Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI-20)
Journal Track

Point at the Triple: Generation of Text Summaries from Knowledge Base Triples (Extended Abstract)*

Pavlos Vougiouklis[1†], **Eddy Maddalena**[2], **Jonathon Hare**[3] and **Elena Simperl**[2]
[1]Huawei Technologies, Edinburgh, United Kingdom
[2]King's College London, London, United Kingdom
[3]University of Southampton, Southampton, United Kingdom
pavlos.vougiouklis@huawei.com, {eddy.maddalena, elena.simperl}@kcl.ac.uk, jsh2@ecs.soton.ac.uk

Abstract

We investigate the problem of generating natural language summaries from knowledge base triples. Our approach is based on a pointer-generator network, which, in addition to generating regular words from a fixed target vocabulary, is able to verbalise triples in several ways. We undertake an automatic and a human evaluation on single and open-domain summaries generation tasks. Both show that our approach significantly outperforms other data-driven baselines.

Triples	Atlas_Shrugged literaryGenre Science_fiction Atlas_Shrugged country United_States John_Galt series Atlas_Shrugged Atlas_Shrugged publicationYear ''1957'' Atlas_Shrugged author Ayn_Rand
Text Summary	Atlas Shrugged is a science fiction novel by Ayn Rand.

Table 1: A simplified NLG example. Our systems generates a text summary from a set of un-ordered triples about *Atlas Shrugged*.

1 Introduction

Natural Language Generation (NLG) is the task of generating text that captures the content of structured-data records in a human-readable way [Reiter and Dale, 2000]. In the context of knowledge graphs, structured data takes the form of triples.

NLG systems for knowledge graphs take as input a subset of the *graph's triples* and output a *text summary* [Bouayad-Agha *et al.*, 2014]. Earlier attempts to generate text from triples tend to use rules or template-based techniques, which work well in domains with a regular structure and limited vocabulary [Bouayad-Agha *et al.*, 2012]. More recent proposals leverage deep learning, which was successful in similar text-generative tasks, including machine translation [Cho *et al.*, 2014; Sutskever *et al.*, 2014; Bahdanau *et al.*, 2014] and text summarisation [Rush *et al.*, 2015; See *et al.*, 2017]. Systems such as [Chisholm *et al.*, 2017; Lebret *et al.*, 2016; Vougiouklis *et al.*, 2018] are able to generate coherent, relevant text for Wikipedia from structured data, without any input from linguists or domain experts. However, these newer systems are not without their challenges: while they could, in principle, scale to open-domain tasks, they only report their performance on one domain, people's biographies. In addition, they struggle to verbalise rare or previously unseen entities, which are represented as *placeholder* tokens in the output and are meant to be replaced in a post-processing step. This introduces a degree of stochastic behaviour when multiple relations from the input match the predicted placeholders.

In this paper, we present an approach that addresses these concerns. Table 1 presents a canonical NLG task. Our aim is to automatically generate a textual summary describing the graph about *Atlas Shrugged*. The input contains triples in which the given entity, in this case `Atlas_Shrugged`, is the subject or the object of the triples. Our approach is inspired by *pointer-generator* networks which have been recently introduced in text summarisation [Gu *et al.*, 2016; See *et al.*, 2017]. Our systems jointly learn to: (i) verbalise the entities from *pointed* triples in several ways; (ii) copy the label or the number in the case that the pointed triple consists of either infrequent entities or numbers; and (iii) generate words or other human-readable realisations of entities from a fixed vocabulary.

Following the methodology of Vougiouklis *et al.*, we create a dataset encompassing the entirety of Wikipedia rather than just the biographies. We use this dataset to demonstrate our model's ability to generalise on a much more challenging task. We evaluate our approach in two ways: automatically and manually. For the former, we use BLEU, ROUGE and METEOR in order to evaluate the performance of our approach in both the widely cited task of biographies generation [Chisholm *et al.*, 2017; Lebret *et al.*, 2016; Liu *et al.*, 2018; Vougiouklis *et al.*, 2018; Yeh *et al.*, 2018], and the generation of open-domain Wikipedia summaries. Furthermore, we run a user study in which we explore the *fluency* and *coverage* of the summaries, as well as the presence of *contradictions*. In all scenarios, our systems outperform a variety of competing baselines of different natures. Our dataset along with the code of our systems is available at: `github.com/pvougiou/Point-at-the-Triple`.

