Studying the Impact of Filling Information Gaps on the Output Quality of Neural Data-to-Text

Craig Thomson, Zhijie Zhao, and Somayajulu Gowri Sripada

Department of Computing Science, University of Aberdeen, UK:
{c.thomson, z.zhao1.19, yaji.sripada}@abdn.ac.uk

Abstract

It is unfair to expect neural data-to-text to produce high quality output when there are gaps between system input data and information contained in the training text. Thomson et al. (2020) identify and narrow information gaps in Rotowire, a popular data-to-text dataset. In this paper, we describe a study which finds that a state-of-the-art neural data-to-text system produces higher quality output, according to the information extraction (IE) based metrics, when additional input data is carefully selected from this newly available source. It remains to be shown, however, whether IE metrics used in this study correlate well with humans in judging text quality.

1 Introduction

The ecological validity (de Vries et al., 2020) of data-to-text tasks requires that tasks resemble, as closely as possible, real-world problems. Only if this is the case can neural data-to-text solutions be operationally deployed with confidence. In the context of data-to-text, one of the issues with ecological validity is that most real-world tasks involve sizeable input data, with longer and more complex texts than are found in 'toy-sized' datasets. We must be able to see a path to a machine learning task which closely resembles a real-world scenario and allows us to investigate important research questions. We should aim to improve both the dataset and the task continuously. Generating summaries of basketball games from tabled data with the Rotowire dataset, as introduced by Wiseman et al. (2017a) for the English language, moves us closer to an ecologically valid data-to-text task.

The original Rotowire dataset has been found to contain gaps between the information in the input data, and the information content of the training text. This makes the task unfair for evaluating neural data-to-text systems that are required to generate high quality output text, with hardly any factual errors. The SportSett:Basketball dataset (Thomson et al., 2020) addresses these data issues by fixing information gaps in the input data, whilst maintaining the original human-authored texts. Given that there is now at least an order of magnitude more data per game, we should consider which subset of data to train the system on, and what if any pre-processing should be preformed.

We added some of this newly available data to an existing state-of-the-art neural data-to-text system (Rebuffel et al., 2020) and found improvement across a range of metrics. We used this system since, at the time of writing, it is one of the most recent, best performing and easiest to configure. We also discuss here which types of data could be added in the future, as well as some difficulties that may be encountered in doing so.

2 Related Work

Many systems have been designed and evaluated using the Rotowire dataset (Wiseman et al., 2017a; Puduppully et al., 2019a,b; Wang, 2019; Gong et al., 2019; Iso et al., 2019; Rebuffel et al., 2020). Most of these works focus on adjusting the architecture of the system whilst using similar input data. Gong et al. (2019) is a notable exception, as their architecture change allows box score and other data from previous games in their input. This was, however, still data from the original Rotowire dataset.

Some works have attempted to better align data to text in other datasets, with techniques such as semantic control (Dušek et al., 2019). For Rotowire, this has been investigated by Wang (2019), which aims to prevent generation of sentences not grounded in the data. With this approach, some of the most common sentence types from the human narrative, such as the subsequent opponents

Proceedings of The 13th International Conference on Natural Language Generation, pages 35–40,
Dublin, Ireland, 15-18 December, 2020. ©2020 Association for Computational Linguistics

of each team, are not generated. This difference is crucial when determining ecological validity of the task. The aim is to replicate the human author in the act of writing a summary of the basketball game, including its full narrative structure.

3 Information Gaps in Data-to-Text

Shown in Figure 1 is an example textual summary from the Rotowire dataset. An example partial box score is shown in Table 1. There are numerous cases where information conveyed by the text is not present in the same form in the box score or other game data. These information gaps should be investigated in order to improve the machine learning task.

We performed a machine-assisted corpus analysis, using the spaCy syntactic parser (Honnibal, 2015) to group sentences which only differ by entity. We do this using an abstraction process where we replace named entities with special tokens comprised of their part-of-speech and entity label. Some manual rules are added to the parser to handle domain specific syntax. By this process the sentence (S01 from Figure 1) *'The Atlanta Hawks (41-9) beat the Washington Wizards (31-19) 105-96 on Wednesday.'*, is transformed to *'PROPN-ORG (X-Y) beat the PROPN-ORG (X-Y) X-Y on NOUN-DATE.'*.

