{vocab}[\mathbf{h}^t; \mathbf{s}^*; \mathbf{v}^*] + \mathbf{b}_{vocab}) \tag{6}$$

Wherein, $\{\mathbf{V}_{vocab}, \mathbf{b}_{vocab}\}$ is a part of model parameters and learned by backpropagation. With efforts above, we could define $\mathcal{P}_{vocab}(y_t)$ in Eq (5) as the prediction probability of the ground truth token y_t. Moreover, in Eq (5), λ is a hyperparameter.

3.2 Link-Attention Mechanism

Moreover, we propose a *link-attention* mechanism which directly models the relationship between different slots and the order information among multiple slots. Our intuition is derived from the observation that, a well-organized text typically has a reasonable order of its contents [37].

We construct a link matrix $\mathbf{L} \in \mathrm{R}^{n_s \times n_s}$, where n_s indicates the number possible slot types in the give structured knowledge graph. The element $\mathbf{L}[j, i]$ is a real-valued score indicating how likely the slot j is mentioned after the slot i . The link matrix $\mathbf{L}$ is a part of model parameters and learned by backpropagation (details in Section 4). Let $\alpha_{t-1,i}^{hybrid}(i = 1, \ldots, n)$ be an attention probability over content words (i.e., $\{s_i, v_i\}$) in the last time step $(t-1)$ during generation. Here, $\alpha_{t-1,i}^{hybrid}$ refers to the hybrid *solt*-attention and *link*-attention, which will be introduced shortly. For a particular data sample whose content words are of slots $\{s_i | i \in [1, n]\}$, we first weight the linking scores by the previous attention probability, and then normalize the weighted score to obtain link-attention probability in the format of softmax (as shown in Fig. 2), as follows:

$$\alpha_{t,i}^{link} = \mathrm{softmax}(\sum_{j=1}^{n} \alpha_{t-1,i}^{hybrid} \cdot \mathbf{L}[j, i]) = \frac{\exp(\sum_{j=1}^{n} \alpha_{t-1,i}^{hybrid} \cdot \mathbf{L}[j, i])}{\sum_{i'=1}^{n} \exp(\sum_{j=1}^{n} \alpha_{t-1,i'}^{hybrid} \cdot \mathbf{L}[j, i'])} \quad (7)$$

3.3 Hybrid-Attention based on Slot-Attention and Link-Attention

To combine the above two attention mechanisms (i.e., slot-attention in Section 3.1 and link-attention in Section 3.2), we use a self-adaptive gate $g_{hybrid} \in (0, 1)$ [37] by a sigmoid unit, as follows:

$$g_{hybrid} = \sigma(\mathbf{W}_{hybrid}[\mathbf{h}_{t-1}; \mathbf{e}_t; \mathbf{y}_{t-1}]) \quad (8)$$

Wherein $\mathbf{W}_{hybrid}$ is a parameter vector, $\mathbf{h}_{t-1}$ is the last step's hidden state, $\mathbf{y}_{t-1}$ is the embedding of the word generated in the last step $(t-1)$, $\sigma(\cdot)$ is a Sigmoid function, and $\mathbf{e}_t$ is the sum of slot type embeddings $\mathbf{s}_i$ weighted by the current step's link-attention $\alpha_{t,i}^{link}$. As $\mathbf{y}_{t-1}$ and $\mathbf{e}_t$ emphasize the slot and link aspects, respectively, the self-adaptive g_{hybrid} is aware of both (Configuration of the self-adaptive g_{hybrid} is discussed in Section 4). Finally, the hybrid attention, a probabilistic distribution over all content words (i.e., $\{s_i, v_i\}$), is given by:

$$\alpha_{t,i}^{hybrid} = \tilde{g}_{hybrid} \cdot \alpha_{t,i}^{slot} + (1 - \tilde{g}_{hybrid}) \cdot \alpha_{t,i}^{link} \quad (9)$$