*This paper is an extended abstract of an article entitled "Point at the Triple: Generation of Text Summaries from Knowledge Base Triples" in the *Journal of Artificial Intelligence Research*.

†Work done while at University of Southampton.

2 Our Model

We assume our model is trained on records consisting of an English language summary and a set of triples. Let $F_z = \{f_1, \ldots, f_E : f_i = (s_i, p_i, o_i)\}$ be the set of triples $f_1, \ldots, f_E$ about the entity z, where s_i, p_i and o_i are the one-hot vector representations of the respective subject, predicate and object of the i-th triple. We build a model that computes the probability of generating a sequence of tokens $\mathbf{y} = y_1, y_2, \ldots, y_T$, given the initial set of triples $f_1, f_2, \ldots, f_E$.

We build upon architectures from the literature, in particular See *et al.* and our own work in Vougiouklis *et al.*. The former introduces a pointer-generator network capable of both copying tokens from the input sequence and generating words for the fixed vocabulary of the decoder. While this model handles sequential inputs and outputs (i.e. sentences), in our case the sets of input triples are un-ordered, and not sequentially correlated. For this reason, we use the encoder proposed in the latter work of Vougiouklis *et al.*, and we compute the attention scores on top of this feed-forward architecture.

In many cases, the entities that participate in the properties contained in the input triples cannot be directly copied to the generated text. For example, the entities of `dbr:Actor` and `dbr:United_States` could be expressed, based on the context, as both "actor" or "actress", and "United States" and "American", respectively. To tackle this, we propose a technique that enables our model to learn different realisations for the entities of the pointed triples. The technique also helps with handling rare entities, for which we do not have good vector representations, numerical values that are in the third, object position of the pointed triples, and their labels.

Decoder. We implement the decoder as a multi-gated RNN variant with Gated Recurrent Units (GRUs). Let $h_t^l \in \mathbb{R}^m$ be the aggregated output of a hidden unit at timestep $t = 1 \ldots T$ and layer depth $l = 1 \ldots L$.

2.1 Triple Encoder

We compute the vector representation h_{f_i} of the i-th triple by forward propagating the triple encoder as follows:

$$\tilde{h}_{f_i} = [\mathbf{W}_{\mathbf{x}\to\tilde{\mathbf{h}}} s_i; \mathbf{W}_{\mathbf{x}\to\tilde{\mathbf{h}}} p_i; \mathbf{W}_{\mathbf{x}\to\tilde{\mathbf{h}}} o_i] \ , \tag{1}$$

$$h_{f_i} = \text{ReLU}(\mathbf{W}_{\tilde{\mathbf{h}}\to\mathbf{h}} \tilde{h}_{f_i}) \ , \tag{2}$$

where $[\ldots;\ldots]$ represents vector concatenation, $\mathbf{W}_{\mathbf{x}\to\tilde{\mathbf{h}}} : \mathbb{R}^{|N|} \to \mathbb{R}^m$ and $\mathbf{W}_{\tilde{\mathbf{h}}\to\mathbf{h}} : \mathbb{R}^{3m} \to \mathbb{R}^m$ are unbiased linear mappings.

Attending the Triples. We implement an attention mechanism over all triples, based on Dong and Lapata's formulation. Our model uses the attention scores to compute a context vector, c_t, as a weighted sum over the representations of the encoded triples. The alignment between this context vector and the information that has already been processed in a generated summary are jointly learned through trainable weights $\mathbf{W_c} : \mathbb{R}^m \to \mathbb{R}^m$ and $\mathbf{W_h} : \mathbb{R}^m \to \mathbb{R}^m$ as follows:

$$h_t^{L+1} = \tanh\left(\mathbf{W_c} c_t + \mathbf{W_h} h_t^L\right) \ . \tag{3}$$

2.2 Dynamically Expanding the Vocabulary

As discussed earlier, the pointer-generator network from See *et al.* learns to copy only a single representation per input token. This means that the system uses the same label for each copied entity regardless of the context in the text, which impacts fluency [Vougiouklis *et al.*, 2018]. Our approach addresses this concern by learning multiple realisations for the entity of a pointed triple.

Our approach is partially inspired by how humans would perform on the same task. When provided with a set of triple-facts which they are asked to summarise in text, people would start summarising by using their own known vocabulary. However, they would focus their attention on a particular triple when they would want to realise an entity's name or a number in the text.

