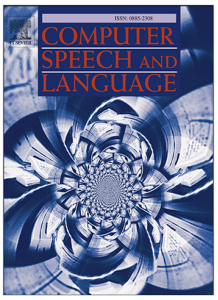
Journal Pre-proof



Evaluating factual accuracy in complex data-to-text

Craig Thomson, Ehud Reiter, Barkavi Sundararajan

PII: S0885-2308(23)00001-3
DOI: https://doi.org/10.1016/j.csl.2023.101482
Reference: YCSLA 101482

To appear in: *Computer Speech & Language*

Received date : 15 April 2022
Revised date : 27 October 2022
Accepted date : 3 January 2023

Please cite this article as: C. Thomson, E. Reiter and B. Sundararajan, Evaluating factual accuracy in complex data-to-text. *Computer Speech & Language* (2023), doi: https://doi.org/10.1016/j.csl.2023.101482.

Evaluating factual accuracy in complex data-to-text

University of Aberdeen

Dept. of Computing Science, Meston Building, 32 Elphinstone Rd, Aberdeen

Craig Thomson, Ehud Reiter, Barkavi Sundararajan

Abstract

It is essential that data-to-text Natural Language Generation (NLG) systems produce texts which are factually accurate. We examine accuracy issues in the task of generating summaries of basketball games, including what accuracy means in this context, how accuracy errors can be detected by human annotators, as well as the types of accuracy mistakes made by both neural NLG systems and human authors. We also look at the effectiveness of automatic metrics in measuring factual accuracy.

Key words: Natural Language Generation, complex data-to-text, evaluation, annotation, factual accuracy, neural data-to-text

1. Introduction

Data-to-text systems use Natural Language Generation (NLG) techniques to produce texts which help readers understand non-linguistic data by describing, summarising, explaining, and more generally giving insights about the data. For example, data-to-text systems can generate written weather forecasts from numerical weather predictions (Arun et al., 2020), or they can summarise complex medical data for clinicians (Portet et al., 2009).

In order to be useful, data-to-text systems must generate texts which are *accurate*. It is not acceptable to give a doctors inaccurate information about a

Email addresses: c.thomson@abdn.ac.uk (Craig Thomson), e.reiter@abdn.ac.uk (Ehud Reiter), b.sundararajan.21@abdn.ac.uk (Barkavi Sundararajan)

patient! An industry panel at a recent INLG conference emphasised that even rare accuracy errors can cause users to lose trust in a data-to-text system and hence stop using it[1].

In simple tasks such as the E2E challenge (Dušek et al., 2020), which involves generating short descriptions of restaurants that explicitly communicate a small number of atomic attributes, accuracy is usually regarded as a combination of *hallucination* (when the output text includes an attribute which was not present in the input data) and *ommission* (when the output text does not mention one or more of the attributes in the input data). However, in more complex data-to-text tasks, where the system is generating multi-paragraph summaries and insights about a large data set, it is harder to define accuracy.

As Van Deemter and Reiter (2018) point out, 'deviations from the truth' are inevitable in such systems, because they are inherent in the complex data-to-text task. Since it is not possible for a 300 word text to completely communicate 1,000 (or 1,000,000) data points, omission in the above sense is inevitable. We can instead assess whether such systems include the most useful insights in the generated text (content selection), but this is not an accuracy issue. Also, systems which calculate insights using domain knowledge and/or use fuzzy words (such as *significant*), both of which are highly desirable in data-to-text, may make mistakes due to inaccurate domain knowledge or inappropriate use of vague words. We can ask such systems to make similar (or fewer) mistakes than human writers, who face the same challenges, but we cannot insist that such systems never communicate information which is incorrect or misleading.

In order to be able to generate accurate texts in a complex data-to-text setting, we first need to define what *accuracy* means in the context of such systems, and how to measure and evaluate it. In this paper, we explore these issues in a specific domain, which is generating moderate length (300 word) summaries of basketball games from box-score statistics.

Specifically, we

[1] https://ehudreiter.com/2021/09/27/inlg-what-real-world-users-want

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], OTHER[O], and NOT CHECKABLE[X].

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

- look at the subtleties of defining accuracy, including difficult edge cases;
- present a human protocol for identifying accuracy errors in this domain;
- analyse the accuracy errors (as determined by this protocol) made by a selection of neural NLG systems, and how this compares to errors made by human writers; and
- summarise work on using automatic metrics to measure accuracy.

To give a concrete example, fig. 2 shows an extract from one of the texts we worked with, which is a game summary produced by a neural NLG system. The figure also shows the accuracy errors in this extract, annotated using the scheme from fig. 1.

2. Related Work

Traditionally, data-to-text systems have been implemented using hand-crafted rules or templates (Reiter, 2007; Reiter and Dale, 2000). Through careful symbolic modelling of the generation process this approach ensures factual accuracy, but can be difficult and time consuming to implement. Neural systems for data-to-text have shown potential for alleviating some of this manual burden (Dušek et al., 2018; Gardent et al., 2017; Lebret et al., 2016) but are often limited to simple problems that are not as challenging for rule-based approaches and

contain only categorical variables, not numeric ones like, for example, in the systems of BabyTalk (Portet et al., 2009) and SUMTIME (Reiter et al., 2005). The RotoWire dataset of basketball game data and human-authored summaries, as well as the task based upon it (Wiseman et al., 2017), sought to push the boundaries of neural approaches by trying to generate longer (about 300 word) texts based on more complex data. The MLB dataset (Puduppully et al., 2019) is similar, except with baseball game data and summaries. Another more recent dataset, ToTTo (Parikh et al., 2020), also allows for the exploration of generation based on numeric, tabular data, although it is not as complex as the multi-table RotoWire, especially with the latter being extended on the data side in SportSett (Thomson et al., 2020) to model the relationship between games and increase granularity within them.

With this shift in research from generating texts using rules that inherently restrict the generation such that it is factually accurate[2], to neural systems that can hallucinate, detection of factual accuracy errors has become a major issue. Various systems have used the RotoWire dataset to investigate complex data-to-text generation (Iso et al., 2019; Puduppully et al., 2019; Puduppully and Lapata, 2021; Rebuffel et al., 2020; Wang, 2019) and whilst some improvement has been seen, all have noted in their examples that a large volume of factual mistakes remain. Part of the difficulty in progressing machine learning tasks on such complex dataset comes from the limitations of the commonly used metrics.

[2]Factual accuracy errors in a rule-based systems are simply code bugs or source data errors.

The Memphis Grizzlies (5-2^{U}) defeated the Phoenix Suns (3 - 2) Monday^{N} 102-91 at the Talking Stick Resort Arena^{N} in Phoenix. The Grizzlies had a strong^{W} first half where they out-scored^{W} the Suns 59^{U}-42^{U}. Marc Gasol scored 18 points, leading^{W} the Grizzlies. Isaiah Thomas added^{C} 15 points, he is averaging 19 points on the season so far^{X}.

List of errors:

- 2^{U}: incorrect number, should be 0 (losses).
- Monday^{N}: incorrect named entity, should be Wednesday.
- Talking Stick Resort Arena^{N}: incorrect named entity, should be US Airways Center.
- strong^{W}: incorrect word, the Grizzlies did not do well in the first half.
- out-scored^{W}: incorrect word, the Suns had a higher score in first half.
- 59^{U}: incorrect number, should be 46.
- 42^{U}: incorrect number, should be 52 .
- leading^{W}: incorrect word. Marc Gasol did not lead the Grizzlies, Mike Conley did with 24 points.
- Isaiah Thomas added^{C}: context error. Thomas played for the Suns, but context implies he played for the Grizzlies and added to their score.
- averaging 10 points on the season so far^{X}: not checkable. This is hard and time consuming to check as the information is not present in this exact form in data sources.

Figure 2: Example text with error annotations. Corrections and explanations are not required, but are included here for clarity. Box score data for this game is available at `https://www.basketball-reference.com/boxscores/201411050PHO.html`.

BLEU (Papineni et al., 2002), a word overlap metric, remains in use despite its flaws being widely known (Mathur et al., 2020; Reiter, 2018). With longer texts that can include a wide variety of different insights to achieve the same communicative goal, using a small set of reference texts can be problematic. There are many ways the story of a basketball game can be correctly told, there is no exact set of facts that should be included. More recent methods for evaluating machine generated texts such as BERTScore (Zhang et al., 2020) and BLEURT (Sellam et al., 2020) may be useful for shorter texts, but share this fundamental limitation. Certainly, they should not be applied to this task without first evaluating their efficacy within it.

The other common metrics used for this task are the information extraction techniques of relation generation (RG), content selection (CS), and content ordering (CO) proposed by Wiseman et al. (2017). These techniques extract information (in the form of semantic triples) from both reference and generated texts, then compare the sets of triples either with each other or the set of triples from the input data. Such an approach has stronger theoretical foundations for application to this task than word overlap metrics, although it has not yet been demonstrated to correlate with human judgment. Predicting true facts is also not the same thing as detecting false facts. Hallucination is one of the most common causes of error in neural data-to-text systems (Dušek et al., 2019) yet the RG-based approaches would miss them because the models are only trained to detect facts that were present in the input data, they never see examples of hallucination.

Human evaluations are also used to investigate data-to-text systems, with the suggested best practice being rating by likert scale (van der Lee et al., 2019). Participants can also be asked to count supported and contradicted facts in the text, based on the source data (Wiseman et al., 2017). These approaches are perhaps better suited for shorter generations as it is difficult to rate overall accuracy for a long text that contains many different errors. They are also be limited in their amenability to error analysis (van Miltenburg et al., 2021) which is crucial in aiding our understanding of what is going on underneath the

numerical results. Concerns have been raised over both task setup within NLP (Raji et al., 2021), as well as the serious issues in NLG evaluation (Gehrmann et al., 2022), with poor evaluations methods remaining in use despite their flaws. Evaluation by annotation, whereby individual errors are highlighted within texts is one way of performing evaluations such that they are both reliable and present us with meaningful individual errors for analysis. This has been investigated for machine translation (Freitag et al., 2021; Popović, 2020), prompted generation (Dou et al., 2022), as well as data-to-text (Thomson and Reiter, 2020).

The FEVER workshops and shared tasks (Thorne et al., 2018b, 2019) have also explored fact checking, aiming to identify factual errors in manually 'mutated' texts (Thorne et al., 2018a). These texts (Wikipedia articles) may, however, be less densely packed with numerical values than sports journalism texts.