We then count and read these abstract sentence types to find statements common to the narrative, but with attribute types not present in the data. For example, sentences with the same abstract form as S1 occur 26 times in the training corpus, with more than 800 further sentences of a similar form (using defeat instead of beat, or also including the location/stadium). It is the most common type of thing to say in the opening sentence of these summaries (which teams played, when, and where). There are, however, important attributes in these sentences which are not provided by the original Rotowire data. When generating game summaries, systems will often hallucinate these attributes as they deem it probable that such language is included in the summary, but the attribute is not available to the copy attention mechanism. In the case of our above example, the day of the week is not available in the data. The stadium in which the game was played, as well as the city and/or state within which the stadium stands, are also not available despite being common in variants of this opening sentence.

3.1 Missing Game Information

In S02 and S11 from Figure 1 we notice that the games being discussed are not the game being summarised, they are previous or subsequent games for these teams. This is common in the training corpus as well. Handling data for previous games is complex (see subsection 3.2). However, data for the subsequent game can be easily obtained provided that a yearly partition scheme like that proposed by Thomson et al. (2020) is used. If such a partition scheme is not used, we cannot guarantee that a previous or subsequent game was not used to condition the system during training.

We also see in S09, a mention of the conference/division structure of the league. These are known sets, which can change over time but are fixed within a season. In the NBA there are 2 conferences, each with 3 divisions of 5 teams.

The hierarchical encoder of Rebuffel et al. (2020) takes as input a set of entities, where each entity is a set of 24 tuples[1], and each tuple describes an attribute of that entity. An example entity would be a PLAYER, which might have attributes such as 'NAME—Kyrie_Irving', 'POINTS—30', 'TEAM—Celtics', and 'REB—8'. If there are not 24 attributes of the entity then it is padded with 'NULL—NULL' tuples.

To model attributes of the current and subsequent games, we include in our input data an additional entity of type GAME, as well as two additional entities of type NEXT-GAME. These entities were chosen because our machine-assisted corpus analysis highlighted that sentences about the game date and location, as well as those for upcoming games for each team, were common in the human-authored texts, but not supported by the original Rotowire data. The newly available data from SportSett allowed us to fill these gaps. In the two NEXT-GAME entities (one each for the two teams which are the focus of the current game summary), we include attributes for season, month, day of the week, stadium, capacity, and finally both team names plus their respective division and conference names. For the GAME entity, we include the same attributes as for NEXT-GAME, plus the attendance for the game (which was obviously not available for NEXT-GAME as those events have yet to take place).

[1] This is configurable in the encoder of Rebuffel et al. (2020), although we did not change it.

S01:	The Atlanta Hawks (41-9) beat the Washington Wizards (31-19) 105-96 on Wednesday.
S02:	The Hawks bounced back after losing their first game of 2015, a 115-100 loss at the hands of the New Orleans Pelicans on Monday.
S03:	Jeff Teague was Atlanta's top scorer against the Wizards, recording 26 points on 9-of-13 shooting from the field. Kyle Korver was kept in check with just six points in a team-high 37 minutes.
S04:	He helped get his teammates involved as he dished out six assists.
S05:	Al Horford has put up at least 20 points and 10 rebounds in three of his last five games.
S06:	The Wizards have now lost four straight, which is their longest losing streak of the season.
S07:	They have lost by single-digits in all four games, and are now 0-3 against Atlanta this season.
S08:	John Wall led Washington with 24 points and nine assists. Bradley Beal was coming off an 18-point, 11-rebound effort against Charlotte on Monday.
S09:	He proceeded to post 23 points in 39 minutes in this matchup of two of the Eastern Conference's top three teams.
S10:	Washington did a great job slowing down Korver, but it wasn't enough to get them the win.
S11:	Washington will take their losing streak to Charlotte on Thursday, while the Hawks will welcome the Golden State Warriors to town Friday.

Figure 1: Example human-authored basketball summary for WAS@ATL on February 4th 2015. Summary is presented as an ordered list of sentences for ease of reference.