With efforts above, we could replace $\alpha_{t,i}^{slot}$ the with $\alpha_{t,i}^{hybrid}$ in Eq. (3), Eq. (4) and loss function Eq. (5), respectively. Therefore, these equations could be updated as follows:

$$\mathbf{s}^* = \sum_{i=1}^{n} \alpha_{t,i}^{hybrid} \mathbf{s}_i \tag{10}$$

$$\mathbf{v}^* = \sum_{i=1}^{n} \alpha_{t,i}^{hybrid} \mathbf{v}_i \tag{11}$$

$$\ell = \sum_t \{-\log \mathcal{P}_{vocab}(y_t) + \lambda \sum_i \min(\alpha_{t,i}^{hybrid}, c_{t,i})\} \tag{12}$$

3.4 Sentence Generation

To deal with the challenge of out-of-vocabulary (OOV) words, we aggregate the attention weights for each unique slot value v_i from $\{\alpha_{t,i}^{hydrid}\}$ and obtain its aggregated source attention distribution $\mathcal{P}_{source}^i = \sum_{m|v_m=v_i} \alpha_{t,m}^{hydrid}$. With efforts above, we could obtain a source attention distribution of all unique input slot values. Furthermore, for the sake of the combination of two types of attention distribution $\mathcal{P}_{source}$ and $\mathcal{P}_{vocab}$ (in Eq. (6)), we introduce a structure-aware gate $p_{gen} \in [0, 1]$ as a soft switch between generating a word from the fixed vocabulary and copying a slot value from the structured input:

$$p_{gen} = \sigma(\mathbf{W}_s \mathbf{s}^* + \mathbf{W}_v \mathbf{v}^* + \mathbf{W}_h \mathbf{h}_t + \mathbf{W}_y \mathbf{y}_{t-1} + \mathbf{b}_{gen}) \tag{13}$$

where $\mathbf{y}_{t-1}$ is the embedding of the previous generated token at step $t-1$. The final probability of a token y at step t can be computed by p_{gen} in Eq (13), $\mathcal{P}_{vocab}$ and $\mathcal{P}_{source}$, as follows:

$$\mathcal{P}(y_t) = p_{gen} \cdot \mathcal{P}_{vocab} + (1 - p_{gen}) \cdot \mathcal{P}_{source} \tag{14}$$

Finally, the loss function (Eq. (12)) could be reformed as follows:

$$\ell = \sum_t \{-\log \mathcal{P}(y_t) + \lambda \sum_i \min(\alpha_{t,i}^{hybrid}, c_{t,i})\} \tag{15}$$

4 Experiments and Results

We introduce dataset `WIKBIO` to compare our model with several baselines After that, we assess the performance of our model on KG-to-Text generation.

4.1 Datasets and Evaluation Metrics

Dataset We use WIKBIO dataset proposed by [25] as the benchmark dataset. WIKBIO contains 728,321 articles from English Wikipedia (Sep 2015). The dataset uses the first sentence of each article as the description of the corresponding infobox. Note that, As shown before, one challenge of KG-to-Text task lies in how to generate a wide variety of expressions (templates and styles which human use to describe the same slot type). For example, to describe a writer's notable work, we could utilize various phrases including "(best) known for", "(very) famous for" and so on. And for that, the existing pairs of structured slots from Wikipedia Infoboxes and Wikidata [45] and the corresponding sentences describing these slots in Wikipedia articles are introduced here as our training data in the proposed model, with independence of human-designed rules and templates [1][7][22].

Table 1 summarizes the dataset statistics: on average, the tokens in the infobox (53.1) are twice as long as those in the first sentence (26.1). 9.5 tokens in the description text also occur in the infobox. The dataset has been divided in to training (80%), testing (10%) and validation (10%) sets.

Table 1: Statistics of WIKBIO dataset.

	#token per sent.	#infobox token per sent.	#tokens per infobox	#slots per infobox
Mean	26.1	9.5	53.1	19.7

Evaluation Metric and Experimental Settings We apply the standard BLEU, METEOR, and ROUGE metrics to evaluate the generation performance, because they can measure the content overlap between system output and ground-truth and also check whether the system output is written in sufficiently good English.