Our focus in this paper is on data-to-text, but similar issues arise in text-to-text applications. Moramarco et al. (2022) present a human protocol which includes some error annotation for evaluating summaries of patient-doctor consultations.

3. Basketball domain

The domain of automated sports journalism provides us with a sensible real-world complex data-to-text problem. Textual summaries of sports games are written commercially, both by humans and automatically[3]. These game summaries often include a transcription of statistics for players and teams within the game, as well as insights that explain to the reader why one team defeated another or why the statistics that are mentioned are significant (in the colloquial sence). An example human-authored game summary from the RotoWire corpus is shown in fig. 3, with partial data tables for the same game shown in table 1 and table 2. In the summary, we can see that it begins with a description of which two teams played, who won, where, and when. The team records

[3]https://www.br.de/nachrichten/sport/wie-textautomatisierung-br-sport-unterstuetzt,SIl2t2b

following the game (wins-losses) are shown in parenthesis. The author then notes that the winning team, the Heat, have been consistently good recently. The summary then details the statistics of prominent players, although it does so through narrative rather than simple transcription. The author notes that all five of the starting players (each team can only have five players on the court at any given time) scored double-digit points, a team effort. They then describe the statistics of the two best players in more detail, including a breakdown of their shooting from different ranges (in parenthesis). The upcoming game for Orlando is then mentioned. The same is repeated for the losing team, Detroit, but with the author explaining that despite a good performance from one player, the team did not perform well overall. This 'difference in the game' is a common narrative used in these summaries, with insights carefully selected so the author can explain to the reader *why* one team defeated the other. The insights used can differ, as can the number of players mentioned. Sometimes the team rebounding totals or shooting percentages will be compared, although in this case the author clearly felt the double-digit performances of the Magic told the story.

Table 1: Orlando Magic partial player statistics and full team totals.

Name	Starter	FG	FGA	FG%	3P	3PA	3P%	FT	FTA	FT%	REB	AST	STL	BLK	TOV	PTS
Tobias Harris	Yes	10	17	0.588	3	6	0.5	1	2	0.5	5	4	5	1	0	24
Nikola Vučević	Yes	10	18	0.556	0	0	-	5	5	1	14	3	0	0	1	25
Victor Oladipo	Yes	3	5	0.6	1	1	1	4	5	0.8	6	3	0	1	4	11
Evan Fournier	Yes	5	13	0.385	3	6	0.5	1	2	0.5	2	8	1	0	4	14
Channing Frye	Yes	4	6	0.667	4	6	0.667	0	0	-	5	3	0	0	1	12
Orlando Totals	-	41	79	0.519	13	26	0.5	12	15	0.8	39	33	9	2	11	107

Table 2: Detroit Pistons partial player statistics and full team totals.

Name	Starter	FG	FGA	FG%	3P	3PA	3P%	FT	FTA	FT%	TRB	AST	STL	BLK	TOV	PTS
Andre Drummond	Yes	5	12	0.417	0	0		2	4	0.5	10	0	0	3	2	12
Caron Butler	No	5	11	0.455	3	5	0.6	7	7	1	6	1	0	0	0	20
Detroit Totals	-	32	80	0.4	11	24	0.458	18	24	0.75	41	17	5	4	14	93

In addition to including transcriptions of basic facts such as the points (PTS), rebounds (REB), assists (AST), steals (STL), or blocks (BLK) of play-

The Orlando Magic (5-7) took down the Detroit Pistons (3-8) 107-93 on the road at The Palace of Auburn Hills on Monday night. The Magic may finally have found their groove. Winners of three of their last four games, Orlando is looking like a team that you don't want to play right now. On Monday night, the team was led by an impressive all-around performance from their starting lineup. Each of the five starters managed to put up double-digit points, with Nikola Vučević posting an impressive double-double that included 25 points (10-18 FG, 5-5 FT), with 14 rebounds and three assists. Tobias Harris also had another strong night, scoring 24 points (10-17 FG, 3-6 3Pt, 1-2 FT), and contributing five rebounds, five steals, four assists and one block. Orlando heads back home next, and will be tested against the Los Angeles Clippers on Wednesday night. The Pistons on the other hand have lost two straight, and five of their last six games. This team can't get the ball rolling in the right direction currently, regardless of who steps up. On Monday, it was Caron Butler who stepped up off the bench for Detroit, scoring 20 points (5-11 FG, 3-5 3Pt, 7-7 FT), with six rebounds and one assist. The Pistons also got a double-double from big man Andre Drummond, who scored 12 points (5-12 FG, 2-4 FT), to go along with 10 rebounds and three blocks. Detroit will remain at home as they take on the Phoenix Suns on Wednesday night.

Figure 3: Human authored game summary for Orlando@Detroit on November 17th 2014. `https://www.basketball-reference.com/boxscores/201411170DET.html`

ers, human authors will also elaborate on these statistics to indicate whether the player had a good game. One common method is to use the domain specific term *'double-double'*, which means that a player had double-digits in exactly two of the aforementioned statistics. The term can also be used for exactly three (triple-double), four (quadruple-double), or five (quintuple-double) categories being in double-digits. These terms are commonly used in the domain to indicate well-rounded performance, and phrases like *'his fourth consecutive double-double'* can convey consistent performance over multiple games. The shot breakdowns, such as *'(10-17 FG, 3-6 3Pt, 1-2FT)'* in fig. 3, transcribe the successful and attempted shots at each of the the three different ranges. In this example, the player made 10 of 17 field goals (FG), of which 6 were shot

from outside the 3-point arc, with 3 of them being made. The player also took two free throws (penalty shots) and made one of them. The shot breakdown is domain specific syntax; other than deciding when to include one, they are entirely deterministic.

4. What does accuracy mean

It is not straightforward to define what constitutes an accuracy error in a basketball game summary. In this section, we define different types of accuracy errors, explain why we focus on real-world accuracy instead of fidelity to input data, and then discuss some difficult edge cases.

4.1. Categories

We categorise errors into one of the following categories (fig. 2 includes examples of most of these):

- *Incorrect named entity* (**NAME**$^{\mathbf{N}}$): This includes people, places, organisations, and days of the week.
- *Incorrect number* (**NUMBER**$^{\mathbf{U}}$): This includes numbers which are spelled out as well as digits.
- *Incorrect word* (**WORD**$^{\mathbf{W}}$): A word or phrase which is not one of the above and is incorrect.
- *Context error* (**CONTEXT**$^{\mathbf{C}}$): A word or phrase which causes an incorrect inference because of context or discourse.
- *Not checkable* (**NOT CHECKABLE**$^{\mathbf{X}}$): A statement which can not be checked; either the information is not available or it is too time-consuming to check.
- *Other* (**OTHER**$^{\mathbf{O}}$): Any other type of mistake (such as nonsensical phrases).

We believe that the above categories are both (A) useful to system developers and (B) understandable to human annotators. We experimented with more linguistically meaningful categories such as *Incorrect referring expression*, but some of our human annotators struggled to understand these categories.

Most of the above are semantic errors where the text explicitly communicates incorrect information. **CONTEXT^{C}** errors are an exception, these cover cases where the text is literally correct but pragmatically encourages the reader to make an incorrect inference. An example is the statement *Isaiah Thomas added 15 points* in fig. 2; this is literally true in the sense that Thomas scored 15 points, however it is pragmatically misleading because in context it implies that Thomas played for the Grizzlies when in fact he played for the Suns. We believe that such pragmatic errors are important and need to be included in accuracy evaluations.

The **NOT CHECKABLE^{X}** category covers statements which cannot be checked (or would take too long for human annotators to check), such as claims about the mental states or ambitions of players. In principle **NOT CHECKABLE^{X}** statements could be true, however in practice they have almost always been false, at least in texts generated by neural NLG systems (as opposed to human-written texts). This is probably because neural NLG systems do not have access to data which would enable them to make such statements.

4.2. Real-world error vs not in the data?

When we measure accuracy, in principle we can look at either:

- *Real-world accuracy*: Is the information in the text true in the real world?
- *Fidelity to data*: Is the information in the text derivable from the system's input data?

We use the first of these, real-world accuracy, because we believe it is more appropriate for complex data-to-text, easier for human annotators, and easier to define.

With regard to the first point, the difference between the 'in-the-data' and 'real-world' approaches is most noticeable when there are facts or insights which are not present in the data but can be inferred from other data with high but not perfect confidence. For example, suppose the data for a basketball game records whether the game is 'home' or 'away' for each team, but not the actual location of the game. We can make strong guesses about location from this data; e.g., a home game played by the Memphis Grizzlies will probably be in Memphis. Neural models could even learn this strong guess from the reference texts. However, there are exceptions (e.g., NBA Global Games[4]). In this case, stating that a home game for the Grizzlies was played in Memphis would always be considered an error under the 'in-the-data' approach, but would only be considered an error under the 'real-world error' approach if the game was actually played somewhere else (which is rare).

We believe that effective summaries of complex data should include insights which are highly likely but not 100% reliable; indeed such insights provide much of the 'value-added' of text summaries compared to graphical or tabular presentations of data. In safety-critical domains such as medical reporting, we may wish to explicitly add hedges such as *probably* to such insights. But such hedges are rare in sports reporting. Therefore, because we do not want to discourage the use of inferred insights in texts, we only regard such insights as inaccurate when the inference is incorrect in the real-world.

Also, from a pragmatic perspective it is easier for human annotators who have domain expertise to detect real-world errors. They do not need to check whether things they already know to be are true are present in the input data, and they can use existing resources (tools, websites, etc) which they are familiar with to find out what actually happened, without worrying about whether all the information in the resource is present in the NLG system's input data.

Last but not least, measuring 'fidelity to data' requires knowing what data the system has access to. Currently this is relatively straightforward in the

[4] `https://en.wikipedia.org/wiki/NBA_Global_Games`

basketball domain. However, in other domains data-to-text systems may pull in additional information from databases, free-text reports, and indeed the internet as a whole. In such cases it is almost impossible to specify exactly what information a system uses and hence assess 'fidelity to data'; but we can still assess whether the information in a text is true in the real world.

