The Accuracy Evaluation Shared Task as a Retrospective Reproduction Study

Craig Thomson
Dept of Computing Science
University of Aberdeen
Aberdeen, UK
c.thomson@abdn.ac.uk

Ehud Reiter
Dept of Computing Science
University of Aberdeen
Aberdeen, UK
e.reiter@abdn.ac.uk

Abstract

We investigate the data collected for the Accuracy Evaluation Shared Task as a retrospective reproduction study. The shared task was based upon errors found by human annotation of computer generated summaries of basketball games. Annotation was performed in three separate stages, with texts taken from the same three systems and checked for errors by the same three annotators. We show that the mean count of errors was consistent at the highest level for each experiment, with increased variance when looking at per-system and/or per-error-type breakdowns.

1 Introduction

To address issues of factual accuracy in data-to-text systems, we developed a protocol for annotating mistakes in NLG texts (Thomson and Reiter, 2020). This protocol was used to create a corpus of errors found in generated basketball game summaries. The corpus was then used in the Accuracy Evaluation Shared Task (Thomson and Reiter, 2021), where participants submitted automatic metrics or alternative human evaluations that were compared to the gold standard. The corpus was created in three stages under experimental conditions which were largely the same. The same user interface and platform (Amazon Mechanical Turk) were used, along with the same three annotators who each checked every game summary in all experiments. The only changed conditions were the game summaries that were checked, and slight clarification of the instructions based on annotator queries and feedback. Therefore, it can be retrospectively considered a reproduction attempt.

The original goal of these human evaluations was to design and develop a reliable protocol, then create gold list of errors (for training metrics, etc). In Thomson and Reiter (2020) we performed an initial run of the protocol, with two subsequent runs in Thomson and Reiter (2021) to extend the first run to form a training set, and then create a test set in the third run for use in a shared task. Generated texts were annotated in equal proportions, within each experiment, from three different systems (Wiseman et al., 2017; Puduppully et al., 2019; Rebuffel et al., 2020) which at the time were representative state of the art systems on the RotoWire dataset (Wiseman et al., 2017)[1] of English language basketball summaries paired with box score data tables.

In this paper we examine whether a similar number of errors were obtained in each experiment. We do this for all systems combined, the ensemble of representative errors that we intended to collect, and also at the per-system level, where we look at the errors for each system in isolation. We also discuss the issues that might be encountered when trying to reproduce or otherwise verify results obtained using the gold standard protocol for factual accuracy.

2 Related work

It is crucial that our evaluation protocols are reliable, something that can be demonstrated by reproducing experimental results under similar conditions. Such reproduction work is seldom carried out within the field of NLP (Belz et al., 2021a). When it is, researchers experience difficulties obtaining the same results or finding the information required to run the experiment at all (Mieskes et al., 2019). Problems with reproduction are not limited to NLP. In a large scale survey of over 1,500 researchers, Baker (2016) found that 90% felt there was a reproducibility crisis, with over 50% deeming it 'significant'. The ReproGen shared task, for which this paper is a submission, aims to document reproduction attempts in NLP and provide an improvement in levels of reproducibility over time.

[1] https://github.com/harvardnlp/boxscore-data

Proceedings of the 15th International Conference on Natural Language Generation: Generation Challenges, pages 71 - 79

