Capturing Entity Hierarchy in Data-to-Text Generative Models

Clément Rebuffel
Sorbonne Université, CNRS, LIP6, F-75005 Paris, France
clement.rebuffel@lip6.fr

Laure Soulier
Sorbonne Université, CNRS, LIP6, F-75005 Paris, France
laure.soulier@lip6.fr

Geoffrey Scoutheeten
BNP Paribas, France
geoffrey.scoutheeten@bnpparibas.com

Patrick Gallinari
Sorbonne Université, CNRS, LIP6, F-75005 Paris, France
Criteo AI Lab, Paris
patrick.gallinari@lip6.fr

ABSTRACT

We aim at generating summary from structured data (i.e. tables, entity-relation triplets, ...). Most previous approaches relies on an encoder-decoder architecture in which data are linearized into a sequence of elements. In contrast, we propose to take into account entities forming the data structure in a hierarchical model. Moreover, we introduce the Transformer encoder in data-to-text models to ensure robust encoding of each element/entities in comparison to all others, no matter their initial positioning. Our model is evaluated on the RotoWire benchmark (statistical tables of NBA basketball games). This paper has been accepted at ECIR 2020.

KEYWORDS

Data-to-Text, Hierarchical Encoding, Deep Learning

1 CONTEXT AND MOTIVATION

Understanding data structure is an emerging challenge to enhance textual tasks, such as question answering [11, 18] or table retrieval [4, 17]. One emerging research field, referred to as "data-to-text" [5], consists in transcribing data-structures into natural language in order to ease their understandablity and their usablity. Numerous examples of applications can be cited: journalism [10], medical diagnosis [12], weather reports [16], or sport broadcasting [2, 21]. Figure 1 depicts a data-structure containing statistics on NBA basketball games, paired with its corresponding journalistic description.

Until recently, efforts to bring out semantics from structured-data relied heavily on expert knowledge (e.g. rules) [3, 16]. Modern data-to-text models [1, 6, 21] leverage deep learning advances and are generally designed using two connected components: 1) an encoder aiming at understanding the structured data and 2) a decoder generating associated descriptions. This standard architecture is often augmented with 1) the attention mechanism which computes a context focused on important elements from the input at each decoding step and, 2) the copy mechanism to deal with unknown or rare words. However, most of work [1, 6, 21] represent the data records as a single sequence of facts to be fed to the encoder. These models reach their limitations on large structured-data composed of several entities (e.g. row in tables) and multiple attributes (e.g. column in tables) and fail to accurately extract salient elements.

To improve these models, a number of work [7, 13] proposed innovating decoding modules based on planning and templates, to ensure factual and coherent mentions of records in generated descriptions. Closer to our work, very recent work [8, 9, 14] have proposed to take into account the data structure. For instance, Puduppully et al. [13] design a more complex two-step decoder: they first generate a plan of elements to be mentioned, and then condition text generation on this plan.

"Copyright © 2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0)."

2 CONTRIBUTION AND MAIN RESULTS

In this paper, we focus on the encoding step of data-to-text models since we assume that a large amount of work is done in language generation and summary. We believe that the most important challenge relies here on the data structure encoding. Therefor, we identify two limitations of previous work :

(1) *Linearization of the data-structure.* In practice, most works focus on introducing innovating decoding modules, and still represent data as a unique sequence of elements to be encoded, effectively losing distinction between rows, and therefore entities. To the best of our knowledge, only Liu et al. [8, 9] propose encoders constrained by the structure but these approaches are designed for single-entity structures.
(2) *Arbitrary ordering of unordered collections in recurrent networks (RNN).* Most data-to-text systems use RNNs as encoders (such as LSTMs), which require in practice their input to be fed sequentially. This way of encoding unordered sequences (*i.e.* collections of entities) implicitly assumes an arbitrary order within the collection which, as in Vinyals et al. [20], significantly impacts the learning performance.

To address these shortcomings, we propose a new structured-data encoder assuming that structures should be hierarchically captured. Our contribution focuses on the encoding of the data-structure, thus the decoder is chosen to be a classical module as used in [13, 21]. Our contribution, illustrated in Figure 2, is threefold:

- We model the general structure of the data using a two-level architecture, first encoding all entities on the basis of their elements, then encoding the data structure on the basis of its entities;
- We introduce the Transformer encoder [19] in data-to-text models to ensure robust encoding of each element/entities in comparison to all others, no matter their initial positioning;
- We integrate a hierarchical attention mechanism to compute the hierarchical context fed into the decoder.