5. Evaluation by annotation

Using the definitions of factual accuracy provided in section 4 we created an annotation procedure whereby individual annotators could mark errors in a textual summary. Annotators are asked to highlight non-overlapping spans of text which contain an error, along with a category for said error using our categories of **NAME**$^{\mathbf{N}}$, **NUMBER**$^{\mathbf{U}}$, **WORD**$^{\mathbf{W}}$, **CONTEXT**$^{\mathbf{C}}$, **OTHER**$^{\mathbf{O}}$, and **NOT CHECKABLE**$^{\mathbf{X}}$. Annotators also provided a correction if possible, or a comment explaining their reasoning for the highlighted span being an error.

Performing such annotation work requires not only general numeracy and literacy, but domain knowledge (in our case, of basketball). As such, potential annotators are screened by a qualification exercise where they annotate mistakes in a text that had already been carefully annotated by us. We also suggest providing annotators with a small quantity of practice work, such that they can ask any questions, before going on to live experiment work. We provide annotators with detailed instructions (about 4 pages) which explain each error category and provide in-domain examples.

5.1. Difficulty: more than one way to annotate errors

Sometimes there are multiple ways of annotating errors. For example, consider the sentence:

> Lou Williams scored 30 points and had six rebounds.

Suppose that it was another player, Solomon Hill, who had 30 points and 6 rebounds. In this case, the sentence could be corrected either by changing the

player name (to Solomon Hill), or by changing the statistics (to the correct ones for Lou Williams):

- Lou Williams scored **30**$^{\mathbf{U}}$ points and had **6**$^{\mathbf{U}}$ rebounds.
- **Lou Williams**$^{\mathbf{N}}$ scored 30 points and had 6 rebounds.

In other words, we can either annotate the numbers as incorrect (should be *14* and *1*) or the name as incorrect (should be *Solomon Hill*).

One possibility is to prefer the annotation with the fewest number of errors, which in the above example is the second one. But this does not cover all cases, and is not always straightforward. For example, one sentence in our corpus was analysed in the two ways shown below:

> **Annotator T1**: The only other$^{\mathbf{W}}$ **Raptor**$^{\mathbf{N}}$ to reach double figures in points was **Dwyane**$^{\mathbf{N}}$ Dragic, who came off the bench$^{\mathbf{W}}$ for 22 points (**9**$^{\mathbf{U}}$-**17**$^{\mathbf{U}}$ FG, 3-7 3Pt, 3-3 FT), **six**$^{\mathbf{U}}$ rebounds and five assists.
>
> **Annotator T3**: The only other$^{\mathbf{W}}$ Raptor to reach double figures in points was **Dwyane Dragic**$^{\mathbf{N}}$, who came off the bench for **22**$^{\mathbf{U}}$ points (9-**17**$^{\mathbf{U}}$ FG, **3**$^{\mathbf{U}}$-7 3Pt, **3**$^{\mathbf{U}}$-**3**$^{\mathbf{U}}$ FT), **six**$^{\mathbf{U}}$ rebounds and **five**$^{\mathbf{U}}$ assists.

Table 3: Statistics for two players in the game Toronto@Miami on April 11$^{\text{th}}$, 2015: https://www.basketball-reference.com/boxscores/201504110MIA.html.

Name	Team	Unit	FG	FGA	3P	3PA	FT	FTA	REB	AST	STL	BLK	PTS
Lou Williams	Raptors	starter	9	18	4	7	7	7	1	2	0	0	29
Goran Dragić	Heat	bench	8	16	3	7	3	3	2	5	0	0	22

The statistics for two players this sentence is most likely to be about are shown in table 3. Two different annotators (anonymised names of T1 and T3) were asked to annotate this sentence for errors. T1 essentially decided to change the player name to *Goran Dragic*; since *Dragic* played for the other team (*Heat*), they

also corrected *Raptors*. They then corrected three of the numbers accordingly and noted that *Dragic* did not come off the bench, he started the game. T3 disagreed, changing the player name to *Lou Williams* who did in fact start for the *Raptors*.

T1's annotation had fewer errors, but T3's annotation required changing fewer characters. T1's annotation was correct according to our instructions as we had asked that when there was a choice, the smaller number of annotations should be used. However, it is not trivial for annotators to search through multiple possible annotations looking for the smallest such set, and both annotators observed that there are many things wrong with the sentence. In a larger sense it is not clear which annotation is 'correct'; is number of errors at the semantic level more or less important than number of errors at the surface character level? Many data-to-text systems, including the one that generated the above text, do not output (or use as input) high-level information that could be used to inform annotators of the intended subject of the sentence, and even examining the attention mechanism would at best show that the model shifts focus between input triples associated with these players.

5.2. Other Difficult Cases

A perhaps related point is that it is difficult to annotate specific errors if the text includes sentences or phrases which are completely nonsensical, such as

> Markieff Morris also had a nice game off the bench, as he scored 20 points and swatted away late in the fourth quarter to give the Suns a commanding Game 1 loss to give the Suns a 118-0 record in the Eastern Conference's first playoff series with at least the Eastern Conference win in Game 5.

There are so many errors in this sentence (especially since the game being described is not a playoff game) that our annotators struggled to mark up specific errors.

There also were cases where results depended on how words were interpreted. For example, some people interpret *frontcourt* to mean 3 players (center, power forward, small forward[5]), while others interpret it to mean 2 players (just center and power forward[6]). Because of this difference, our annotators disagreed on whether the below sentence was an error or not.

> The Bucks' frontcourt did most of the damage.

We experimented with adding a glossary to resolve such issues. However the glossary was not always effective because an annotator who already knew what *frontcourt* meant might not check it in the glossary.

Finally, it is not always clear which specific words should be annotated as being part of an error. For example, one annotator could highlight **Boston Celtics**$^{\mathbf{N}}$ while a second may highlight **the Boston Celtics**$^{\mathbf{N}}$. Another example is where one annotator highlights **on the road**$^{\mathbf{W}}$, while a second simply highlights **the road**$^{\mathbf{W}}$, or even just **road**$^{\mathbf{W}}$ for the location of an upcoming game. This is not a problem if we are simply trying to count the number of errors in a text, but it is an issue if we want to compare or combine annotations, for example if we want to check whether a metric produces the same error annotation (at the token level) as a human evaluation.

6. Gold-standard protocol

With the gold-standard protocol we apply our method of annotation, having multiple annotators check each text. We then go through a curation process where annotations which are supported by multiple annotators are taken forward to form the Gold Standard Mistake List (GSML). For the Shared Task on Evaluating Accuracy (Thomson and Reiter, 2021) we did this on Amazon Mechanical Turk, where we recruited three workers with good standing on the

[5] https://www.sportsrec.com/8338357
[6] https://en.wikipedia.org/wiki/Basketball_positions

platform (Masters) who also held US Bachelors degrees. We aimed to pay workers $20US per hour and following feedback we believe we got our estimates right. Workers are paid per task not per hour, and since the length of text and number of workers can vary, so can the time taken for them to check a text. In a small number of cases where there were 40+ errors in a text, we paid workers a bonus.

For the shared task, we annotated 30 texts each from the systems of Wiseman et al. (2017), Puduppully et al. (2019), and Rebuffel et al. (2020). These 90 texts were originally split into sets of 60 training (20 per system) and 30 test (10 per system), although with the shared task concluded, we consider the whole set for discussions here. These texts were generated from randomly selected game records in the RotoWire test set, excluding games that are also present in the training or validation sets; see Iso et al. (2019) and Thomson et al. (2020) for details of this issue. The results of this annotation exercise are available at `https://github.com/ehudreiter/accuracySharedTask`.

6.1. Transcription interface

To apply the gold-standard protocol we first require an interface through which annotators can mark errors in text. A custom interface could be constructed, although to keep things simple we asked annotators to mark within an MS Word document any errors in a contained basketball summary, given links to game data on https://www.basketball-reference.com, using red and/or an underline. They also listed below the summary, each error they had highlighted, along with a category and a correction or note as to why the highlighted text constitutes an error. We then transcribed these annotations 'as is' to the WebAnno annotation tool[7]. Having annotators enter records directly into an annotation tool would save time, although eccentricities with the interface as well as issues of securely hosting the annotation tool online made the low-tech solution easier in this case. It is also easier for annotators to convey information should anything go wrong, they are not constrained to our interface and it is

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

also easier if work needs to be resubmitted for any reason (such as an issue with the MTurk platform), workers just send us an updated document and no work is lost because of a form submission error.

6.2. Curation

Each document is annotated by three annotators. When they have finished their work and their annotations have been transcribed, a curator then looks at the three annotations and creates a parallel annotation that will form part of the Gold Standard Mistake List (GSML). The curator looks at all errors that annotators marked and takes only those where the majority agreed that there was an error. Where annotators agree exactly on the token span of the error, this is simple. However, the complex annotation issue described in section 5.1 means that sometimes annotators will have found the same underlying problem but annotated it differently. In these cases, so long as it is clear they meant the same thing, the curator will take the simplest annotation method but record (as a property of the curation) how many annotators found the error in any form. For example, if one annotator marked *"**Durant**^{N} had 30 points and 10 rebounds."* and another *"Durant had **30**^{U} points and **10**^{U} rebounds."* then both have clearly determined that these are not the stats for Durant. Since our goal is to use annotations with the minimal number of errors, the curator should select the former but note that 2 annotators agreed there was an error relating to the annotation of **Durant**^{N}. The curator also records separately that only 1 annotator agreed on the category, and 1 annotator on the span of tokens that has been taken forward as the gold error (curation).

The curator also cleans any correction, category, and comment information that annotators noted. Usually this simply involves noting what the text should be replaced with, although in complex cases a note of how annotators disagreed on the error or method of annotation is included. For categories, the curator takes the majority category that was reported unless the annotators have clearly gone against the instructions (for example, accidentally marking a day of the week as a **WORD**^{W} rather than **NAME**^{N} error), in which case the curator

could override the majority.

The curation process can be seen as resolving the complexity that can occur in data-to-text errors. In most cases the curator did not need to intervene, making adjustments only when necessary. Whilst a researcher performed this task in our work, it could be done by another non-research annotator. However, the nature of the work means that it might not be well suited for allocation to a crowd-sourced worker.

6.3. Preparing the GSML

The Gold Standard Mistake List (GSML) is a list (in CSV format) of all curated errors that were found in the evaluated texts. Each row contains the document ID, as well as the sentence and token IDs for the error. The correction (if a direct replacement could be made) is noted, with a comment describing the error if a direct correction is not possible. The category of the error is also included. The GSML serves as a list of errors where we have high confidence that each entry is an error (see section 7 for verification of this). Whilst coverage is important, it is likely impossible due to complexity and disagreement, that there is a single ground truth set of errors.