2.1 Evaluation by annotation

Whilst evaluation of text generation systems is usually done by rating or ranking (van der Lee et al., 2019; Gehrmann et al., 2022), approaches for evaluation by annotation have also been proposed. Popović (2020) asked participants to highlight problematic spans within machine translated text, which were then categorised by severity. This allowed for the count of errors to be used to rank systems, but with the benefit of the individually reported errors being amenable to error analysis, something that is important for MT and NLG (van Miltenburg et al., 2021). With the SCARECROW framework, Dou et al. (2022) asked annotators to highlight problematic spans of text in prompted generation, these were then categorised. The categories are diverse, covering grammatical issues as well as issues readers might have, such as needing an external resource to check a fact. There is also the task-dependant category of 'off prompt'. Agreement for many categories was low, with errors for all categories except 'off-prompt' being reported by two or more annotators (out of ten who annotated each paragraph) in less than 50% of cases. Freitag et al. (2021) instructed annotators to highlight errors within machine translated texts, then categorise each error with one type from a hierarchy of error types. Errors were also assigned a severity level by annotators. For text simplification, Devaraj et al. (2022) used an approach whereby annotators highlighted spans of text then labelled them using the task specific label of whether information was inserted, deleted, or substituted, as well as how severe the error was.

3 The Gold standard protocol for factual accuracy

The gold standard protocol detailed in Thomson and Reiter (2020) uses human annotators to check the factual accuracy of generated texts. As part of this work, basketball games summaries were annotated for factual errors. These summaries are based on complex data, often including information from outwith the game being summarised, such as aggregated statistics or upcoming game schedules. This presents problems of error detection that are not found in simpler tasks. Fact checking is performed against a comprehensive external data source rather than system input data, i.e., annotators check whether the text truthfully reflects what actually happened in the basketball games. Full details can be found in Thomson and Reiter (2020) and Thomson and Reiter (2021), although a brief overview is included here.

3.1 Annotation

Annotators are asked to highlight non-overlapping spans of text that are factually inaccurate, then mark each span with an error type, as well as a correction or comment explaining why the text is inaccurate. The types are:

NAME$^{\mathbf{N}}$: Incorrect named entity - Including people, places, organisations, and days of the week.

NUMBER$^{\mathbf{U}}$: Including both numbers which are spelled out, and those expressed as digits.

WORD$^{\mathbf{W}}$: A word which is not one of the above and is incorrect.

CONTEXT$^{\mathbf{C}}$: A phrase which causes an incorrect inference because of context or discourse.

NOT CHECKABLE$^{\mathbf{X}}$: A statement which can not be checked because the information is not available, or it would be too time-consuming.

OTHER$^{\mathbf{O}}$: Any other mistakes, a last-resort category for when the text is nonsensical.

The colours and superscript for these types are explained in Figure 1.

3.2 Curation and complex annotation

When multiple annotators check each text, a curation process is used to resolve disagreement between annotators. This is done by a researcher, although it could be performed by separate, suitably trained annotator. All errors that are found by the majority of annotators (2/3 in the shared task) are taken to form the Gold Standard Mistake List (GSML). In cases where the spans or categories differ slightly, but it is clear the annotators found the same fundamental problem in the text, the curator can include the error in the GSML, noting how many annotators found the underlying problem.

To highlight errors in text using our annotation scheme we use an accessible colour palette (https://davidmathlogic.com/colorblind, https://personal.sron.nl/~pault) with the addition of superscript letters such that annotations can be read even in black and white. Our annotation categories with their styles are: NAME^{N} NUMBER^{U} WORD^{W} CONTEXT^{C} NOT CHECKABLE^{X} and OTHER^{O}

Figure 1: Annotation key for error types (used throughout)

For example, consider the two following annotated sentences:

> Steph Curry scored 30^{U} points to go with 9 rebounds.
>
> Steph Curry^{N} scored 30 points to go with 9 rebounds.

If in the game Curry had 9 rebounds, but only 25 points, then the sentence can be annotated as per the first example. However, if another player, Kevin Durant, had 30 points and 9 rebounds, then an annotator could instead mark the name as an error (second example). We refer to such cases as complex annotations, where there might be multiple valid ways to indicate an error in the text. To help mitigate this problem, annotators are asked to use as few annotations as possible to express the underlying error. They are also asked to prioritise error categories: NAME^{N} > NUMBER^{U} > WORD^{W} > CONTEXT^{C} > NOT CHECKABLE^{X} > OTHER^{O}, e.g., Steph Curry^{N} would be the preferred annotation in the above example. Errors in neural generated texts are not always this simple. Generally speaking, the more errors that are in a sentence, the more difficult it becomes to find the preferred annotation.