We define $G^{k_d} = \{g_1^{k_d}, g_2^{k_d}, \ldots, g_Q^{k_d}\}$ as the set of all possible verbalisations through which our model learns to express an entity k_d in the generated summary. Q is a dataset-specific hyper-parameter for our model, and is calculated by averaging the number of possible realisations q^{k_d} $\forall k_d \in K$.

Let $H^{(f)}$ and $H^{(i)}$ be the sets of all the frequent and infrequent entities that participate in the triples. In addition, let $\mathcal{E} = \{e_1, e_2, \ldots, e_E\}$ s.t. $e_j \in (s_j, o_j)$ and $e_j \neq z$ $\forall j \in [1, E]$ be the set of all the items (numerical values or entities) other than entity z that participate in the corresponding relationships in F. We assume a fixed target vocabulary $V^\dagger$. In comparison to similar pointer-generator networks that expand the decoder's fixed vocabulary by the length of their input E, we expand it by $Q \cdot E$, and we define the dynamic vocabulary extension (where the values are based on the input triples), $V^{\text{ext}} = \{v_1^{\text{ext}}, v_2^{\text{ext}}, \ldots, v_{Q\cdot E}^{\text{ext}}\}$ along with its subsets V_{ext}^f, $V_{\text{ext}}^{\texttt{copy}}$ and $V_{\text{ext}}^{\texttt{null}}$, s.t.:

$$v_j^{\text{ext}} = \begin{cases} g_{j\%Q}^{e_{\lceil j/Q \rceil}} \in V_{\text{ext}}^f & e_{\lceil j/Q \rceil} \in H^{(f)} \\ g_1^{e_{\lceil j/Q \rceil}} \in V_{\text{ext}}^{\texttt{copy}} & e_{\lceil j/Q \rceil} \in H^{(i)} \text{ and } j\%Q = 1 \\ g_1^{e_{\lceil j/Q \rceil}} \in V_{\text{ext}}^{\texttt{copy}} & e_{\lceil j/Q \rceil} \in \mathbb{R} \text{ and } j\%Q = 1 \\ \texttt{null} \in V_{\text{ext}}^{\texttt{null}} & \text{otherwise} \end{cases} \tag{4}$$

$\forall j \in [1, Q \cdot E]$, where $\lceil \ldots \rceil$ represents the ceiling function.

During both training and testing, for each set of input triples we form the values of V^{ext}. Each triple is provided with Q slots in V^{ext}. In case a rare entity is either the subject or the object of a triple in the triple set, it is replaced by its corresponding instance type token before it is provided to our model. In such scenario, all values of V^{ext} that correspond to this particular triple are filled with `null`, except the first one which refers to the copy of the label of this rare entity. A similar methodology is used for numbers.

2.3 Summarising By Pointing and Generating

The probability distribution q_t for each entry in the vocabulary extension V^{ext} after distributing the attention scores over the realisations of the relevant triples is computed as follows:

$$\widetilde{q}_t^{(i)} = \begin{cases} \exp[a_t^{(\lceil i/Q \rceil)}] & v_i^{\text{ext}} \in V_{\text{ext}}^f \cup V_{\text{ext}}^{\texttt{copy}} \\ 0 & v_i^{\text{ext}} \in V_{\text{ext}}^{\texttt{null}} \end{cases} \tag{5}$$

$$q_t^{(i)} = \frac{\widetilde{q}_t^{(i)}}{\sum_{j=1}^{Q\cdot E} \widetilde{q}_t^{(j)}} \ . \tag{6}$$