Player	MP	FG	FGA	FG%	3P	3PA	3P%	FT	FTA	FT%	REB	AST	STL	BLK	TOV	PTS
Kyle Korver	37	1	7	.143	1	6	.167	3	3	1.000	5	6	0	0	1	6
Al Horfor	34	10	24	.417	0	0	—	1	1	1.000	13	2	1	1	1	21
Paul Millsap	33	4	7	.571	2	3	.667	1	2	.500	7	1	3	1	1	11
DeMarre Carroll	33	4	10	.400	3	7	.429	3	4	.750	2	0	1	0	1	14
Jeff Teague	31	9	13	.692	2	4	.500	6	8	.750	2	8	2	0	4	26

Table 1: Example partial box score for WAS@ATL on February 4th 2015, showing Atlanta starters. Full box scores show approx 24 players, Rotowire also includes team totals, as well as points totals for each original game period (although not overtime)

3.2 Data of Varying Forms

In subsection 3.1 we described adding data which easily fit our chosen encoder. It is worth noting, however, that there is much more data available in the SportSett database, and it can be presented to our neural system in different formats. One argument is that we should take all the atomic data, along with, perhaps, its structure, then create an encoder which accepts data in that form. The atomic entities are players, which are grouped into teams, divisions, conferences and leagues. The atomic events are plays, the act of one or more players acquiring countable statistics. See Thomson et al. (2020) Section 2.1 for a more detailed analysis of entities and dimensions in this dataset. The NLG system would then be tasked with learning both the language, as well as the underlying mathematics for the statistics.

Whilst we could include all atomic events as training data, this would greatly increase the size of each input sample. We could alternatively include carefully selected aggregated forms of data, although creating rules to determine what should be included may be time consuming, and likely domain specific. There would also be many combinations of these derived attributes. The key question is should all possible attributes which are to be realised be available to copy attention, or, should all attributes be transformable from the atomic data? An approach combining these different types of input data could also be used.

Examples of aggregated forms of data could be anything from the percentage of shots which were successful for a player/team, to the average points a player has scored over an arbitrary span of games. One common inclusion in summaries is the aggregated statistic 'double-double'[2], where players are said to have recorded double-digits in exactly two of points, rebounds, assists, blocks and steals. Mentions of previous games in the summary frequently

[2] https://en.wikipedia.org/wiki/Double-double

use phrases in the form 'X of his/their last Y'. This can be seen in S05 of Figure 1 where we learn that Al Holford has scored greater than 20 points, and recorded more than 10 rebounds in exactly 3 of his last 5 games. There would be an impractical quantity of combinations for *'X out of Y'* based statements, even if Y had a maximum of 5-10. Other similar aggregations, such as *'scored a combined 60 points over his last 3 games.'* compound this problem.

It is unclear whether models could learn such mathematical operations since even very large and powerful models, such as GPT-3 (Brown et al., 2020), currently only demonstrate simple addition and subtraction. This important and difficult aspect of defining the problem requires further research, but is essential if we are to create a machine learning task which is ecologically valid. The original Rotowire dataset contains a mix of countable statistics (at the game level only) and derived statistics such as percentages.

4 Evaluation by Information Extraction

The information extraction (IE) metrics of Relation Generation (RG), Content Selection (CS), and Content Ordering (CO), have been used extensively to compare systems operating on the Rotowire based datasets. These metrics are also key to the system design philosophy of Wiseman et al. (2017a) which aims to

> "[exploit] the fact that post-hoc information extraction is significantly easier than generation itself."

All of these metrics are based on IE models which learn to link statements in the text to tuples in the data. These models are trained on a corpus which has been automatically annotated using rules. Names or numbers in the text are linked to the possible data tuples which could represent them. During evaluation, tuples are predicted for the test text summaries, with a name, value and a type, e.g. 'Atlanta—96—TEAM-PTS'. These predicted tuples can be compared with the tuples in the data in order to determine whether facts predicted from the text match those in the data. For example, if the fact extracted from the text is 'Atlanta—96—TEAM-PTS', but in the data we see 'Atlanta—105—TEAM-PTS', then based on the match of both name and type, we can determine that the number is wrong.

The CS metrics use these tuples to measure how many of the tuples from the predicted text exist in the gold standard text. The CO metric measures the order of these tuples. For more details see (Wiseman et al., 2017b) (we have purposefully cited the arXiv version of this paper as it includes an additional appendix detailing the procedure).