4 Experiment setup

Generated basketball summaries from the same three systems were used in each experiment. The systems were the conditional copy system of Wiseman et al. (2017), the document plan system of Puduppully et al. (2019), and the hierarchical encoder of Rebuffel et al. (2020). These systems were chosen because each was considered state-of-the-art (by one or more metrics) at the time of publication. Generated game summaries were provided by the authors of each paper, with the original RotoWire dataset and partitions having been used. The set of distinct games from the Rotowire test set was taken then randomly converted to a list. Selection of games from within this random list was arbitrary, with games for the training GSML taken from the start of the list, and those for the test GSML taken working backwards from the end.

Each input game record was processed by only one system, therefore there was no comparison between systems of generated texts for the same game data. This was because the original goal was the development of annotation techniques and a list of gold errors, and not the comparison of different systems. When comparing systems retrospectively as we are in this paper, we do so with this caveat.

The experiments we performed to collect data for the shared task were were:

> **Experiment A**: 21 texts, 7 per system (training set pt. 1). Collected in July 2020.
>
> **Experiment B**: 39 texts, 13 per system (training set pt. 2). Collected in January 2021.
>
> **Experiment C**: 30 texts, 10 per system (test set). Collected in March 2021.

where each text is a complete summary (approx. 300 words) of a basketball game, generated by one of the three neural systems.

4.1 Rotowire dataset partitions

The standard partitions of the RotoWire dataset have problems of training, validation, and test partition contamination, whereby the same game record exists within multiple partitions but with a different reference text (Iso et al., 2019; Thomson et al., 2020). Neural systems will memorise the text seen for a game in training, meaning that texts generated for such games in the test set will exhibit human-like levels of factual accuracy. For this reason, games in the standard RotoWire test set that had been seen during training or validation were excluded from selection for our experiment.

4.2 Annotator recruitment, instruction, and fair treatment

Annotators were recruited on the Amazon Mechanical Turk platform. We limited applicants to those

from the United States (where basketball is a popular sport), who held U.S. Bachelor degrees and were MTurk Masters[2]. We also screened participants with a qualifying task whereby they had to find 14 of 20 known errors in a text we had already annotated ourselves. Some errors such as whether a team could be said to 'dominate' might be subjective. It is for this reason that the qualifying bar was not set higher. Recruitment was performed only once, before any experiments. Four workers passed the qualification task and three chose to undertake the annotation work; these same three annotators each examined all 90 texts over the 3 experiments.

Workers were paid $8US per tasks, with each task taking 20-25 minutes. The aim was to pay $20US per hour, well above the minimum wage in the UK or any U.S. state. Based on feedback from the workers, we met or exceeded this rate. All workers were paid for all tasks, even those who failed qualification (with the exception of workers who submitted forms with zero errors).

In addition to paying workers fairly and promptly, we considered the impact that doing the work may have on their well-being, and made efforts to provide a positive working environment. Annotators can find repetitive tasks stressful Strassel et al. (2000). This stress could be compounded on crowd-source platforms where workers might have prior experience of being treated unfairly (Shmueli et al., 2021).

We maintained good communication by responding to queries they had and reassuring them that we were interested in their opinion, and they would not be punished for a "wrong" answer. In cases where annotators made procedural mistakes, we still paid them for the work and simply asked that they supply a correction. Feedback from annotators was highly positive, on both the level of communication and how much they enjoyed the work (it was less repetitive than other tasks they had done). Our approach was borne out of common courtesy, there was no complex process and it did not slow down the project. It also hopefully resulted in higher quality of annotation.