As shown in Figure 2, our model relies on two encoders:

• the **Low-level encoder** encodes each entity e_i on the basis of its record embeddings $\mathbf{r}_{i,j}$. Each record embedding $\mathbf{r}_{i,j}$ is compared to other record embeddings to learn its final hidden representation

TEAM	H/V	WINS	LOSSES	PTS	REB	AST	...
Hawks	H	46	12	95	42	27	...
Magic	V	19	41	88	40	22	...

PLAYER	PTS	REB	AST	STL	BLK	CITY	...
Al Horford	17	13	4	2	0	Atlanta	...
Kyle Korver	8	3	2	1	2	Atlanta	...
Jeff Teague	17	0	7	2	0	Atlanta	...
N. Vucevic	21	15	3	1	1	Orlando	...
Tobias Harris	15	4	1	2	1	Orlando	...
...	...	...	...	...	...	...	...

H/V: home or visiting; PTS: points; REB: rebounds;
AST: assists; STL: steals; BLK: blocks

The **Atlanta Hawks** (**46-12**) beat the **Orlando Magic** (**19-41**) **95-88** on Friday. **Al Horford** had a good all-around game, putting up **17 points, 13 rebounds, four assists and two steals** in a tough matchup against **Nikola Vucevic**. **Kyle Korver** was the lone Atlanta starter not to reach double figures in points. **Jeff Teague** bounced back from an illness, he scored **17 points** to go along with **seven assists and two steals**. After a rough start to the month, the **Hawks** have won three straight and sit atop the Eastern Conference with a nine game lead on the second place Toronto Raptors. The **Magic** lost in devastating fashion to the Miami Heat in overtime Wednesday. They blew a seven point lead with 43 seconds remaining and they might have carried that with them into Friday's contest against the **Hawks**. **Vucevic** led the **Magic** with **21 points and 15 rebounds**. **Aaron Gordon** (ankle) and **Evan Fournier** (hip) were unable to play due to injury. The **Magic** have four teams between them and the eighth and final playoff spot in the Eastern Conference. The **Magic** will host the Charlotte Hornets on Sunday, and the **Hawks** with take on the Heat in Miami on Saturday.

Figure 1: Example of structured data from the RotoWire dataset. Rows are entities (a team or a player) and each cell a record, its key being the column label and its value the cell content. Factual mentions from the table are boldfaced in the description.

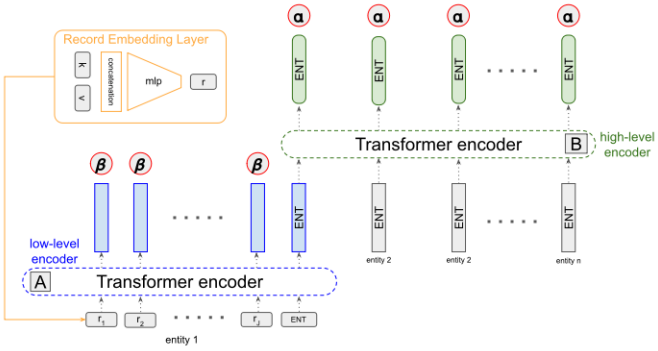

Figure 2: Our proposed hierarchical encoder. Once the records are embedded, the low-level encoder works on each entity independently (A); then the high-level encoder encodes the collection of entities (B). In circles, we represent the hierarchical attention scores: the α scores at the entity level and the β scores at the record level.

$\mathbf{h}_{i,j}$. We also add a special record [ENT] for each entity, illustrated in Figure 2 as the last record. Since entities might have a variable number of records, this token allows to aggregate final hidden record representations $\{\mathbf{h}_{i,j}\}_{j=1}^{J_i}$ in a fixed-sized representation vector $\mathbf{h}_i$.

• the **High-level encoder** encodes the data-structure on the basis of its entity representation $\mathbf{h}_i$. Similarly to the **Low-level encoder**, the final hidden state $\mathbf{e_i}$ of an entity is computed by comparing entity representation $\mathbf{h}_i$ with each others. The data-structure representation $\mathbf{z}$ is computed as the mean of these entity representations, and is used for the decoder initialization.