The GSML, as well as the texts in WebAnno are carefully tokenized in advance (annotators marked up versions of texts with regular syntax as to not confuse them). Different syntactic parses such as Stanford NLP (Manning et al., 2014), NLTK (Bird et al., 2009), and spaCy (Honnibal and Johnson, 2015) often return different token lists for the same document. Sentence boundary disambiguation is also not a completely solved problem (Sadvilkar and Neumann, 2020). We therefore had to ensure before loading texts into WebAnno that texts were both tokenized and split into sentences (to avoid incorrect automated splitting during names such as *'C.J. Miles'*). We maintain dictionaries of these tokenization schemes for provenance. The RotoWire corpus contains only tokenized texts without provenance, some incorrectly tokenized texts had to be manually cleaned.

7. Verifying reported errors

The gold-standard protocol showed strong agreement between annotators (Thomson and Reiter, 2020), and the curation step used to compile the GSML provides an additional check that filters the list such that we can be confident that (for the most part) it contains only definite errors. This is, however, something that can and should be verified. There are also questions over the reported errors which were not included in the GSML, for example because only one annotator reported them. Were these false positives? Or, were they cases where the other two annotators missed the error? To investigate these issues we applied a verification process to the annotations collected when making the GSML.

7.1. The verification process

To verify previously reported errors we created an interface that showed a reported error from the annotation process, within the sentence that contains it, along with the category (or categories) that had been reported for that exact span of tokens. The interface also included an option to show the previous and subsequent sentences, in case the error depends on context from outwith the sentence. We showed these annotated errors to four participants on the Mechanical Turk platform (henceforth referred to as verifiers), who had passed the same qualification task as the gold-standard protocol. This included the three workers who provided the original annotations, although they were not told what their original annotation (if any) was. There was one new annotator, and whilst we tried to recruit more, two withdrew before doing any work. We asked them to indicate whether:

- they thought the highlight did indicate an error (yes/no).
- the given category was correct (yes, or select the correct alternative).
- they thought other annotators might disagree with them (yes/no), i.e., subjectivity.

We did this for all GSML entries that did not have full and exact agreement between annotators. Full and exact agreement occurs when all three annotators highlight the exact same span of tokens, and assign it the same category. We also checked every reported annotation that was not included in the GSML by the curation process.

Table 4: Error count by verifier agreement on the GSML (train+test).

Agreement	Assumed valid	4/4 Err	3/4 Err	2-2 Split	3/4 ¬Err	4/4 ¬Err
# Errors	1040	702	57	13	11	13
Percentage (of total)	56.64	38.24	3.10	0.71	0.60	0.71
Percentage (excluding assumed valid)	-	88.19	7.16	1.63	1.38	1.63

Table 4 shows the results of this verification process on the GSML errors. Just over 43% of the GSML was checked, with the remainder assumed valid due to existing exact agreement. Even when considering only the checked subset of 796 errors that were not assumed valid, participants agreed completely (all 4 verifiers) that over 88% of errors were valid. A further 7% had majority agreement (3 of 4 verifiers) for a valid error, with the remaining 5% divided between the verifiers being split or the majority deeming the GSML error invalid. We believe it is safe to assume that the non-checked errors would be verified at an equal or greater level due to the strict criteria of exact agreement. Valid errors (3+ verifiers agree it is an error) rise to 98% if we include those that were assumed valid and not checked. The GSML from the shared task is therefore of very high quality, although can still be improved for future use by removing or otherwise marking the small number (37) of invalid or disagreed upon errors.

7.2. Recall and precision of annotators

It is important that we also understand the level of recall and precision exhibited by either individual or groups of annotators. To determine recall for annotators who created the GSML, we considered the Verified GSML (VGSML), where the 37 errors invalidated in section 7.1 are removed. Table 5 shows that

the recall of every annotator was above 0.81, meaning that even individually they perform well. When operating in pairs, the union of annotations (by any combination) achieves a recall of above 0.95. This means that pairs of annotators are likely to find most errors in a text. The reason that the three annotator recall is not perfect was two mistakes by the curator of the GSML, where errors were missed and not included.

For precision, we considered that annotators were correct when any error they reported was considered valid by a majority of verifiers. Correct precision is awarded when the reported error has been verified by majority (regardless of whether it was in the GSML), or, the error is assumed valid, having exact and complete agreement from all three annotators when originally reported. Table 6 shows that even the lowest precision of an individual annotator was 0.916. When two or more annotators both reported the exact same error, precision was very high.

Table 5: Recall of annotator combinations.

	Recall		
Combination	**Any**	**Majority**	**All**
T1	0.817	-	-
T2	0.873	-	-
T3	0.822	-	-
T1-T2	0.978	-	0.713
T1-T3	0.962	-	0.678
T2-T3	0.958	-	0.737
T1-T2-T3	0.998	0.901	0.613

7.3. Subjectivity

We also asked verifiers whether they thought others might disagree with them, i.e., whether the error is the error subjective. The results for the response to the subjectivity question, for checked errors from the GSML, are shown

Table 6: Precision of annotator combinations

Combination	Precision	# Annotations
T1	0.967	1753
T2	0.979	1801
T3	0.916	2049
T1-T2	0.995	1298
T1-T3	0.986	1244
T2-T3	0.992	1353
T1-T2-T3	0.999	1106

Table 7: Agreement on whether *GSML* errors are subjective.

Annotator split	Yes (4-0)	Yes (3-1)	2-2 Split	No (3-1)	No (4-0)	Total
# Errors	0	7	19	73	697	796

in table 7. There were zero cases where all four verifiers felt the error was subjective, and in 88% of cases there was complete agreement that the error was not subjective. Many of the facts in these basketball summaries will not be subjective, a player either has 30 points or they do not. Whilst subjective words such as 'dominated' are important, they are less frequent.

7.4. Verification of category

Verifiers were asked if the category for an error was correct, and if not, what it should be changed to. Of the 796 checked GSML errors, 671 (84%) had complete agreement between the GSML and the verification responses, i.e, all 4 verifiers agreed on the category, and this category matched what was recorded in the GSML. There were two main types of confusion and these are shown as a confusion matrix in table 8. NUMBER^{N} errors could be confused with WORD^{W} errors. For example, with phrases like 'a pair' or 'double-double', or with ordinal numbers. NAME^{N} errors were sometimes confused

Table 8: Confusion matrix when there is not complete agreement between verifiers.

	Name	Number	Word	Context	Other	Not Checkable
Name	0	0	4	12	2	4
Number	0	0	32	0	1	27
Word	4	32	0	6	1	31
Context	12	0	6	0	0	1
Other	2	1	1	0	0	1
Not Checkable	4	27	31	1	1	0

with CONTEXT^{C} errors. This highlighted a difficult to handle edge case in the annotation protocol when the NAME^{N} might be wrong but even if it were corrected it would still be a CONTEXT^{C} error. Since our protocol did not allow overlapping spans, annotators would sometimes mark the forename as one category and the surname as another, or simply choose whichever they felt was most important. By the instructions these should be NAME^{N} errors, as the errors have a priority of NAME^{N} > NUMBER^{U} > CONTEXT^{C} > WORD^{W} > NOT CHECKABLE^{X} > OTHER^{O}, but the annotators clearly wanted to convey more information about the error. Finally, there was confusion with NOT CHECKABLE^{X} annotations. This is not about what the category should be as such, and more about whether it should be checked. Annotators were asked to only go five games back, although some were able to deduce that facts that go further back were false using only information from the last five games. For example, if a team has lost all 5 of those games, it is impossible for them to have won 4 of their last 8. The ability of annotators to do this varied.

8. Accuracy of Neural NLG systems

Despite not setting out to compare systems, the creation of the GSML does provide us with some insights into the systems of Wiseman et al. (2017), Pudup-

Table 9: Breakdown of mean errors per system, by cagtegory, for Verified GSML errors.

System	#Texts	Name	Number	Word	Context	Other	Not Checkable	Total
Wiseman	30	6.50	10.33	6.30	0.17	0.00	1.03	24.33
Puduppully	30	5.67	7.50	4.37	0.33	0.00	0.93	18.80
Rebuffel	30	5.53	4.67	4.63	1.13	0.03	0.83	16.83

Table 10: Breakdown of mean errors for human authors, by category.

System	#Texts	Name	Number	Word	Context	Other	Not Checkable	Total
Human	40	0.25	0.93	0.28	0.10	0.03	3.25	4.83

pully et al. (2019), and Rebuffel et al. (2020). The error profiles based on 30 texts from each of these systems are shown in table 9. We can see that whilst some progress has been made, particularly on **NUMBER**$^{\mathbf{U}}$ errors, all systems make on average at least 16 errors per text. The common metric of RG that was previously used on these systems returns precision of over 0.87 for the two most recent systems we used (Puduppully et al., 2019; Rebuffel et al., 2020). This is worrying as such a high score might appear to indicate higher factual accuracy than is actually present in these texts. The RG metric only predicts facts, it does not detect errors as such, which will be one cause for this difference.

9. Errors in human-authored reference texts

Humans are not perfect, nor do they always agree. Sometimes they will be under time pressure, which can lead to mistakes. In other cases, they might have strong opinions that differ from other experts (Reiter and Sripada, 2002). Knowing the number of errors in human-authored texts is essential if we are to determine whether systems can achieve human-like performance. When using machine learning to train systems, we need to know the quality of any human authored reference texts (Belz and Reiter, 2006).

To measure this, we took a set of 40 human-authored reference texts from the RotoWire dataset (different games from those previously seen by annotators).

A gold standard mistake list (GSML) was created for these 40 texts using an almost identical protocol to that defined in section 6. A total of five annotators performed this work, with each text being checked by three of them. We then transcribed these annotations to the WebAnno[8] tool before curating them to form a GSML.