4.3 User Interface

We considered creating a custom annotation interface, although due to the relatively small number of annotated texts we instead opted for having annotators highlight errors in an MS Word document, then list the error type and correction in a list below the text. A researcher[3] then transcribed verbatim the annotations to an annotation tool, WebAnno[4]. The transcription process increases the time taken to process each annotated summary, which might be prohibitive in larger studies. It may also introduce a small amount of human error, which could be checked by repeating the transcription. However, given the volume of errors, we believe that mistakes in transcription will have a negligible effect on error counts in this study. This may change as models approach or exceed human levels of factual accuracy. In our case, we believe that the manual transcription work did not take more time than development and deployment of an interface would have. As a low-tech approach, it also reduced the possibility of failure. In the worst case scenario where a document failed to upload (did not happen) a worker could simply send us the document again. A failure on the interface could have resulted in data loss, so software testing would have been required.

Each MS Word document included our 4 pages of instructions and an annotated example that workers had been shown during qualification. Workers were told these were for their reference, and only the text to be annotated changed in each document. These instructions did change slightly between experiments, with difficult examples that annotators had queried being included as examples. The **NOT CHECKABLE**^{X} was also clarified.

4.4 Mean error count

Whilst the purpose of the original study was to find a list of representative errors for analysis and comparison with alternative approaches, we define here the mean error count (MEC) as a measure. The mean is calculated as the total number of errors by the number of summaries.

We consider pairwise combinations of system and error type granularity. System groupings are:

> **Ensemble**: Errors from all systems. This is what we originally set out to collect; a set of errors that is representative of the types of mistakes found in neural system output.
>
> **System**: Errors for each individual system.

[2]Reliable workers as determined by an Amazon internal metric

[3]The first author of this paper.

[4]https://webanno.github.io

Error type groupings are:

Overall Errors: The count of all errors, of any type.

Per-type Errors: The breakdown of errors by type.

For this study we consider the MEC at the level of reported errors, i.e., we count annotated token spans within each basketball summary that were provided by the annotators then combined by the curation process. This was the simplest option. We considered normalizing by token count but decided against it because annotator reported errors can span anything from a single token, to five or more. This does not necessarily mean the 5-token error is equivalent by any measure of severity to 5 single-token errors. Consider the two[5] annotated and tokenized sentences below:

> Steph Curry scored 28 points (9^{U} - 15^{U} - FG ; 4^{U} - 10^{U} 3Pt ; 2^{U} - 3 FT) .
>
> The Warriors were the dominant team in this second half of a back - to - back^{W}

These sentence may seem equally erroneous if normalized at the token level; both sentences have 5 annotated tokens. However, the annotations in the first sentence represent 5 separately reported NUMBER^{U} errors, whereas in the second there is a single WORD^{W} error spanning 5 tokens. The numbers in the first sentence are part of a shot breakdown, a terse domain specific syntax which shows the made and attempted shots at different ranges. A back-to-back means the team will play games on consecutive days. The problem described here may be compounded by the numbers within the shot breakdown always being included in pairs, they are the numerator and denominator of a fraction and each pair could be considered as a single error. Since we had asked to annotators to report NUMBER^{U} errors individually in our instructions, we performed the analysis at this level.

5 Results

We calculated the mean error counts (total errors by documents in experiment), as well as the coefficient of variation, CV*[6] (Belz et al., 2022). See our repository[7] for complete code and data, including the calculation of mean errors from the GSML. All values are calculated then rounded to two decimal places for inclusion in tables here.

Table 1: Mean Error Count (MEC) for Ensemble

experiment MEC			
A	**B**	**C**	**CV***
19.62	20.56	20.73	3.61

Table 2: Mean Error Count (MEC) for each type within the Ensemble

	experiment MEC			
error type	**A**	**B**	**C**	**CV***
NAME	5.33	5.26	7.07	21.26
NUMBER	8.86	7.38	7.47	12.80
WORD	4.43	6.18	4.67	22.80
CONTEXT	0.76	0.90	0.27	63.22
N-CHECK	0.19	0.85	1.27	86.35
OTHER	0.05	0.00	0.00	211.73

Ensemble overall errors: We can see from Table 1 than the mean error count (MEC) had low variance between experiments, with a coefficient of variation of 3.61. This is what the experiments had originally set out to do; acquire representative samples of errors from neural systems. That similar quantities were found from the same ensemble of systems within each experiment is reassuring.