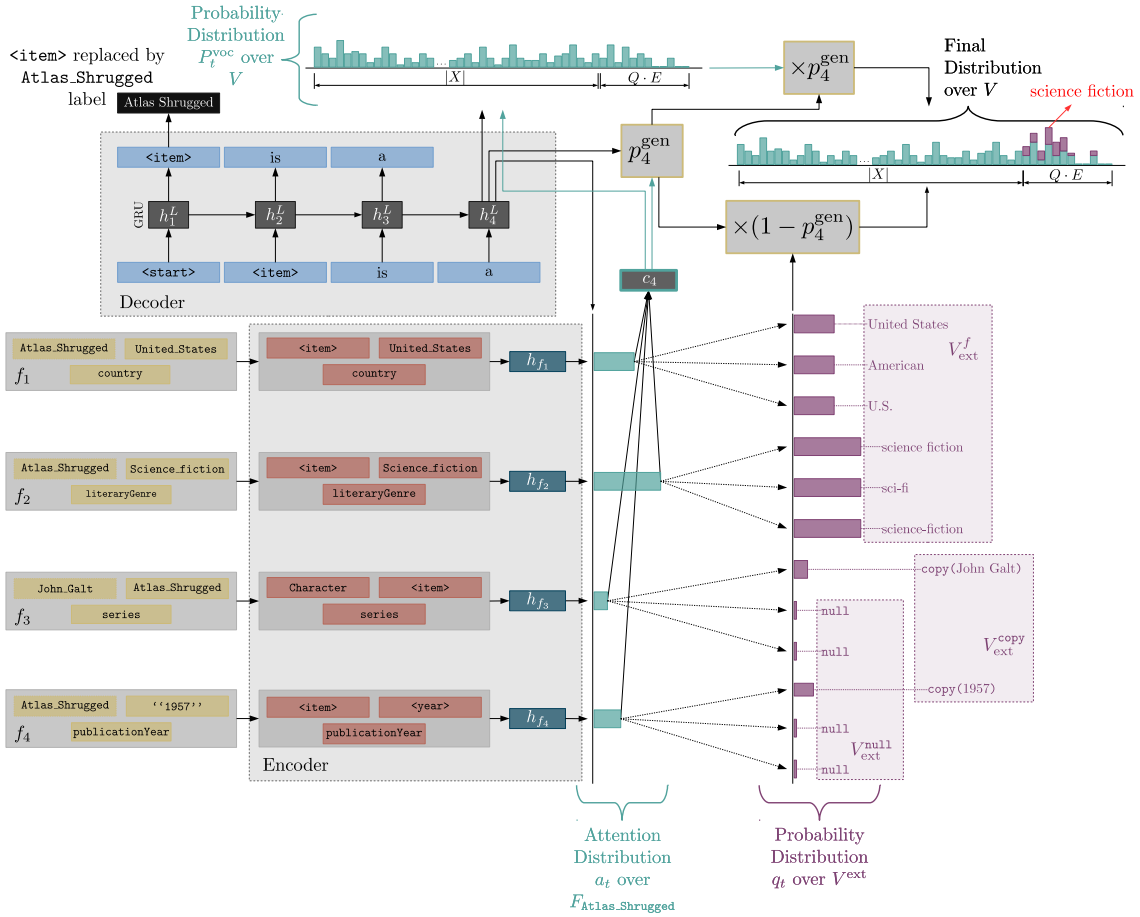

Figure 1: At $t = 4$, p_4^{gen} weighs the probability of copying a word from V^{ext} higher than generating a word from the fixed vocabulary $V^{\dagger}$. The decoder learns to interpret the weighted sum of h_4^L and c_4 in order to compute a probability distribution for the most appropriate text realisation given the context of the triples. The attention mechanism highlights f_2 as the most important triple for the generation of the upcoming token. The attention scores are distributed among the entries of V^{ext}, and accumulated into the final distribution over V. As a result, the model copies "science fiction", one of the surface forms associated with f_2.

$\forall i \in [1, Q \cdot E]$. We adopt the notion of *generation probability* $p_t^{\text{gen}} \in [0, 1]$, to simulate a soft switch at each timestep t between generating a token from the fixed vocabulary or copying either the surface form or the label of an entity from the highlighted triple [See *et al.*, 2017]. Our model computes the following probability distribution for each entry w in the extended vocabulary $V = V^{\dagger} \cup V^{\text{ext}}$ as follows:

$$P_t(w) = \begin{cases} p_t^{\text{gen}} P_t^{\text{voc}}(w) + \\ (1 - p_t^{\text{gen}}) q_t^{(w)} & w \in V_{\text{ext}}^{f} \cup V_{\text{ext}}^{\texttt{copy}} \\ p_t^{\text{gen}} P_t^{\text{voc}}(w) & w \in V^{\dagger} \\ 0 & w \in V_{\text{ext}}^{\texttt{null}} \end{cases}, \quad (7)$$

where $P_t^{\text{voc}} = \text{softmax}(\mathbf{W}_{\mathbf{y}} h_t^{L+1})$, and $\mathbf{W}_{\mathbf{y}} : \mathbb{R}^m \rightarrow \mathbb{R}^{|X|+Q \cdot E}$ is a trainable weight matrix.

