Generation Challenges: Results of the Accuracy Evaluation Shared Task

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

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

Abstract

The Shared Task on Evaluating Accuracy focused on techniques (both manual and automatic) for evaluating the factual accuracy of texts produced by neural NLG systems, in a sports-reporting domain. Four teams submitted evaluation techniques for this task, using very different approaches and techniques. The best-performing submissions did encouragingly well at this difficult task. However, all automatic submissions struggled to detect factual errors which are semantically or pragmatically complex (for example, based on incorrect computation or inference).

1 Introduction

Users expect data-to-text natural language generation (NLG) systems to generate textual summaries which are accurate. However, many NLG systems, especially neural ones, generate texts which are factually incorrect.

The most reliable way to assess the accuracy of a generated text is to ask human annotators to carefully fact-check the text. However this is a time-consuming and expensive process. In earlier work, we developed a protocol (Thomson and Reiter, 2020) where three Mechanical Turk workers (who had been screened and passed a qualifying test) carefully annotated factual errors in a text produced by a neural NLG system. The protocol was effective and showed high interannotator agreement, but it took annotators 20-30 minutes (each) to fact-check a moderately complex 300-word paragraph produced by a neural data-to-text NLG system. The total cost of the process (including fees to Amazon and money spent on the screening process for potential annotators) was about US$30 per text.

It would be very useful to the NLG community if we could come up with quicker and easier ways of measuring accuracy and factual correctness which have good correlations with the protocol of Thomson and Reiter (2020). Such methods could be based on less time-consuming human evaluations or on automatic metrics. However, these techniques should only be used if they have good agreement and correlation with careful high-quality human fact-checking by multiple annotators.

In this shared task, participating teams submitted techniques (both human and automatic) for evaluating the factual accuracy of summaries of basketball games produced from box score (and other game data) by three neural NLG systems. These techniques were evaluated by computing precision and recall (of identified factual errors) against a gold-standard human annotation produced by Thomson and Reiter (2020)'s protocol. Some of the systems did well overall, but it was also clear that some types of factual errors are difficult to detect.

We hope that our shared task encourages researchers from many fields to work on the problem of identifying factual errors in generated texts; progress in this area would be very helpful for NLG. Full details of the shared task requirements, as well as both the training and test corpus can be found at https://github.com/ehudreiter/accuracySharedTask.

2 Task

Participants were asked to submit a technique for identifying incorrect statements in a generated text. This meant statements which are not true in the real world (ie, classic fact-checking), not just statements which disagree with (or are not derivable from) the system run-time data (see Section 3.1 of Thomson and Reiter (2020)). Techniques could be

- Human evaluation protocols. Subjects would have access to data about the game and the teams, and also (if part of the protocol) to a human-authored reference text.

Proceedings of the 14th International Conference on Natural Language Generation (INLG), pages 240–248,
Aberdeen, Scotland, UK, 20-24 September 2021.

- Automatic metric (algorithm). The algorithm will have access to data about the game and the teams, and to a reference text.

- A combination of human evaluation and automatic metrics.

The output of the evaluation protocol or metric was a list of mistakes in the text. Each mistake was characterised by

- Its position in the text (start token and end token).

- A category. We use the following categories, which are based on Thomson and Reiter (2020)
  - *Incorrect number:* It does not matter whether the number is spelled out or is in digits.
  - *Incorrect name (for named entities):* In a sports reporting context, this includes people, places, teams, and days of the week.
  - *Incorrect word:* A word which is not one of the above and is incorrect.
  - *Context error:* A phrase which causes an incorrect inference because of context or discourse.
  - *Not checkable:* A statement which can not be checked, either because the information is not available or because it is too time-consuming to check.
  - *Other:* Any other type of mistake.

An example is shown in Figure 1. Note that this example combines fragments from texts produced by several different systems, along with some manual adjustments, in order to illustrate different types of mistakes in a simple way.

3 Data

We manually annotated, using the procedure of Thomson and Reiter (2020), 90 texts produced by three neural NLG systems that use basketball box score data: Wiseman et al. (2017), Puduppully et al. (2019a), and Rebuffel et al. (2020). In total, 30 texts were annotated from each system. Of these, 60 texts (20 from each system) were given to shared task participants as training data, and 30 texts (10 from each system) were reserved for a separate test set, which participants did not see until they had submitted their solutions.