The experimental settings of the proposed model, are concluded as follows: (i) the vocabulary size ($|s| + |v|$) is 46,776; (ii) The value/type embedding size is 256; (iii) The position embedding size is 5; (iv) The slot embedding size is 522; (v) The decoder hidden size is 256; (vi) The coverage loss λ is 1.5; (vii)In practice, we adjust gate $\tilde{g}_{hybrid}$ by $\tilde{g}_{hybrid} = 0.25g_{hybrid} + 0.45$ empirically; (viii) The optimization ADAM [13] Learning rate is 0.001.

4.2 Baselines

We compare the proposed model with several statistical language models and other competitive sequence-to-sequence models. The baselines are listed as follows:

KN: The Kneser-Ney (KN) model is a widely used Language Model proposed by [12]. We use the *KenLM* toolkit to train 5-gram models without pruning following [29].

Template KN: Template KN is a KN model over templates which also serves as a baseline in [25].

NLM: NLM is a naive statistical language model proposed by [25] for comparison, which uses only the slot value as input without slot type information and link position information.

Table NLM: The most competitive statistical language model proposed by [25], including local and global conditioning over the table by integrating related slot and position embedding into the table representation.

Pointer: The Pointer-generator [36] introduces a soft switch to choose between generating a word from the fixed vocabulary and copying a word from the input sequence. Besides, Pointer-generator concatenates all slot values as the input sequence, e.g., {Charles John Huffam Dickens, 7 February 1812 Landport Hampshire England, 9 June 1870 aged 58 Higham Kent England, ... } for Figure 1.

Seq2Seq: The Seq2Seq attention model [2] concatenates slot types and values as a sequence, e.g., {Name, Charles John Huffam Dickens, Born, 7 February 1812 Landport Hampshire England,... } for Figure 1, and apply the sequence to sequence with attention model to generate a text description.

Vanilla Seq2Seq: The vanilla seq2seq neural architecture uses the concatenation of word embedding, slot embedding and position embedding as input, and could operate local addressing over the infobox by the natural advantages of LSTM units and word level attention mechanism, as discussed in [29].

Structure-Aware Seq2seq: It is a structure-aware Seq2Seq architecture to encode both the content and the structure of a table for table-to-text generation [29]. The model consists of field-gating encoder and description generator with dual attention.

4.3 Results and Analysis

The assessment for KG's description generation is listed in Table 2 (referring some results reported in [29]). Besides, the statistical t-test is employed here: To decide whether the improvement by algorithm `A` over algorithm `B` is significant, the t-test calculates a value p based on the performance of `A` and `B`. The smaller p is, the more significant the improvement is. If the p is small enough ($p < 0.05$), we conclude that the improvement is statistically significant. Moreover, the case study of link-attention for person's biography generation is illustrated in Fig. 3.

The results show the proposed model improves the baseline models in most cases. We have following observations: (i) Neural network models perform much better than statistical language models; (ii) neural network-based model are considerably better than traditional **KN** models with/without templates; (iii) The proposed seq2seq architecture can further improve the KG-to-Text generation compared with the competitive **Vanilla Seq2Seq** and **Structure-Aware Seq2Seq**; (iv) slot-attention mechanism and link-attention mechanism are able to boost the model performance by over BLEU compared to vanilla attention mechanism (**Vanilla Seq2Seq**) and dual attention mechanism **Structure-Aware Seq2Seq**. Overall, many recent models [25][5][17][16][29][37] aim at generating a person's biography from an input structure (e.g., example in Fig. 1), including the comparative models mentioned above. The difference could be concluded as follows: instead of modeling the input structure as a single sequence of facts and generating one sentence only, we introduce a link-attention (details in Section 3.2) and a

Table 2: BLEU-4 and ROUGE-4 for the proposed model (Row 9 and Row 10), statistical language models (Row 1-4), and vanilla seq2seq model with slot input (Row 5) and link input (Row 6), structure-aware seq2seq model (Row 7), and Pointer Network (Row 8).