9.1. Manually introduced errors

Additional errors were pseudo-randomly introduced to each document to ensure that annotators did not become complacent when checking texts. We had expected very few errors in these texts. This was done using simple rules which randomly selected between 8 and 12 sentences from a document (or all sentences if there are fewer than 8). For each selected sentence, a token position near which the error should be introduced was generated. Also generated was a preferred error type (from **NAME**$^{\mathbf{N}}$, **NUMBER**$^{\mathbf{U}}$, and **WORD**$^{\mathbf{W}}$). No more than one error was introduced to any given sentence. For **NAME**$^{\mathbf{N}}$ errors, players were replaced with another from the same team. Teams, cities, stadiums, divisions and conference were also randomly changed to other names of the same type. **NUMBER**$^{\mathbf{U}}$ errors were randomly mutated by +/- 30% (minimum of +/- 1). **WORD**$^{\mathbf{W}}$ errors were introduced manually, using only antonyms such as the words 'win' and 'positive' being replaced with 'loss' and 'negative' respectively. Some domain specific words, such as 'double-double' could also be changed to alternative but similar words like 'triple-double' and vice versa. When introducing any error, we started at the randomly selected token, then, tried to insert an error of the preferred category as close to it as possible. If no errors could be introduced, the second preferred category was used, then the the third. Only in a very small number of cases was it impossible to add an error. Care was taken to avoid introducing complex errors such as those shown in section 5.1. A total of 355 errors were Manually Introduced (MI) to a set of 40 human-authored reference texts (556 total sentences). Figure 4 shows these

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

markers in one of the texts, as well as other mistakes the annotators found.

9.2. Examining the Human GSML

In the GSML, 94% of known MI errors were present, having been found by two or more annotators. Such high recall is expected, as these were simple errors, introduced to maintain the attention of annotators. Shown in table 10 are the mean error counts per category for the human reference texts. Human-authored summaries had a high number of **NOT CHECKABLE^{X}** annotations, with a mean of 3.25 per text compared to neural systems which all had approximately one per text. This difference is likely because the human authors had access to first-hand information as well as historical insights which they used in their writing. Conversely, neural systems did not generate as much of this content as they did not have supporting data to learn such insights, including it only through hallucination. If neural systems were to achieve zero hallucinations then it would be good in terms of their design with respect to the data. However, they would not be achieving the same goal as the human-authored texts because the systems would be missing the ability to include certain key insights.

Excluding MI and **NOT CHECKABLE^{X}** annotations, the GMSL includes real errors that were identified by the annotators. A total of 63 errors were found, a mean of 1.58 per text. This was surprising, as we had assumed that these gold reference texts would be of very high quality.

In some cases, the annotators disagreed with the human authors. For example, one author described the Pacers team as 'desperate' whereas the annotators felt that it was too early in the season make this claim because the Pacers had won half of their games and there was still about one quarter of the season (19 games) to play. Another such example is seen in fig. 4 when the author claims that all bar one Pacers player was 'ice-cold' in their shooting. The annotators disagreed and commented that two other players on the team (C.J. Miles and Monta Ellis) shot fairly well. Some errors are more clear-cut, for example the first use of 'game-high' in fig. 4 is strictly wrong, Paul George had more points. The **NUMBER^{U}** error of **43^{U}** in the fourth sentence is also interesting, the

author has transposed the digits, it should be 34 minutes. We saw similar cases in other games where authors appeared to have incorrectly copied values from adjacent columns or rows, understandable forms of human error.

Annotated text:

The Charlotte Hornets lost to^{MI} the Indiana Pacers, 100-88, [**MI=W,N,U**] Monday at Spectrum Center. The Hornets (27-35) won their second straight game in convincing fashion after opening up an 11-point lead by the end of the second^{MI} quarter and never letting go of [**MI=U,N,W**] the lead. Kemba Walker lead the way with yet another great all-around performance, scoring a game-high^{W} 28 points on 10-of-22 shooting from the field to go [**MI=U,N,W**] along with nine^{MI} assists and six rebounds in 35 minutes. [**MI=U,N,W**] Nicolas Batum wasn't too far behind, scoring 24^{MI} points in 43^{U} minutes. In the end, however, it was Charlotte's defensive effort [**MI=U,N,W**] that guided them to victory, as the team tallied up eight^{MI} steals and forced the opposition into 15 total turnovers. Their opposition, the Pacers (32-30^{U}), struggled mightily offensively outside of their star Paul George, who finished with a game-high 36 points on 15-of-25 shooting from the field, 10 rebounds and four assists^{MI} [**MI=U,W,N**] in 36 minutes. Unfortunately for Charlotte^{MI}, those 36 points were nearly half of [**MI=N,W,U**] the team's total points, and his teammates were ice cold^{W} from the field. Outside of George, the Pacers shot 22-of-56 (39 percent) from the field, and their inability to support their start ultimately won^{MI} them Monday [**MI=W,U,N**] 's game. Up next, the Hornets will travel to [**MI=N,U,W**] Utah^{MI} on Wednesday to take on the surging Heat, while the Pacers will look to bounce back Wednesday against the Pistons.

List of human errors:

- game-high^{W}: Kemba Walkers only had a team-high
- 43^{U} minutes: Digits transposed, should be 34 minutes
- 30^{U}: The pacers had 31 wins following this game
- ice cold^{W}: Not all players who were claimed to be 'ice cold' had an actively bad game.

Figure 4: Annotated errors in text generated for game between Pacers and Hornets (https://www.basketball-reference.com/boxscores/201703060CHO.html). NAME^{N}, NUMBER^{U}, and WORD^{W} mistakes are highlighted in the summary, along with MI^{MI} errors and the markers containing a priority for error types that guided their creation (N=Number, U=Number, W=Word).

Whilst we expected there to be NOT CHECKABLE^{X} content, we did not expect the high level of genuine errors in these texts. In many ways they are understandable, humans do make mistakes, especially if they are in a rush. Whilst attaining human-like performance for factual accuracy is a good medium term goal for neural systems, users might be less forgiving of machine error than human error.

10. Other domains and languages

In this paper we have focused on English language generation in the sport (Basketball) domain. However, we believe that our approach can be applied to many other domains and languages.

10.1. Other languages

Our core categories will exist for all languages. Words that can be highlighted under the categories of NAME^{N} and WORD^{W} will be found in all languages, and NUMBER^{U} will be relevant for the vast majority of languages (Everett, 2012). Our category of CONTEXT^{C} is used to indicate cases where the text is strictly true, but could lead to an incorrect inference, this is not language dependant. The NOT_CHECKABLE^{X} category is not affected by language at all, it indicates that the annotator has not been able to check a factual claim in the text based on the available data. Similarly, the OTHER^{O} category is used when the annotator cannot understand what claim is being made in the text.

To illustrate this, we use our protocol to annotate German basketball summaries, produced by the the system of Puduppully et al. (2022). The annotation was performed by the first author of this paper with the assistance of machine translation tools; he is knowledgeable about the domain but does not know German. A native speaker of German checked all of the error annotations and verified that they were correct, Therefore, there may be some false negatives in the example (annotations that were missed) but there should be no false posi-

tives. As can be seen in fig. 5, there are similar types of error as were found in the English-language summaries.

We did encounter two minor issues when doing the annotation, both relating to numbers. For **103:82**$^{\mathbf{U}}$ we would have corrected only **103**$^{\mathbf{U}}$ but the string *103:82* had not been split into tokens, whereas **103 - 82**$^{\mathbf{U}}$ is split into tokens in the English summaries. This is issue of tokenization rather than language, but it is worth acknowledging that the tool-chain we use to process language can have an effect on output and its evaluation. A more interesting observation was **zweitbeste Werfer**$^{\mathbf{W}}$. The annotator felt that this player was not the second best shooter on the team, but could not highlight the number **zweit**$^{\mathbf{U}}$ in isolation as it is part of a German compound word. In order to achieve consistent annotation, annotation instructions for German probably need to specify how errors in compound nouns are handled; for example, whether a compound that contains an incorrect numeric component should be annotated as a **NUMBER**$^{\mathbf{U}}$ error, annotated as a generic **ERROR**$^{\mathbf{E}}$, or handled in some other way.

10.2. Other domains

To illustrate the use of the gold standard protocol for other domains, we first show an example annotated baseball game summary from the system of Puduppully et al. (2022). As can be seen in fig. 6, all categories of error except **OTHER**$^{\mathbf{O}}$ are found in the generation. What is interesting in this example is that there are examples of **CONTEXT**$^{\mathbf{C}}$ errors that might be confused with **WORD**$^{\mathbf{W}}$ and **NUMBER**$^{\mathbf{U}}$ errors. In the clause:

> *Prado* ***advanced***$^{\mathbf{C}}$ *to third on Jeff Francoeur 's groundout*

This could be considered an error because Prado pinch ran[9] (he was substituted in for the player who was on second base, prior to the pitch). It could be argued that not including this information could lead to an incorrect inference that there

[9] https://en.wikipedia.org/wiki/Pinch_runner

Die Milwaukee Bucks (18 - 17) schlugen die New York Knicks (5 - 31) am Mittwoch^{N} im Smoothie King Center^{N} in New York mit 103:82^{U} . Brandon Knight war der Star^{W} , indem er 17 Punkte , fünf^{U} Rebounds und zwei^{U} Assists sammelte . Giannis Antetokounmpo war der zweitbeste Werfer^{W} und kam auf 16 Punkte und zwölf Rebounds . Kein anderer Miami-Spieler , der zweistellige Punktzahlen erreichte^{O} , war Kanl Kanter^{N} und schloss mit 15 Punkten in 21 Minuten ab . New York wird versuchen , am Mittwoch^{N} in einem Auswärtsspiel gegen die Chicago Bulls^{N} . Die Knicks (5 - 31) haben jetzt fünf ihrer letzten fünf Spiele gewonnen^{W} und kämpfen immer noch^{W} um den ersten Platz im Westen . Jr Leuer^{N} lag nicht weit dahinter^{W} , denn er erzielte 15 Punkte und sieben Rebounds . Cole Als^{N} hatte eines seiner besten Spiele der Saison^{W} mit insgesamt zwölf Punkten , sieben Rebounds und sieben^{U} Assists . Die Knicks bleiben jetzt zuhause^{W} und treffen am Freitag^{N} auf die Trail Blazers^{N} . New York empfängt^{W} Charlotte^{N} am Samstag^{N} .

Figure 5: Neural generated German-language game summary for Milwaukee@New York on January 4th 2015. `https://www.basketball-reference.com/boxscores/201501040NYK.html`

was no such substitution. The errors of two^{C} near the end of the summary are because this is the first game of the season; saying a player has done something over the span of two days implies the season has been going for at least that long. In basketball summaries, almost all CONTEXT^{C} errors had been on spans containing proper nouns.

We also annotated two neural generated texts in the ToTTo dataset (Parikh et al., 2020). With this dataset, the task is for neural NLG models to generate output texts that faithfully describe highlighted cells in a Wikipedia table and their corresponding headers, Page Title and Section Title.