4.1 Extended IE Metrics

We extend the IE based metrics using the data now available in Thomson et al. (2020). Details can be found on GitHub[3]. We make two modifications:

- Extend the annotation logic such that it can detect the additional entities and attributes we added in subsection 3.1. For example, days of the week for both the current game and the subsequent game for each team.
- Use a season-based partition scheme so that the IE model is not being used to evaluate data upon which it was previously conditioned. We use the 2014, 2015 and 2016 seasons to train, 2017 to validate, and 2018 to test. This is the same problem in the partition scheme which was identified for the text generation system by Thomson et al. (2020).

5 Experimental Setup

5.1 NLG System Setup

We created two datasets:

D1: Where we emulated as closely as possible the data format and content used by Rebuffel et al. (2020) except using season-based partitions.

D2: Keeping all data from D1, but adding a new entity for the GAME, and two for the NEXT-GAME as detailed in subsection 3.1.

We then trained models using the system of Rebuffel et al. (2020) on each dataset, with 10 different random seeds, to determine whether adding the additional information improved the results. We also tested whether changing the early-stop strategy impacted the results, taking a snapshot from training using each of the best BLEU (Papineni et al., 2002), RG, CS-PREC (precision), CS-REC (recall), and CO.

[3] `https://github.com/nlgcat/adding_data`

Dataset	Stopping Metric	BLEU	RG	CS-PREC	CS-REC	CO
D1	BLEU	17.18 ± 0.386	0.70 ± 0.021	0.39 ± 0.015	0.38 ± 0.009	0.19 ± 0.006
D2	BLEU	17.39 ± 1.189	0.75 ± 0.034	0.43 ± 0.033	0.40 ± 0.019	0.21 ± 0.015
D1	RG	16.97 ± 0.435	0.71 ± 0.016	0.38 ± 0.015	0.38 ± 0.009	0.18 ± 0.008
D2	RG	17.00 ± 1.207	0.77 ± 0.029	0.42 ± 0.028	0.40 ± 0.013	0.21 ± 0.009
D1	CS-PREC	17.08 ± 0.358	0.71 ± 0.018	0.39 ± 0.012	0.38 ± 0.009	0.19 ± 0.007
D2	CS-PREC	17.30 ± 1.301	0.76 ± 0.026	0.44 ± 0.034	0.41 ± 0.015	0.21 ± 0.015
D1	CS-REC	17.12 ± 0.314	0.71 ± 0.017	0.39 ± 0.011	0.38 ± 0.007	0.19 ± 0.005
D2	CS-REC	17.27 ± 1.191	0.77 ± 0.026	0.43 ± 0.029	0.41 ± 0.015	0.21 ± 0.013
D1	CO	17.09 ± 0.540	0.70 ± 0.022	0.39 ± 0.012	0.38 ± 0.010	0.19 ± 0.003
D2	CO	17.34 ± 1.348	0.76 ± 0.025	0.44 ± 0.034	0.41 ± 0.014	0.22 ± 0.011
Gold	N/A	—	0.92	—	—	—

Table 2: Experiment results; Comparing D1 to D2 within every cell pair is statistically significant ($p < 0.005$) with the exception of entries in the BLEU column. Note that BLEU, CS, and CO all inherently achieve 100% on gold standard texts.

To summarise, we used two datasets, with 10 random seed each, all with 4 different early-stop strategies for 80 models total (2*10*4). We then calculated BLEU, RG, CS-PREC, CS-REC, and CO for each model.

5.2 Automated Metric Setup

We trained IE models following the general procedure proposed by (Wiseman et al., 2017b). We trained with different random seeds and learning rates, then chose the best 3 LSTM and the best 3 Convolutional models to ensemble. We then used the model ensemble, as well as BLEU-4 (Papineni et al., 2002), to evaluate the NLG system itself.

6 Results

Table 2 shows the results of our evaluation. We find a statistically significant difference ($p < 0.005$) for *all* information extraction based metrics (RG, CS-PREC, CS-REC, and CO) when we add the additional information as described in subsection 3.1. Information extraction based metrics increased in all cases when adding data, regardless of early-stopping method. Whilst BLEU scores also appeared to increase, we did not find the changes in them to be statistically significant. This is not surprising given that BLEU is known to not correlate for NLG (Reiter and Belz, 2009; Reiter, 2018), and even in machine translation it only correlates when differences are large (Mathur et al., 2020).