3 Experiments

We trained and evaluate our system on two corpora. The first is the D1 Biographies corpus provided by Vougiouklis *et al.*, which consists of triples from DBpedia aligned with Wikipedia biographies. The second corpus, which we refer to as the Full corpus, uses the same methodology as in [Vougiouklis *et al.*, 2018], applied, however, to the entire Wikipedia. We compute a Q value of 2 (for the D1 Biographies dataset), and 3 (for the Full corpus), which results in $\sim 98\%$ coverage of the total number of textual realisations of the triples' entities for both corpora (see Section 2.2).

For each dataset we ran two sets of experiments: one in which the surface form tuples [Vougiouklis *et al.*, 2018] were part of the fixed vocabulary of the decoder, and one in which they were treated as regular words. We evaluated our approach by comparing it against a set of competitive baselines: (i) Random, (ii) Kneser-Ney (KN) language model, (iii) Information Retrieval (IR), (iv) Triples2GRU and Triples2LSTM from Vougiouklis *et al.*, and (v) Pointer-Generator, which has been adapted from See *et al.*.

3.1 Automatic Evaluation

We evaluated our performance in terms of the following three metrics: (i) BLEU (Bilingual Evaluation Understudy),

(ii) ROUGE (Recall-Oriented Understudy for Gisting Evaluation), and (iii) METEOR.

In almost all scenarios, we outperformed the baselines. On the Full corpus, both our systems achieved an improvement that ranges from 0.59 to 0.79 and 0.16 to 0.47 BLEU 4 and ROUGE points, respectively, in comparison to Pointer-Generator, which is our strongest competitor. On the Biographies corpus, we can show small improvements of at least 0.02 in terms of both BLEU and ROUGE points. We believe that the lower performance gain on the smaller corpus is mainly a function of the limited linguistic variability of biographies—in 92.67% of the cases, entities from the triples in that corpus were realised in the text with their most frequent surface form. The advantages of our approach become more clear on the Full corpus, which is linguistically more challenging and includes entities with varied realisations.

3.2 Human Evaluation

The three metrics we used in the automatic evaluation do not capture performance well in tasks where the input and the output are loosely correlated [Reiter, 2010]. To understand how well our approach would do in practice, we undertook two user studies with participants recruited from the Figure Eight crowdsourcing platform. In the first one, we looked at the performance of our networks against the two most competitive baselines, Triples2GRU and Pointer-Generator, on the open-domain task. In the second one, we explored whether training our systems on the Full dataset with the same hyper-parameters (except the size of the input/output vocabularies) would yield results comparable to systems that were trained on the biographies task.

Inputs and Outputs

We included only input sets of at least six triples. For the first study we sampled 32 sets of triples according to the instance-type distribution of the main discussed entities in the Full dataset. For the second study, we used a random sample of the same size. In both cases, we took the summaries generated by the four systems and asked 10 participants to assess them against three criteria: (i) *fluency*, (ii) *coverage*, which is concerned with the triples in the input whose content is mentioned either implicitly or explicitly in the text, and (iii) *contradiction*, which refers to information that is conveyed by the sentence, but conflicts with one or more of triples from the input set.

For each of the 32 sets of triples used in each study, we also compiled a gold standard of 8 additional sets of triples with manually written summaries. We designed three crowdsourcing tasks, one for fluency, one for coverage, and one for contradiction.

Results of the First Study

The results of the two studies are in alignment with the results of the automatic evaluation.

Fluency. In the first study, which focused on the performance on the Full dataset, the summaries generated by both our systems were ranked significantly higher than those of the the two baselines (one-way ANOVA test, $p < .05$). Among our two systems, the one with surface form tuples was also found to generate significantly more fluent summaries than the one without.

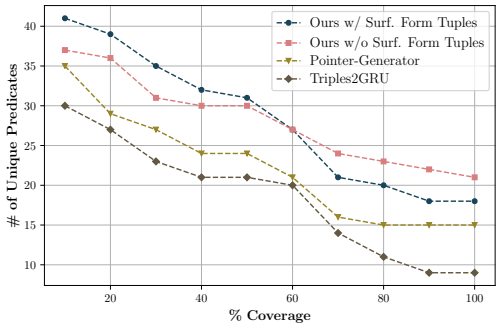

Figure 2: The percentage with which the unique predicates from the input triples are covered in the summaries.