Row	Model	BLEU	ROUGE
1	**KN**	2.21	0.38
2	**Template KN**	19.80	10.70
3	**NLM**	4.17	1.48
4	**Table NLM**	34.70	25.80
5	**Vanilla Seq2Seq (slot)**	43.34	39.84
6	**Vanilla Seq2Seq (link)**	43.65	40.32
7	**Structure-Aware Seq2Seq**	44.89	41.21
8	**Pointer**	43.21	39.67
9	**Pointer+Slot (Ours)**	45.95	42.18
10	**Pointer+Slot+Link (Ours)**	**46.46**	**42.65**

novel hybrid-attention (details in Section 3.3), to capture the dependencies among facts in multiple slots.

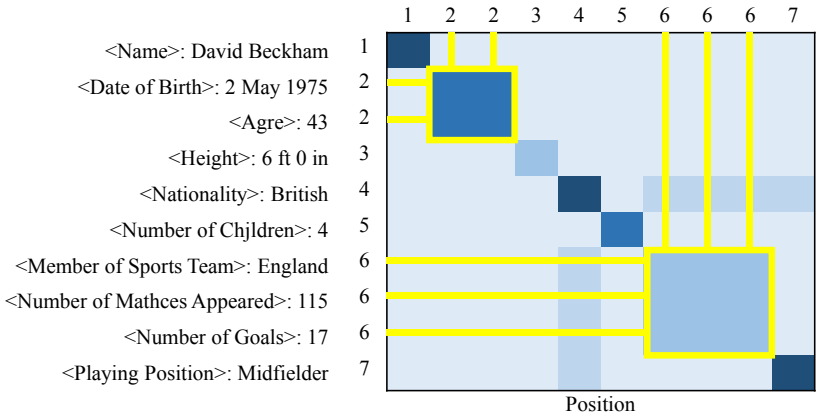

Fig. 3: Link-Attention Visualization (Case study of person's biography generation).

5 Conclusion

The paper proposes a novel KG-to-text generation model with slot-attention and link-attention. We evaluated our approach on the real-world dataset `WIKIBIO`. Experimental results show that we outperform previous results by a large margin in terms of BLEU and ROUGE scores.

Acknowledgements

The authors are very grateful to the editors and reviewers for their helpful comments. This work is funded by: (i) the China Postdoctoral Science Foundation (No.2018M641436); (ii) the Joint Advanced Research Foundation of China Electronics Technology Group Corporation (CETC) (No.6141B08010102); (iii) 2018 Culture and tourism think tank project (No.18ZK01); (iv) the New Generation of Artificial Intelligence Special Action Project (18116001); and (v) the Joint Advanced Research Foundation of China Electronics Technology Group Corporation (CETC) (No.6141B0801010a).