Ensemble per-type errors: When looking at the per-type breakdown for Ensemble errors (Table 2), we can see that each individual variance is higher than for the overall counts. This is not unexpected, given the complex error resolution problem. The greatest variance is seen in error types having lower frequency; of the 1,836 total errors in the GSML, only about 4% were NOT CHECKABLE^{X}, 3% were CONTEXT^{C}, and a single OTHER^{O} error was reported between all systems and experiments.

Table 3: Mean Error Count (MEC) for each system

	experiment MEC			
system	**A**	**B**	**C**	**CV***
cond-copy	21.57	25.54	26.60	13.19
doc-plan	21.86	17.77	18.90	13.23
h-encoder	15.43	18.38	16.70	10.77

[5]This is an artificial example for clarity of comparison and conciseness, although multiple errors of both types can be found in the GSML.

[6]https://github.com/asbelz/coeff-var

[7]https://github.com/nlgcat/uoa-reprogen-2022

Table 4: Mean Error Count (MEC) for each error type within each system

system	error type	experiment MEC A	B	C	CV*
conditional copy	NAME	5.57	6.00	7.80	22.39
conditional copy	NUMBER	9.29	10.92	11.40	12.87
conditional copy	WORD	5.86	7.15	6.00	13.72
conditional copy	CONTEXT	0.43	0.23	0.10	79.89
conditional copy	NOT CHECKABLE	0.43	1.23	1.30	60.02
conditional copy	OTHER	0.00	0.00	0.00	-
document plan	NAME	5.71	5.08	6.40	14.12
document plan	NUMBER	11.14	6.15	7.00	40.30
document plan	WORD	4.43	5.38	3.80	21.50
document plan	CONTEXT	0.57	0.54	0.10	79.77
document plan	NOT CHECKABLE	0.00	0.62	1.60	133.60
document plan	OTHER	0.00	0.00	0.00	-
hierarchical encoder	NAME	4.71	4.69	7.00	29.64
hierarchical encoder	NUMBER	6.14	5.08	4.00	25.82
hierarchical encoder	WORD	3.00	6.00	4.20	41.95
hierarchical encoder	CONTEXT	1.29	1.92	0.60	63.71
hierarchical encoder	NOT CHECKABLE	0.14	0.69	0.90	82.68
hierarchical encoder	OTHER	0.14	0.00	0.00	211.73

System overall errors: The mean error count remained fairly constant for each system, although there were higher coefficients of variation than for the ensemble, ranging from 10.77 to 13.23 as shown in Table 3.

System per-type errors: When looking at the per-type breakdown for per-system errors, we see in Table 4 we see higher variance, especially for the less frequent error types.

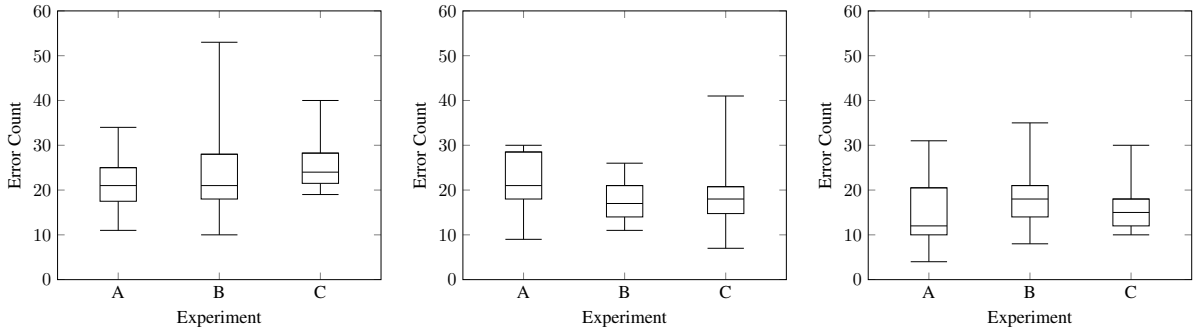
Figure 2 shows the spread of per-document error means for each system, within each experiment. It is worth noting that no generated text was error free and they rarely had fewer than 10 errors.