We made one change when applying our protocol to the ToTTo dataset, which was to explicitly annotate DATE^{D} errors. Information related to calendar dates is much more common in the *Politics* domain of ToTTo than in RotoWire or MLB (although day and month names do occur in sports journal-

ATLANTA^{N} – Chipper Jones has been waiting for his power surge^{X} . Zimmerman homered with two outs in the bottom of the ninth inning to give the Washington Nationals a 3 - 2 victory over the Atlanta Braves on Friday^{N} night . It was Zimmerman 's fourth^{U} homer of the season . It was Jones^{C} ' fourth^{U} homer of the season . Jon Rauch (1 - 0) allowed an unearned run in the top of the ninth for the Nationals , who have won four^{U} of five^{U} . Mark Teixeira doubled with one out in the ninth and moved to third^{W} on Jeff Francoeur 's groundout . Peter Moylan (0 - 1) retired the first two batters in the ninth before Zimmerman drove a 1 - 0 pitch over the wall in left^{W} for his second^{U} homer of the season . It was Zimmerman 's fourth^{U} homer of the season . It was Zimmerman 's third^{U} homer of the season . Jon Rauch (1 - 0) got the win despite allowing a one - out double to Mark Teixeira in the ninth . Prado advanced^{C} to third on Jeff Francoeur 's groundout and scored on Paul Lo Duca 's passed ball . Braves manager Fredi Gonzalez^{N} said he did n't want to take any chances^{X} . Washington took a 2 - 0 lead in the first on Nick Johnson 's two - out RBI double and Austin Kearns ' RBI single . The Nationals took a 2 - 0 lead in the bottom half . Chipper Jones tied^{W} it in the fourth with a solo shot off Odalis Perez . It was his third^{U} homer of the season and second^{U} in two^{C} days . It was his third^{U} homer in two^{C} games . Hudson^{N} allowed two runs and three hits in seven innings . Perez allowed one run and four hits in five innings .

Figure 6: Neural generated English-language game summary for Atlanta@Washington on March 30th 2008. `https://www.baseball-reference.com/boxes/WAS/WAS200803300.shtml`

ist datasets). Hence we decided to adjust the gold standard protocol to separate NAME^{N} errors into two subcategories of DATE^{D}, which includes dates, as well as names of days, months, etc., and PROPER NAME^{N}, which includes everything else that would otherwise have been classified as NAME^{N}. An example of DATE^{D} errors in the *Politics* domain is shown Table 11.

This is a good example of how the taxonomy of the gold standard protocols can be extended whilst retaining the same high-level base categories. As the protocol is applied to more domains, we expect that other extensions will also be made; for example it may make sense to explicitly annotate incorrect

medications in a medical reporting domain, and incorrect company names in a financial news domain.

Table 12 shows the second ToTTo example, from the *History* domain. Here annotated errors are very similar to the type we saw RotoWire.

Table 11: Annotation of errors for ToTTo; politics domain

Input: Table (partial) with Title, Highlighted cells and their headers

Page Title: 67th United States Congress

Section Title: Senate

State	**Senator**	**Reason for Vacancy**	**Successor**	**Date of Successor's Installation**
Pennsylvania (1)	Philander C. Knox (R)	Died October 12, 1921. Successor was appointed.	William E. Crow (R)	October 24, 1921
Pennsylvania (1)	William E. Crow (R)	Died August 2, 1922. Successor was appointed and subsequently elected.	David A. Reed (R)	August 8, 1922

Output: Annotated neural generation

Generated text: Philander C. Knox (R) served as senator until October 12, 1921 which was later served[W] by William E. Crow, Jr. from October 24, 1922[D] to August 2, 1923[D] and[W] later succeeded by David A. Reed, R, from August 8, 1932[D].

Suggested correction: Philander C. Knox (R) served as senator until October 12, 1921 before being replaced by William E. Crow, Jr. from October 24, 1921 to August 2, 1922 who was in turn succeeded by David A. Reed, R, from August 8, 1922.

Table 12: Annotation of errors for ToTTo; history domain

Input: Table with Title, Highlighted cells and their headers

Page Title: M36 tank destroyer

Section Title: Production

Month	M36	M36B1	M36B2
April 1944	25		
May 1944	100		
June 1944	120		
July 1944	155		
August 1944	100		
October 1944	75	50	
November 1944	290	93	
December 1944	348	44	
May 1945	10		50
Post-war	190		674
Total	1,413	187	724

Output: Annotated neural generation

Generated text: Production of M36[N] tank destroyers increased to 50 in October, 93 in November, and[W] 96[U] in December.

Suggested correction: Production of M36B1 tank destroyers increased to 50 in October, 93 in November, before falling to 44 in December.

11. Limitations and possible extensions

The gold standard protocol that we have detailed in this paper was designed to be as simple for annotators as possible, applicable to multiple domains and languages, and useful for any generation task where facts are to be validated against a reference. Because of the inherently high cost of robust human evaluations, we believe that extending this base protocol for specific project require-

ments is more practical than trying to exhaustively define all information that researchers might wish to capture within one monolithic framework.

11.1. Error severity

Some errors will affect the ability of the text to fulfil its purpose more than others. Whilst the base gold standard protocol treats all errors (spans of highlighted tokens) as equally severe, an additional label for severity could be recorded along with each span. For example, Moramarco et al. (2022) used a protocol where errors in a medical summary were classified as Critical or Non-Critical.

One complication is that there may be relationships between errors, i.e., the presence of two specific errors within one sentence might have more of an impact on the user than the individual errors would have. One example in the basketball domain is when statistics like rebounds cross thresholds such as 10; derived terms such as double-double are based on this. If a player had 9 rebounds then a model stating 10 could be considered more severe of an error than stating 8, even though the absolute difference is the same.

In any case, one approach could be to ask annotators to indicate whether their trust in the system is reduced by this error. For example, perhaps users mistrust systems that make numerical errors (claiming a basketball player had 20 points rather than 30 points), but still trust systems that show a different opinion to their own when describing what the data means (claiming that 30 points was a *very strong* performance, when the user thinks it is only *strong*).

The purpose of the protocol is to find individual errors, using this more granular information to rank systems can be treated as a separate research question. Evaluations that rank systems should be done with care, users are more likely to be concerned with whether a system works than what rank it holds under some academic metric.

11.2. More detailed annotation

The gold standard protocol uses a simple set of error categories on non-overlapping spans of text in order to simplify both the human annotation pro-

cess and any machine learning models that may seek to emulate the human process. More detailed information could be captured during annotation, for example whether an error was semantic or pragmatic. Syntactic trees that connect related errors could also be identified, perhaps using a parser such as spaCy (Honnibal and Johnson, 2015). The key point is that all such annotations are an extension of the base protocol.

11.3. Comparison to reference-dependant metrics

One question we are frequently asked when discussing the gold standard protocol is how it can be compared to reference-dependant metrics such as BLEU (Papineni et al., 2002). Reference-based metrics, BLEU in particular, should not be used to evaluate factual accuracy for complex data-to-text tasks. Such metrics fundamentally perform a different operation, they check whether tokens can be found in one or more reference texts. In other words, they check similarity of surface forms, they do not check semantic correctness. We have seen no evidence that BLEU or other reference-based metrics are good predictors of semantic accuracy.

12. Automatic and hybrid techniques to evaluate accuracy

Our protocol is expensive (US$30 total to annotate a 300-word text with 3 annotators), and we realise that many researchers need cheaper evaluation techniques. Such techniques could be based on fully automatic metrics to detect accuracy errors, or on hybrid approaches which use automation to reduce the amount of human effort required.

In order to encourage the development of such techniques, we organised a shared task (Thomson and Reiter, 2021) where participants submitted either fully automatic or hybrid techniques to detect accuracy errors. We assessed how well these agreed with our gold-standard protocol.

12.1. Task and Participants

The goal of the shared task was to find accuracy errors in texts that summarised basketball games; systems were asked to report the type and span of each error. Participants were given as training data 60 game summaries which had been annotated under our gold standard protocol, and were evaluated on a test set (which they did not see) of 30 additional games which had been annotated by the same method. Participants were given original system input data for each game, as well as links to the external resources (`basketball-reference.com`) that were used for fact checking. They were also given the corpus texts for each game as potential reference texts, but only one submission used these.

We had four submissions (one hybrid and three fully automatic)

- Garneau and Lamontagne (2021) proposed a hybrid process. In the first (automatic) step, a set of rules and classifiers were used to highlight potential accuracy errors. In the second (human) step, a human annotator used the results of the first step to annotate actual accuracy errors. We evaluated both the two-step process as a whole, and the first (automatic) step on its own, without human annotation.

- Kasner et al. (2021) developed an automatic metric which used a rule-based system and a semantic similarity filter to produce known-to-be-accurate sentences which are similar to the sentence being evaluated for accuracy. A model was then trained to detect accuracy errors, using as input both the sentence being assessed and the known-to-be-accurate sentences.

- Nomoto (2021) proposed an automatic metric which used an ensemble of different techniques (including rules and classifiers) to detect different kinds of accuracy errors. This system used the human corpus texts in some of its techniques.

- Rezgui et al. (2021) treated this as a fact-checking process, and developed

an automatic metric which used the three steps (which are common in fact checking) of claim identification, property identification, and claim verification.

12.2. Results

The submissions were evaluated by computing their recall and precision against the test portion of the GSML. In other words, for each submission, we calculated how many of the gold-standard mistakes were detected by that submission (recall), and how many of the mistakes detected by that submission were present as gold-standard annotations (precision). We calculated these at the level of both mistakes (overlapping spans of tokens) and individual tokens.

Table 13 shows the recall and precision of the submissions against the gold-standard manually annotated texts, for the 30 texts in the test set. We can see that Garneau and Lamontagne (2021)'s hybrid system did best. Amongst the automatic evaluations, Kasner et al. (2021)'s system had the best recall and precision.

Table 14 shows the recall/precision on different error types, for the best-performing metric overall (Kasner et al., 2021). We can see that it was unable to detect CONTEXT^{C}, NOT CHECKABLE^{X}, and OTHER^{O}, and only had around 50% precision and recall for WORD^{W} errors. Overall, this suggests that semantically more complex errors are harder to detect automatically, which is not surprising.

As a point of comparison, the Relation Generation metric (Wiseman et al., 2017), which has been widely used by many previous papers to evaluate accuracy, can only detect NAME^{N} and NUMBER^{U} errors and has a recall of less than 40% for these types of errors (Thomson and Reiter, 2020). This is considerably worse than Kasner et al. (2021)'s system.