Coverage. Both our systems realised more information from the input triples in the generated summaries; the coverage of both systems is also significantly better than Triples2GRU. Figure 2 shows the extent to which different number of predicates from the input triples are covered in the text. Our architectures are able to realise not only more predicates than the competition but also to address these predicates with higher consistency in the summaries.

Contradiction. All systems scored low with respect to the amount of information in the text that is contradicted by the triples, and no statistically significant differences were noted.

Results of the Second Study

Training our systems on a much more challenging corpus results in minimal performance differences in comparison to the performance of the same systems when trained solely on Biographies. Nonetheless, we observed that when trained on the Full corpus and tested on biographies, the system equipped with surface form tuples is significantly ($p < .05$) more fluent than the one without.

4 Conclusion

We presented a data-driven approach to generate open-domain text summaries from knowledge base triples. We proposed a pointer-generator network that jointly learns to verbalise in a different number of ways the content from the triples, while retaining the ability to generate regular words from a fixed target vocabulary. We trained and evaluated two system variants on two different datasets of aligned DBpedia triples with Wikipedia summaries with respective vocabulary sizes of 400k and 1114k words.

We evaluated our approach using well-established automatic text similarity metrics and conducted two user studies. Our approach outperformed the state of the art; in particular, compared to other encoder-decoder architectures, our summaries are significantly more fluent and convey a greater share of the content of the input triples.

Acknowledgments

This research is partially supported by the Data Stories project, funded by EPSRC research grant No. EP/P025676/1.

References

[Bahdanau *et al.*, 2014] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

[Bouayad-Agha *et al.*, 2012] Nadjet Bouayad-Agha, Gerard Casamayor, Simon Mille, and Leo Wanner. Perspective-oriented generation of football match summaries: Old tasks, new challenges. *ACM Trans. Speech Lang. Process.*, 9(2):3:1–3:31, August 2012.

[Bouayad-Agha *et al.*, 2014] Nadjet Bouayad-Agha, Gerard Casamayor, and Leo Wanner. Natural language generation in the context of the semantic web. *Semantic Web*, 5(6):493–513, 2014.

[Chisholm *et al.*, 2017] Andrew Chisholm, Will Radford, and Ben Hachey. Learning to generate one-sentence biographies from Wikidata. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 633–642, Valencia, Spain, April 2017. Association for Computational Linguistics.

[Cho *et al.*, 2014] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics.

[Dong and Lapata, 2016] Li Dong and Mirella Lapata. Language to logical form with neural attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33–43, Berlin, Germany, August 2016. Association for Computational Linguistics.

[Gu *et al.*, 2016] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, Berlin, Germany, August 2016. Association for Computational Linguistics.

[Lebret *et al.*, 2016] Rémi Lebret, David Grangier, and Michael Auli. Neural text generation from structured data with application to the biography domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1203–1213. Association for Computational Linguistics, 2016.

[Liu *et al.*, 2018] Tianyu Liu, Kexiang Wang, Lei Sha, Baobao Chang, and Zhifang Sui. Table-to-text generation by structure-aware seq2seq learning, 2018.

[Reiter and Dale, 2000] Ehud Reiter and Robert Dale. *Building Natural Language Generation Systems*. Cambridge University Press, New York, NY, USA, 2000.

[Reiter, 2010] Ehud Reiter. *Natural Language Generation*, chapter 20, pages 574–598. Wiley-Blackwell, 2010.

[Rush *et al.*, 2015] Alexander M. Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[See *et al.*, 2017] Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083. Association for Computational Linguistics, 2017.

[Sutskever *et al.*, 2014] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc., 2014.

[Vougiouklis *et al.*, 2018] Pavlos Vougiouklis, Hady Elsahar, Lucie-Aimée Kaffee, Christophe Gravier, Frédérique Laforest, Jonathon Hare, and Elena Simperl. Neural wikipedian: Generating textual summaries from knowledge base triples. *Journal of Web Semantics*, 2018.

[Yeh *et al.*, 2018] S. Yeh, H. Huang, and H. Chen. Precise description generation for knowledge base entities with local pointer network. In *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pages 214–221, Dec 2018.