6 Discussion

The experiments showed that when taking an ensemble of 3 models to create the GSML, the mean error count remained relatively stable between experiments. This adds to the evidence of the gold standard protocol being a reliable method of obtaining instances of errors which can then be used to evaluate alternative methods, such as metrics (Kasner et al., 2021; Nomoto, 2021; Rezgui et al., 2021) or cheaper human evaluations (Garneau and Lamontagne, 2021).

The level of reproducibility for the gold standard protocol when evaluating systems is harder to determine. With a small number of texts per system in each experiment, the means are susceptible to the effects of outlier documents, such as rare cases where the document had 50 or more errors. The per-system coefficient of variation of ranged from about 10 to 13, which is similar to some CV* values reported for other human evaluations (Belz et al., 2021b). The per-type results are limited by the low frequency of some types, but also by the complex resolution problem. In some cases there can be many correct ways in which a text can be annotated for errors, using different combinations of error types.

An alternative way to measure the reproducibility of the protocol would be to run all three experiments again with different annotators. We could then look at how the sets of errors from the original experiment and the reproduction overlap. However, the problem of complex error resolution rears its head again. Just because annotation spans or categories differ, does not necessarily mean that both sets of annotators did not correctly identify the same underlying problem. This addresses the issues of complex error resolution in a way which the exact comparison of token spans and labels does not. Error verifiers can be asked to consider whether a reported error is one valid way to indicate the underlying problem. For discussion of complex annotation see Thomson and Reiter (2021).



(a) Conditional Copy (b) Document Plan (c) Hierarchical Encoder

Figure 2: Box plot for each system showing the spread of errors within each experiment.

Measuring reproducibility allows us to determine whether our evaluation protocols are reliable. However, it is not the only method for doing so. An alternative for validating the GSML would be to show individual errors to participants that are familiar with the annotation process, then ask them to indicate whether the highlight represents an error. This would allow us to measure the precision of annotators. We might also check in the same way, any errors reported by a minority of annotators. This would determine whether these errors were false positives, simply missed by the other annotators, or the result of differing annotations for complex errors.

7 Conclusion

This reproduction study showed that there was little variance in the mean error count between the different experiments that were used for the shared task data collection. Increased variance was observed when comparing the mean counts of different error types, and/or when comparing systems. These values do not, however, tell the whole story of this detailed evaluation protocol. For annotation based approaches the agreement between annotators can be measured (Popović and Belz, 2021), although with complex data-to-text, a lack of measurable agreement (based on token overlap) does not necessarily mean that annotators did not find similar underlying problems. An alternative when working at the level of individual errors might be to verify each reported error by asking additional annotators whether they agree with the reported error.

Acknowledgements

We would like to thank the Mechanical Turk annotators for their diligent work, as well as the reviewers for their helpful feedback. The original study was performed as part of Craig Thomson's PhD research, which was funded by an EPSRC NPIF studentship grant (EP/R512412/1).

References

Monya Baker. 2016. Is there a reproducibility crisis? *Nature*, 533:452–454.

Anya Belz, Shubham Agarwal, Anastasia Shimorina, and Ehud Reiter. 2021a. A systematic review of reproducibility research in natural language processing. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 381–393, Online. Association for Computational Linguistics.