12.3. Error analysis: What the metrics missed

When viewed as an ensemble, the three automatic metrics submitted to the shared task had a blind spot, i.e., there were some errors that no metric

Table 13: Results of the Accuracy Evaluation Shared Task for all submissions.

	Mistake		Token	
System	recall	precision	recall	precision
Garneau and Lamontagne (2021)*	0.841	0.879	0.668	0.859
Kasner et al. (2021)	0.691	0.756	0.550	0.769
Nomoto (2021)	0.523	0.494	0.349	0.505
Garneau and Lamontagne (2021)	0.503	0.334	0.410	0.397
Rezgui et al. (2021)	0.080	0.311	0.046	0.202

The * denotes the hybrid evaluation for Garneau and Lamontagne (2021)'s system. All other submissions were metrics.

picked up. This was the case for 84 mistakes in the test set (of 622). Most of these mistakes related to complex language, describing complex data; only 27 of them were simple errors of incorrect entity names or direct attributes (such as number of points). The complex mistakes included cases where two teams were being compared (26 errors). There were also 14 cases involving the phrase 'only other', for example; '*The only other*^{W} *Net to reach double figures in points was Ben McLemore*'. To be correct, this would require the named player to have double-digit points, and that all other players on their team who had the same were previously mentioned. In terms of data analytics this is one of the more complex insights that is still seen fairly often. Other mistakes involved incorrectly describing groups (duos or trios) of players. As systems improve and are better able to transcribe direct attributes such as players points, assists, and rebounds, these more complex errors are likely to become the main problem for metrics, rather than the difficult edge cases they are now.

13. Conclusion

Data-to-text NLG systems must generate texts that are factually accurate. In order to make progress on this goal, we first need to be able to measure the

Table 14: Kasner et al. (2021) per-type results.

	Mistake		Token	
Team	recall	precision	recall	precision
Name	0.750	0.846	0.759	0.862
Number	0.777	0.750	0.759	0.752
Word	0.514	0.483	0.465	0.529
Context	0.000	-	0.000	-
Not checkable	0.000	-	0.000	-
Other	0.000	-	0.000	-
Overall	0.691	0.756	0.550	0.769

accuracy of generated texts. In this paper, we have presented a human annotation protocol reliably finds factual errors in summaries of basketball games; the protocol classifies errors into six different categories (**NAME**$^{\mathbf{N}}$, **NUMBER**$^{\mathbf{U}}$, **WORD**$^{\mathbf{W}}$, **CONTEXT**$^{\mathbf{C}}$, **NOT CHECKABLE**$^{\mathbf{X}}$, and **OTHER**$^{\mathbf{O}}$). We used a verification process to check the reliability of our annotations, and how this varied with different numbers of annotators. We also discussed some of the difficulties in defining exactly what 'accuracy' means, and how we handle associated edge cases.

We used our protocol to analyse the type (based on a simple error taxonomy) and number of mistakes made by neural NLG systems and also by human corpus authors. In both cases the number of mistakes was considerably higher than we expected.

Our protocol is expensive, which we realise is a problem for many researchers, so we organised a shared task where participants were asked to propose cheaper evaluation techniques which agreed with our gold-standard protocol. Garneau and Lamontagne (2021) showed that costs could be reduced substantially using a hybrid approach which combined human and automatic processing. Several completely automatic evaluations (metrics) were submitted to our shared task; these did reasonably well at detecting **NUMBER**$^{\mathbf{U}}$ and **NAME**$^{\mathbf{N}}$ errors, but

were less effective at detecting WORD^{W} errors and were not able to detect other types of errors.

Accuracy is of course essential in most NLG applications, not just the generation of sports stories. In medicine, for example, NLG systems which generate incorrect documents at best will need to have their reports carefully checked by a human post-editor, and at worst could encourage inappropriate decisions about clinical care (Moramarco et al., 2022). We hope that the concepts and protocols which we have developed for generating sports stories can be applied to other NLG tasks where accuracy is of paramount importance.

14. Acknowledgements

We are very grateful for the hard work of the Mechanical Turk annotators who did excellent work and provided helpful feedback. We would like to thank all of the participants in the shared task, the combination of their hard work and diverse approaches has been essential to furthering understanding of the factual accuracy problem in NLG. We would also like to thank Sam Wiseman, Ratish Puduppully, and Clément Rebuffel for providing outputs from their respective systems. The constructive and insightful feedback from the two anonymous reviewers was very helpful and we greatly appreciate their input. We would also like to thank Anya Belz for checking the German translation, as well as Moray Greig, our basketball domain expert. Finally, we would like to thank members of the Aberdeen CLAN group for their advice and feedback. Craig Thomson's work on this project was supported under an EPSRC NPIF studentship grant (EP/R512412/1).

References

Arun, A., Batra, S., Bhardwaj, V., Challa, A., Donmez, P., Heidari, P., Inan, H., Jain, S., Kumar, A., Mei, S., Mohan, K., White, M., 2020. Best practices for data-efficient modeling in NLG:how to train production-ready neural models

with less data, in: Proceedings of the 28th International Conference on Computational Linguistics: Industry Track, International Committee on Computational Linguistics, Online. pp. 64–77. URL: https://aclanthology.org/2020.coling-industry.7, doi:10.18653/v1/2020.coling-industry.7.

Belz, A., Reiter, E., 2006. Comparing automatic and human evaluation of NLG systems, in: 11th Conference of the European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, Trento, Italy. pp. 313–320. URL: https://aclanthology.org/E06-1040.

Bird, S., Klein, E., Loper, E., 2009. Natural language processing with Python: analyzing text with the natural language toolkit. " O'Reilly Media, Inc.".

Dou, Y., Forbes, M., Koncel-Kedziorski, R., Smith, N.A., Choi, Y., 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), Association for Computational Linguistics, Dublin, Ireland. pp. 7250–7274. URL: https://aclanthology.org/2022.acl-long.501, doi:10.18653/v1/2022.acl-long.501.

Dušek, O., Howcroft, D.M., Rieser, V., 2019. Semantic noise matters for neural natural language generation, in: Proceedings of the 12th International Conference on Natural Language Generation, Association for Computational Linguistics, Tokyo, Japan. pp. 421–426. URL: https://aclanthology.org/W19-8652, doi:10.18653/v1/W19-8652.

Dušek, O., Novikova, J., Rieser, V., 2018. Findings of the E2E NLG challenge, in: Proceedings of the 11th International Conference on Natural Language Generation, Association for Computational Linguistics, Tilburg University, The Netherlands. pp. 322–328. URL: https://aclanthology.org/W18-6539, doi:10.18653/v1/W18-6539.

Dušek, O., Novikova, J., Rieser, V., 2020. Evaluating the state-of-the-art of end-to-end natural language generation: The e2e nlg chal-

lenge. Computer Speech and Language 59, 123–156. URL: https://www.sciencedirect.com/science/article/pii/S0885230819300919, doi:https://doi.org/10.1016/j.csl.2019.06.009.

Everett, C., 2012. A closer look at a supposedly anumeric language 1. International Journal of American Linguistics 78, 575–590. URL: http://www.jstor.org/stable/10.1086/667452.

Freitag, M., Foster, G., Grangier, D., Ratnakar, V., Tan, Q., Macherey, W., 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. URL: https://aclanthology.org/2021.tacl-1.87, doi:10.1162/tacl_a_00437.

Gardent, C., Shimorina, A., Narayan, S., Perez-Beltrachini, L., 2017. The WebNLG challenge: Generating text from RDF data, in: Proceedings of the 10th International Conference on Natural Language Generation, Association for Computational Linguistics, Santiago de Compostela, Spain. pp. 124–133. URL: https://aclanthology.org/W17-3518, doi:10.18653/v1/W17-3518.

Garneau, N., Lamontagne, L., 2021. Shared task in evaluating accuracy: Leveraging pre-annotations in the validation process, in: Proceedings of the 14th International Conference on Natural Language Generation, Association for Computational Linguistics, Aberdeen, Scotland, UK. pp. 266–270. URL: https://aclanthology.org/2021.inlg-1.26.

Gehrmann, S., Clark, E., Sellam, T., 2022. Repairing the cracked foundation: A survey of obstacles in evaluation practices for generated text. arXiv:2202.06935.

Honnibal, M., Johnson, M., 2015. An improved non-monotonic transition system for dependency parsing, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Lisbon, Portugal. pp. 1373–1378. URL: https://aclweb.org/anthology/D/D15/D15-1162.

Iso, H., Uehara, Y., Ishigaki, T., Noji, H., Aramaki, E., Kobayashi, I., Miyao, Y., Okazaki, N., Takamura, H., 2019. Learning to select, track, and generate for data-to-text, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy. pp. 2102–2113. URL: https://aclanthology.org/P19-1202, doi:10.18653/v1/P19-1202.

Kasner, Z., Mille, S., Dušek, O., 2021. Text-in-context: Token-level error detection for table-to-text generation, in: Proceedings of the 14th International Conference on Natural Language Generation, Association for Computational Linguistics, Aberdeen, Scotland, UK. pp. 259–265. URL: https://aclanthology.org/2021.inlg-1.25.

Lebret, R., Grangier, D., Auli, M., 2016. 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, Association for Computational Linguistics, Austin, Texas. pp. 1203–1213. URL: https://aclanthology.org/D16-1128, doi:10.18653/v1/D16-1128.

van der Lee, C., Gatt, A., van Miltenburg, E., Wubben, S., Krahmer, E., 2019. Best practices for the human evaluation of automatically generated text, in: Proceedings of the 12th International Conference on Natural Language Generation, Association for Computational Linguistics, Tokyo, Japan. pp. 355–368. URL: https://aclanthology.org/W19-8643, doi:10.18653/v1/W19-8643.

Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., McClosky, D., 2014. The Stanford CoreNLP natural language processing toolkit, in: Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Association for Computational Linguistics, Baltimore, Maryland. pp. 55–60. URL: https://aclanthology.org/P14-5010, doi:10.3115/v1/P14-5010.