References

1. Angeli, G., Liang, P., Dan, K.: A simple domain-independent probabilistic approach to generation. In: Conference on Empirical Methods in Natural Language Processing, EMNLP 2010, 9-11 October 2010, Mit Stata Center, Massachusetts, Usa, A Meeting of Sigdat, A Special Interest Group of the ACL. pp. 502–512 (2010)
2. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. Eprint Arxiv (2014)
3. Cawsey, A.J., Webber, B.L., Jones, R.B.: Natural language generation in health care. Journal of the American Medical Informatics Association **4**(6), 473–482 (1997)
4. Chen, D.L., Mooney, R.J.: Learning to sportscast:a test of grounded language acquisition. In: International Conference. pp. 128–135 (2008)
5. Chisholm, A., Radford, W., Hachey, B.: Learning to generate one-sentence biographies from wikidata (2017)
6. Cho, K., Merrienboer, B.V., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. Computer Science (2014)
7. Duma, D., Klein, E.: Generating natural language from linked data: Unsupervised template extraction (2013)
8. Flanigan, J., Dyer, C., Smith, N.A., Carbonell, J.: Generation from abstract meaning representation using tree transducers. In: Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 731–739 (2016)
9. Green, N.: Generation of biomedical arguments for lay readers (2006)
10. Gu, J., Lu, Z., Li, H., Li, V.O.K.: Incorporating copying mechanism in sequence-to-sequence learning pp. 1631–1640 (2016)
11. He, R., Lee, W.S., Ng, H.T., Dahlmeier, D.: An unsupervised neural attention model for aspect extraction. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). vol. 1, pp. 388–397 (2017)
12. Heafield, K., Pouzyrevsky, I., Clark, J.H., Koehn, P.: Scalable modified kneser-ney language model estimation. In: Meeting of the Association for Computational Linguistics. pp. 690–696 (2013)
13. Hu, C., Kwok, J.T., Pan, W.: Accelerated gradient methods for stochastic optimization and online learning. In: International Conference on Neural Information Processing Systems (2009)
14. Huang, H., Wang, Y., Feng, C., Liu, Z., Zhou, Q.: Leveraging conceptualization for short-text embedding. IEEE Transactions on Knowledge and Data Engineering **30**, 1282–1295 (2018)

15. Jabreel, M., Moreno, A.: Target-dependent sentiment analysis of tweets using a bi-directional gated recurrent unit. In: Webist 2017 : International Conference on Web Information Systems and Technologies (2017)
16. Kaffee, L.A., Elsahar, H., Vougiouklis, P., Gravier, C., Laforest, F., Hare, J., Simperl, E.: Mind the (language) gap: Generation of multilingual wikipedia summaries from wikidata for articleplaceholders. In: European Semantic Web Conference. pp. 319–334 (2018)
17. Kaffee, L.A., Elsahar, H., Vougiouklis, P., Gravier, C., Laforest, F., Hare, J., Simperl, E.: Learning to generate wikipedia summaries for underserved languages from wikidata. In: Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume. pp. 640–645 (2018)
18. Kiddon, C., Zettlemoyer, L., Choi, Y.: Globally coherent text generation with neural checklist models. In: Conference on Empirical Methods in Natural Language Processing. pp. 329–339 (2016)
19. Kim, Y., Denton, C., Hoang, L., Rush, A.M.: Structured attention networks (2017)
20. Konstas, I., Lapata, M.: Concept-to-text generation via discriminative reranking. In: Meeting of the Association for Computational Linguistics: Long Papers. pp. 369–378 (2012)
21. Konstas, I., Lapata, M.: Unsupervised concept-to-text generation with hypergraphs. In: Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 752–761 (2012)
22. Konstas, I., Lapata, M.: A global model for concept-to-text generation. AI Access Foundation (2013)
23. Kukich, K.: Design of a knowledge-based report generator. In: Meeting of the ACL. pp. 145–150 (1983)
24. Laha, A., Jain, P., Mishra, A., Sankaranarayanan, K.: Scalable micro-planned generation of discourse from structured data. CoRR **abs/1810.02889** (2018)
25. Lebret, R., Grangier, D., Auli, M.: Neural text generation from structured data with application to the biography domain (2016)
26. Liang, P., Jordan, M.I., Dan, K.: Learning semantic correspondences with less supervision. In: Joint Conference of the Meeting of the ACL and the International Joint Conference on Natural Language Processing of the Afnlp: Volume. pp. 91–99 (2009)
27. Lin, Y., Shen, S., Liu, Z., Luan, H., Sun, M.: Neural relation extraction with selective attention over instances. In: Proceedings of ACL. vol. 1, pp. 2124–2133 (2016)
28. Lin, Z., Feng, M., Santos, C.N.D., Yu, M., Xiang, B., Zhou, B., Bengio, Y.: A structured self-attentive sentence embedding (2017)
29. Liu, T., Wang, K., Sha, L., Chang, B., Sui, Z.: Table-to-text generation by structure-aware seq2seq learning. CoRR **abs/1711.09724** (2018)
30. Luong, M.T., Sutskever, I., Le, Q.V., Vinyals, O., Zaremba, W.: Addressing the rare word problem in neural machine translation. Bulletin of University of Agricultural Sciences and Veterinary Medicine Cluj-Napoca. Veterinary Medicine **27**(2), 82–86 (2014)
31. Luong, T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. In: EMNLP (2015)
32. Ma, S., Sun, X., Xu, J., Wang, H., Li, W., Su, Q.: Improving semantic relevance for sequence-to-sequence learning of chinese social media text summarization (2017)
33. Mahapatra, J., Naskar, S.K., Bandyopadhyay, S.: Statistical natural language generation from tabular non-textual data. In: International Natural Language Generation Conference. pp. 143–152 (2016)
34. Mei, H., Bansal, M., Walter, M.R.: What to talk about and how? selective generation using lstms with coarse-to-fine alignment. Computer Science (2015)
35. See, A., Liu, P.J., Manning, C.D.: Get to the point: Summarization with pointer-generator networks pp. 1073–1083 (2017)