Anya Belz, Maja Popovic, and Simon Mille. 2022. Quantified reproducibility assessment of NLP results. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16–28, Dublin, Ireland. Association for Computational Linguistics.

Anya Belz, Anastasia Shimorina, Shubham Agarwal, and Ehud Reiter. 2021b. The ReproGen shared task on reproducibility of human evaluations in NLG: Overview and results. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 249–258, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Ashwin Devaraj, William Sheffield, Byron Wallace, and Junyi Jessy Li. 2022. Evaluating factuality in text simplification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7331–7345, Dublin, Ireland. Association for Computational Linguistics.

Yao Dou, Maxwell Forbes, Rik Koncel-Kedziorski, Noah Smith, and Yejin Choi. 2022. Is GPT-3 text indistinguishable from human text? scarecrow: A framework for scrutinizing machine text. In *Proceedings of the 60th Annual Meeting of the Association*

for Computational Linguistics (Volume 1: Long Papers), pages 7250–7274, Dublin, Ireland. Association for Computational Linguistics.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

Nicolas Garneau and Luc Lamontagne. 2021. Shared task in evaluating accuracy: Leveraging pre-annotations in the validation process. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 266–270, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Sebastian Gehrmann, Elizabeth Clark, and Thibault Sellam. 2022. Repairing the cracked foundation: A survey of obstacles in evaluation practices for generated text.

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.

Zdeněk Kasner, Simon Mille, and Ondřej Dušek. 2021. Text-in-context: Token-level error detection for table-to-text generation. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 259–265, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Margot Mieskes, Karën Fort, Aurélie Névéol, Cyril Grouin, and Kevin Cohen. 2019. Community perspective on replicability in natural language processing. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 768–775, Varna, Bulgaria. INCOMA Ltd.

Tadashi Nomoto. 2021. Grounding NBA matchup summaries. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 276–281, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Maja Popović. 2020. Informative manual evaluation of machine translation output. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5059–5069, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Maja Popović and Anya Belz. 2021. A reproduction study of an annotation-based human evaluation of MT outputs. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 293–300, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-text generation with content selection and planning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6908–6915.

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.

Rayhane Rezgui, Mohammed Saeed, and Paolo Papotti. 2021. Automatic verification of data summaries. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 271–275, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Boaz Shmueli, Jan Fell, Soumya Ray, and Lun-Wei Ku. 2021. Beyond fair pay: Ethical implications of NLP crowdsourcing. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3758–3769, Online. Association for Computational Linguistics.

Stephanie Strassel, David Graff, Nii Martey, and Christopher Cieri. 2000. Quality control in large annotation projects involving multiple judges: The case of the TDT corpora. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00)*, Athens, Greece. European Language Resources Association (ELRA).

Craig Thomson and Ehud Reiter. 2020. A gold standard methodology for evaluating accuracy in data-to-text systems. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 158–168, Dublin, Ireland. Association for Computational Linguistics.

Craig Thomson and Ehud Reiter. 2021. Generation challenges: Results of the accuracy evaluation shared task. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 240–248, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Craig Thomson, Ehud Reiter, and Somayajulu Sripada. 2020. SportSett:basketball - a robust and maintainable data-set for natural language generation. In *Proceedings of the Workshop on Intelligent Information Processing and Natural Language Generation*, pages 32–40, Santiago de Compostela, Spain. Association for Computational Lingustics.

Chris van der Lee, Albert Gatt, Emiel van Miltenburg, Sander Wubben, and Emiel Krahmer. 2019. Best practices for the human evaluation of automatically generated text. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 355–368, Tokyo, Japan. Association for Computational Linguistics.

Emiel van Miltenburg, Miruna Clinciu, Ondřej Dušek, Dimitra Gkatzia, Stephanie Inglis, Leo Leppänen, Saad Mahamood, Emma Manning, Stephanie Schoch, Craig Thomson, and Luou Wen. 2021. Underreporting of errors in NLG output, and what to do about it. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 140–153, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. 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.