Mathur, N., Baldwin, T., Cohn, T., 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, Association for Computational Linguistics, Online. pp. 4984–4997. URL: https://aclanthology.org/2020.acl-main.448, doi:10.18653/v1/2020.acl-main.448.

van Miltenburg, E., Clinciu, M., Dušek, O., Gkatzia, D., Inglis, S., Leppänen, L., Mahamood, S., Manning, E., Schoch, S., Thomson, C., Wen, L., 2021. Underreporting of errors in NLG output, and what to do about it, in: Proceedings of the 14th International Conference on Natural Language Generation, Association for Computational Linguistics, Aberdeen, Scotland, UK. pp. 140–153. URL: https://aclanthology.org/2021.inlg-1.14.

Moramarco, F., Papadopoulos Korfiatis, A., Perera, M., Juric, D., Flann, J., Reiter, E., Belz, A., Savkov, A., 2022. Human evaluation and correlation with automatic metrics in consultation note generation, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland. pp. 5739–5754. URL: https://aclanthology.org/2022.acl-long.394, doi:10.18653/v1/2022.acl-long.394.

Nomoto, T., 2021. Grounding NBA matchup summaries, in: Proceedings of the 14th International Conference on Natural Language Generation, Association for Computational Linguistics, Aberdeen, Scotland, UK. pp. 276–281. URL: https://aclanthology.org/2021.inlg-1.28.

Papineni, K., Roukos, S., Ward, T., Zhu, W.J., 2002. Bleu: a method for automatic evaluation of machine translation, in: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Philadelphia, Pennsylvania, USA. pp. 311–318. URL: https://aclanthology.org/P02-1040, doi:10.3115/1073083.1073135.

Parikh, A., Wang, X., Gehrmann, S., Faruqui, M., Dhingra, B., Yang, D., Das, D., 2020. ToTTo: A controlled table-to-text generation dataset, in: Pro-

ceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online. pp. 1173–1186. URL: https://aclanthology.org/2020.emnlp-main.89, doi:10.18653/v1/2020.emnlp-main.89.

Popović, M., 2020. Informative manual evaluation of machine translation output, in: Proceedings of the 28th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Barcelona, Spain (Online). pp. 5059–5069. URL: https://aclanthology.org/2020.coling-main.444, doi:10.18653/v1/2020.coling-main.444.

Portet, F., Reiter, E., Gatt, A., Hunter, J., Sripada, S., Freer, Y., Sykes, C., 2009. Automatic generation of textual summaries from neonatal intensive care data. Artificial Intelligence 173, 789–816. URL: https://www.sciencedirect.com/science/article/pii/S0004370208002117, doi:https://doi.org/10.1016/j.artint.2008.12.002.

Puduppully, R., Dong, L., Lapata, M., 2019. Data-to-text generation with content selection and planning. Proceedings of the AAAI Conference on Artificial Intelligence 33, 6908–6915. URL: https://ojs.aaai.org/index.php/AAAI/article/view/4668, doi:10.1609/aaai.v33i01.33016908.

Puduppully, R., Fu, Y., Lapata, M., 2022. Data-to-text generation with variational sequential planning. Transactions of the Association for Computational Linguistics 10, 697–715. URL: https://aclanthology.org/2022.tacl-1.40, doi:10.1162/tacl_a_00484.

Puduppully, R., Lapata, M., 2021. Data-to-text generation with macro planning. Transactions of the Association for Computational Linguistics 9, 510–527. URL: https://aclanthology.org/2021.tacl-1.31, doi:10.1162/tacl_a_00381.

Raji, I.D., Bender, E.M., Paullada, A., Denton, E., Hanna, A., 2021. Ai and the everything in the whole wide world benchmark. arXiv:2111.15366.

Rebuffel, C., Soulier, L., Scoutheeten, G., Gallinari, P., 2020. A hierarchical model for data-to-text generation, in: Jose, J.M., Yilmaz, E., Magalhães, J., Castells, P., Ferro, N., Silva, M.J., Martins, F. (Eds.), Advances in Information Retrieval, Springer International Publishing, Cham. pp. 65–80.

Reiter, E., 2007. An architecture for data-to-text systems, in: Proceedings of the Eleventh European Workshop on Natural Language Generation (ENLG 07), DFKI GmbH, Saarbrücken, Germany. pp. 97–104. URL: https://aclanthology.org/W07-2315.

Reiter, E., 2018. A structured review of the validity of BLEU. Computational Linguistics 44, 393–401. URL: https://aclanthology.org/J18-3002, doi:10.1162/coli_a_00322.

Reiter, E., Dale, R., 2000. Building Natural Language Generation Systems. Studies in natural language processing, Cambridge University Press, USA.

Reiter, E., Sripada, S., 2002. Should corpora texts be gold standards for NLG?, in: Proceedings of the International Natural Language Generation Conference, Association for Computational Linguistics, Harriman, New York, USA. pp. 97–104. URL: https://aclanthology.org/W02-2113.

Reiter, E., Sripada, S.G., Hunter, J., Yu, J., Davy, I.P., 2005. Choosing words in computer-generated weather forecasts. Artif. Intell. 167, 137–169.

Rezgui, R., Saeed, M., Papotti, P., 2021. Automatic verification of data summaries, in: Proceedings of the 14th International Conference on Natural Language Generation, Association for Computational Linguistics, Aberdeen, Scotland, UK. pp. 271–275. URL: https://aclanthology.org/2021.inlg-1.27.

Sadvilkar, N., Neumann, M., 2020. PySBD: Pragmatic sentence boundary disambiguation, in: Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS), Association for Computational Linguistics, On-

line. pp. 110–114. URL: https://aclanthology.org/2020.nlposs-1.15, doi:10.18653/v1/2020.nlposs-1.15.

Sellam, T., Das, D., Parikh, A., 2020. BLEURT: Learning robust metrics for text generation, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online. pp. 7881–7892. URL: https://aclanthology.org/2020.acl-main.704, doi:10.18653/v1/2020.acl-main.704.

Thomson, C., Reiter, E., 2020. A gold standard methodology for evaluating accuracy in data-to-text systems, in: Proceedings of the 13th International Conference on Natural Language Generation, Association for Computational Linguistics, Dublin, Ireland. pp. 158–168. URL: https://aclanthology.org/2020.inlg-1.22.

Thomson, C., Reiter, E., 2021. Generation challenges: Results of the accuracy evaluation shared task, in: Proceedings of the 14th International Conference on Natural Language Generation, Association for Computational Linguistics, Aberdeen, Scotland, UK. pp. 240–248. URL: https://aclanthology.org/2021.inlg-1.23.

Thomson, C., Reiter, E., Sripada, S., 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, Association for Computational Lingustics, Santiago de Compostela, Spain. pp. 32–40. URL: https://aclanthology.org/2020.intellang-1.4.

Thorne, J., Vlachos, A., Christodoulopoulos, C., Mittal, A., 2018a. FEVER: a large-scale dataset for fact extraction and VERification, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), Association for Computational Linguistics, New Orleans, Louisiana. pp. 809–819. URL: https://www.aclweb.org/anthology/N18-1074, doi:10.18653/v1/N18-1074.

Thorne, J., Vlachos, A., Cocarascu, O., Christodoulopoulos, C., Mittal, A., 2018b. The fact extraction and VERification (FEVER) shared task, in: Proceedings of the First Workshop on Fact Extraction and VERification (FEVER), Association for Computational Linguistics, Brussels, Belgium. pp. 1–9. URL: https://www.aclweb.org/anthology/W18-5501, doi:10.18653/v1/W18-5501.

Thorne, J., Vlachos, A., Cocarascu, O., Christodoulopoulos, C., Mittal, A., 2019. The FEVER2.0 shared task, in: Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER), Association for Computational Linguistics, Hong Kong, China. pp. 1–6. URL: https://www.aclweb.org/anthology/D19-6601, doi:10.18653/v1/D19-6601.

Van Deemter, K., Reiter, E., 2018. Lying and computational linguistics, in: Meibauer, J. (Ed.), The Oxford Handbook of Lying. Oxford University Press.

Wang, H., 2019. Revisiting challenges in data-to-text generation with fact grounding, in: Proceedings of the 12th International Conference on Natural Language Generation, Association for Computational Linguistics, Tokyo, Japan. pp. 311–322. URL: https://aclanthology.org/W19-8639, doi:10.18653/v1/W19-8639.

Wiseman, S., Shieber, S., Rush, A., 2017. Challenges in data-to-document generation, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Copenhagen, Denmark. pp. 2253–2263. URL: https://aclanthology.org/D17-1239, doi:10.18653/v1/D17-1239.

Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., Artzi, Y., 2020. Bertscore: Evaluating text generation with bert. arXiv:1904.09675.

Highlights for: Evaluating factual accuracy in complex data-to-text

- Factual accuracy problems limit the usefulness of neural solutions for complex data-to-text.
- Existing evaluation methods miss many of these errors, such as hallucination.
- We propose an evaluate a gold standard protocol for detecting factual errors in generated text.
- We show how this gold standard can be used to measure the efficacy of other methods.
- We also explore the common types of error in both human-authored and neural data-to-text systems.

Craig Thomson

Craig Thomson is a Research Assistant in the department of Computing Science at the University of Aberdeen. His interests include task definition, datasets, and evaluation methods for complex data-to-text problems, as well as reproducibility and error analysis of evaluation methods in NLP.

Ehud Reiter

Ehud Reiter is a Professor of Computing Science at the University of Aberdeen, and also Chief Scientist of Arria NLG, which he co-founded in 2009. Reiter is one of the leading world experts in Natural Language Generation (NLG). He is chair of ACL Special Interest Group on Generation (SIGGEN), has a Google Scholar H-index of 54, and writes a widely read blog on NLG (ehudreiter.com). Currently his research focuses on health-care applications of NLG, evaluation of NLG, and explainable AI.

Barkavi Sundararajan

Barkavi Sundararajan is a PhD Research Student at the University of Aberdeen focusing on Neural Language Models. Her research interests include Table-To-Text Natural Language Generation (NLG), Visual XAI tools for Neural Language Models, and Evaluation of NLG.

Credit Statements

Craig Thomson: Conceptualization, Methodology, Software, Investigation, Data Curation, Writing Original Draft, Visualisation, Project Administration. **Ehud Reiter:** Conceptualization, Methodology, Investigation, Writing Original Draft, Supervision. **Barkavi Sundararajan**: Software, Investigation, Data Curation, Writing – Original Draft

https://www.elsevier.com/authors/policies-and-guidelines/credit-author-statement

Conflict of interests statment

Declaration of interests

☐The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☒The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Craig Thomson reports financial support was provided by Engineering and Physical Sciences Research Council.