7 Conclusion and Future Work

Our results show that identifying data which should be included, then modelling it within the system architecture, increased all information extraction based metrics. Existing metrics have only been evaluated in limited ways for this domain. Improved metrics could help us evaluate systems, as well as find and categorise information gaps.

The subset of data selected for input, the form it takes (atomic versus aggregated), as well as the inclusion of system components/techniques (copy attention mechanism, hierarchical encoder, separate document plan, fact grounding, etc.), are all variables which could affect system performance. We plan in future work to perform ablation studies to determine which such variables, and in which combination, produce the best results. As part of this, we aim to create a unified code-base which will allow for components to be selected and configured in combination, for as many different data forms as possible.

Beyond this, we hope to move away from end-to-end system designs. This is similar in spirit to the idea proposed in Puduppully et al. (2019a), where a single model is not attempting to learn everything, the document plan is learned separately. We would extend such ideas to the data itself, if we can use both the known ontology from the structured data, as well as relationships and other information extracted with NLU or other tools, then this additional information could be input to systems which realize the language, meaning they are not left to solve both data analytic and language problems with a single model.

If we can define our data operations in terms of standard data models, such as relational models, then we will be closer to a general approach for filling the information gap in data-to-text.

Acknowledgments

We would like to thank our reviewers for their insightful feedback and questions.

The work presented here is partially funded by the Engineering and Physical Sciences Research Council (EPSRC), which funds Craig Thomson under a National Productivity Investment Fund Doctoral Studentship (EP/R512412/1).

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Ondřej Dušek, David M. Howcroft, and Verena Rieser. 2019. Semantic noise matters for neural natural language generation. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 421–426, Tokyo, Japan. Association for Computational Linguistics.

Heng Gong, Xiaocheng Feng, Bing Qin, and Ting Liu. 2019. Table-to-text generation with effective hierarchical encoder on three dimensions (row, column and time). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3143–3152, Hong Kong, China. Association for Computational Linguistics.

Matthew Honnibal. 2015. spacy. `https://spacy.io/`.

Hayate Iso, Yui Uehara, Tatsuya Ishigaki, Hiroshi Noji, Eiji Aramaki, Ichiro Kobayashi, Yusuke Miyao, Naoaki Okazaki, and Hiroya Takamura. 2019. Learning to select, track, and generate for data-to-text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2102–2113, Florence, Italy. Association for Computational Linguistics.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2020. Tangled up in BLEU: Reevaluating the evaluation of automatic machine translation evaluation metrics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4984–4997, Online. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019a. Data-to-text generation with content selection and planning. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, volume 33, pages 6908–6915, Honolulu, Hawaii.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019b. Data-to-text generation with entity modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2023–2035, Florence, Italy. Association for Computational Linguistics.

Clément Rebuffel, Laure Soulier, Geoffrey Scoutheeten, and Patrick Gallinari. 2020. A hierarchical model for data-to-text generation. In *Advances in Information Retrieval*, pages 65–80, Cham. Springer International Publishing.

Ehud Reiter. 2018. A structured review of the validity of BLEU. *Computational Linguistics*, 44(3):393–401.

Ehud Reiter and Anja Belz. 2009. An investigation into the validity of some metrics for automatically evaluating natural language generation systems. *Computational Linguistics*, 35(4):529–558.

Craig Thomson, Ehud Reiter, and Somayajulu Gowri Sripada. 2020. Sportsett:basketball - a robust and maintainable dataset for natural language generation. In *Proceedings of IntelLanG 2020*.

Harm de Vries, Dzmitry Bahdanau, and Christopher Manning. 2020. Towards ecologically valid research on language user interfaces.

Hongmin Wang. 2019. Revisiting challenges in data-to-text generation with fact grounding. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 311–322, Tokyo, Japan. Association for Computational Linguistics.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017a. Challenges in data-to-document generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark. Association for Computational Linguistics.

Sam Wiseman, Stuart M. Shieber, and Alexander M. Rush. 2017b. Challenges in data-to-document generation.