36. See, A., Liu, P.J., Manning, C.D.: Get to the point: Summarization with pointer-generator networks. In: ACL (2017)
37. Sha, L., Mou, L., Liu, T., Poupart, P., Li, S., Chang, B., Sui, Z.: Order-planning neural text generation from structured data. CoRR **abs/1709.00155** (2018)
38. Shen, T., Zhou, T., Long, G., Jiang, J., Pan, S., Zhang, C.: Disan: Directional self-attention network for rnn/cnn-free language understanding (2017)
39. Shen, T., Zhou, T., Long, G., Jiang, J., Zhang, C.: Bi-directional block self-attention for fast and memory-efficient sequence modeling (2018)
40. Song, L., Zhang, Y., Wang, Z., Gildea, D.: A graph-to-sequence model for amr-to-text generation (2018)
41. Sutskever, I., Martens, J., Hinton, G.E.: Generating text with recurrent neural networks. In: International Conference on Machine Learning, ICML 2011, Bellevue, Washington, Usa, June 28 - July. pp. 1017–1024 (2016)
42. Tang, Y., Xu, J., Matsumoto, K., Ono, C.: Sequence-to-sequence model with attention for time series classification. In: IEEE International Conference on Data Mining Workshops. pp. 503–510 (2017)
43. Turner, R., Sripada, Y., Reiter, E.: Generating approximate geographic descriptions. Springer Berlin Heidelberg (2010)
44. Vinyals, O., Fortunato, M., Jaitly, N.: Pointer networks. In: International Conference on Neural Information Processing Systems (2015)
45. Vrandei, D., Krtzsch, M.: Wikidata: a free collaborative knowledgebase. Communications of the Acm **57**(10), 78–85 (2014)
46. Wang, Q., Pan, X., Huang, L., Zhang, B., Jiang, Z., Ji, H., Knight, K.: Describing a knowledge base. In: INLG (2018)
47. Wang, Y., Huang, H., Feng, C., Zhou, Q., Gu, J., Gao, X.: Cse: Conceptual sentence embeddings based on attention model. In: 54th Annual Meeting of the Association for Computational Linguistics. pp. 505–515 (2016)
48. Wang, Y., Huang, M., Zhu, X., Zhao, L.: Attention-based lstm for aspect-level sentiment classification. In: Conference on Empirical Methods in Natural Language Processing. pp. 606–615 (2017)
49. Wiseman, S., Shieber, S., Rush, A.: Challenges in data-to-document generation (2017)
50. Yang, Z., Hu, Z., Deng, Y., Dyer, C., Smola, A.: Neural machine translation with recurrent attention modeling. arXiv preprint arXiv:1607.05108 (2016)
51. Yong, Z., Wang, Y., Liao, J., Xiao, W.: A hierarchical attention seq2seq model with copynet for text summarization. In: 2018 International Conference on Robots & Intelligent System (ICRIS). pp. 316–320 (2018)