We report experiments on the RotoWire benchmark [21] which contains around $5K$ statistical tables of NBA basketball games paired with human-written descriptions. Comparisons against baselines show that introducing the Transformer architecture is a promising way to implicitly account for data structure, and leads to better content selection even before introducing hierarchical encoding. Furthermore, our hierarchical model outperforms all baselines on content selection, showing that capturing structure in the encoding process is more effective that predicting a structure in the decoder (e.g., planning or templating). We show via ablation studies that further constraining the encoder on structure (through hierarchical attention) leads to even better performances.

For a more in-depth understanding of our contribution, please read our ECIR paper [15].

3 ACKNOWLEDGEMENTS

We would like to thank the H2020 project AI4EU (825619) which partially supports Laure Soulier and Patrick Gallinari.

REFERENCES

[1] Shubham Agarwal and Marc Dymetman. 2017. A surprisingly effective out-of-the-box char2char model on the E2E NLG Challenge dataset. In *SIGdial*. 158–163.
[2] David L. Chen and Raymond J. Mooney. 2008. Learning to Sportscast: A Test of Grounded Language Acquisition *(ICML 2008)*.
[3] Dong Deng, Yu Jiang, Guoliang Li, Jian Li, and Cong Yu. 2013. Scalable column concept determination for web tables using large knowledge bases. *Proceedings of the VLDB Endowment* (2013).
[4] Li Deng, Shuo Zhang, and Krisztian Balog. 2019. Table2Vec: Neural Word and Entity Embeddings for Table Population and Retrieval. In *SIGIR 2019*.
[5] Albert Gatt and Emiel Krahmer. 2018. Survey of the State of the Art in Natural Language Generation: Core Tasks, Applications and Evaluation. *J. Artif. Int. Res.* (2018).
[6] Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural Text Generation from Structured Data with Application to the Biography Domain. In *EMNLP*.
[7] Liunian Li and Xiaojun Wan. 2018. Point Precisely: Towards Ensuring the Precision of Data in Generated Texts Using Delayed Copy Mechanism. In *ICCL*.
[8] Tianyu Liu, Fuli Luo, Qiaolin Xia, Shuming Ma, Baobao Chang, and Zhifang Sui. 2019. Hierarchical Encoder with Auxiliary Supervision for Neural Table-to-Text Generation: Learning Better Representation for Tables. *AAAI* (2019).
[9] Tianyu Liu, Kexiang Wang, Lei Sha, Baobao Chang, and Zhifang Sui. 2018. Table-to-text Generation by Structure-aware Seq2seq Learning. In *AAAI*.
[10] Will Oremus. 2014. The First News Report on the L.A. Earthquake Was Written by a Robot.
[11] Panupong Pasupat and Percy Liang. 2015. Compositional Semantic Parsing on Semi-Structured Tables. In *IJCNLP*.
[12] Steffen Pauws, Albert Gatt, Emiel Krahmer, and Ehud Reiter. 2019. *Making Effective Use of Healthcare Data Using Data-to-Text Technology: Methodologies and Applications*. 119–145.
[13] Ratish Puduppully, Li Dong, and Mirella Lapata. 2018. Data-to-Text Generation with Content Selection and Planning. In *AAAI*.
[14] Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-text Generation with Entity Modeling. In *ACL 2019*.
[15] Clément Rebuffel, Laure Soulier, Geoffrey Scoutheeten, and Patrick Gallinari. 2020. A Hierarchical Model for Data-to-Text Generation. In *ECIR 2020*. 65–80.
[16] Ehud Reiter, Somayajulu Sripada, Jim Hunter, Jin Yu, and Ian Davy. 2005. Choosing Words in Computer-generated Weather Forecasts. *Artif. Intell.* (2005).
[17] Anish Das Sarma, Lujun Fang, Nitin Gupta, Alon Y. Halevy, Hongrae Lee, Fei Wu, Reynold Xin, and Cong Yu. 2012. Finding Related Tables. In *SIGMOD*.
[18] Huan Sun, Hao Ma, Xiaodong He, Wen-tau Yih, Yu Su, and Xifeng Yan. 2016. Table Cell Search for Question Answering. In *Proceedings of the 25th International Conference on World Wide Web - WWW '16*. ACM Press, 771–782.
[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Kaiser, and Illia Polosukhin. [n.d.]. Attention is All You Need *(NIPS 2017)*.
[20] Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. 2016. Order matters: Sequence to sequence for sets. In *ICLR*.
[21] Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in Data-to-Document Generation. In *EMNLP*.