Annotators were recruited on the Amazon Me-

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

List of errors:

- 2: incorrect number, should be 0.
- Monday: incorrect named entity, should be Wednesday.
- Talking Stick Resort Arena: incorrect named entity, should be US Airways Center.
- strong: incorrect word, the Grizzlies did not do well in the first half.
- out-scored: incorrect word, the Suns had a higher score in first half.
- 59: incorrect number, should be 46.
- 42: incorrect number, should be 52 .
- leading: incorrect word. Marc Gasol did not lead the Grizzlies, Mike Conley did with 24 points.
- Isaiah Thomas added: context error. Thomas played for the Suns, but context here implies he played for the Grizzlies and added to their score.
- averaging 10 points on the season so far: not checkable. This is very hard to check, since data sources report performance per season and per game, not performance up to a particular point in a season.

Figure 1: 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 .

---

chanical Turk platform. Fair treatment and compensation of workers is essential (Silberman et al., 2018), not only from an ethical standpoint, but to ensure high quality annotations. We paid annotators approximately US$20 per hour. The same three annotators marked up all 90 texts.

3.1 Systems Used

The three neural systems we used explored different ways of modifying the neural architecture. The system of Wiseman et al. (2017) defined the Rotowire task and provided initial benchmarks for machine translation systems using copy attention, it is included for this reason. Puduppully et al. (2019a) learned a document plan which was then used to generate text, whilst Rebuffel et al. (2020) used a hierarchical encoder to group attributes (such as statistics) by their respective entities (players/teams).

Other systems in this domain which could be used for evaluation include Puduppully et al. (2019b), Wang (2019), Gong et al. (2019), and Iso et al. (2019). Our aim, however, is to assess how well results produced by the participant's evaluation techniques correlate with the gold-standard fact-checking. Hence we are looking for a set of systems which generate texts that contain a significant number of accuracy errors, not complete coverage of all systems that generate texts from basketball box score data.

3.2 Multiple Correct Annotations

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

> Lou Williams led the team in scoring, dropping 30 points, six rebounds and seven assists

Suppose that it was another player, Solomon Hill, who had 30 points, 6 rebounds, and 7 assists. 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). In such cases we asked annotators to try to find the smallest number of annotations required to correct the sentence, prioritising categories in the order of Name, Number, Word, Context, Other, Not checkable. This is straightforward in this example. where the choice is correcting a single player name, or three numbers.

There were, however, a few cases where multiple complex annotations were plausible and the preferred one was not clear to our annotators. For example, in our test we encountered a sentence that was marked up by annotators as shown in Figure 2:

> **Annotator T1**: The only other Raptor to reach double figures in points was Dwyane Dragic, who came off the bench for 22 points (9-17 FG, 3-7 3Pt, 3-3 FT), six rebounds and five assists.
>
> **Annotator T2**: The only other Raptor to reach double figures in points was Dwyane Dragic, who came off the bench for 22 points (9-17 FG, 3-7 3Pt, 3-3 FT), six rebounds and five assists.
>
> **Annotator T3**: The only other Raptor to reach double figures in points was Dwyane Dragic, who came off the bench for 22 points (9-17 FG, 3-7 3Pt, 3-3 FT), six rebounds and five assists.

Figure 2: Annotations by each annotator, showing Name, Number, and Word errors.

T1 and T2 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*. Whilst this minimised Name and Word errors, it required correcting 7 of the numbers, leading to 9 errors in all, compared to the 7 errors annotated by T1 and T2.

The majority annotation (T1 and T2) was correct in this case according to our 'choose annotation with smallest number of errors'. But it is not trivial for annotators to search through multiple possible annotations looking for the optimal one, and in a larger sense it is not clear which annotation is 'correct'.

4 Accuracy Errors Observed

In this section we discuss and give some insights about the accuracy errors we observed in the manually-annotated training data (i.e, the 60 annotated texts given to participants as training data). We look separately at the different types of errors listed in section 2, and also at the impact of position

Error	Type	count	note
NUM-DIGIT	Number	270	number in digits, such as an incorrect quantity of points
TEAM	Name	162	name of team, such as *Miami Heat*
NUM-WORD	Number	130	a number spelled as a word or words
DAY-WEEK	Name	128	a day of the week, such as *Wednesday*
PLAYER	Context	50	player name (used in incorrect context)
led	Word	40	word *led*, often indicates a player led their team by some measure
a (an)	Number	34	*a* or *an* meaning the number 1
ORDINAL	Number	26	ordinal number often describing consecutive games
double-double	Word	23	word *double-double*, a basketball metric
PLAYER	Name	21	name of a player, such as *LeBron James*

Table 1: Errors that occurred at least 20 times in the training data. NUM-DIGIT, TEAM, NUM-WORD, DAY-WEEK, ORDINAL refer to types of words. Number, Name, Context, Word refer to types of errors.

and the neural NLG system used. Table 1 lists all errors that occurred at least 20 times in the training data.

4.1 Number errors

Number errors are the most common type of error in our corpus; there were 474 Number errors in the 60 texts in the training data. This category includes errors in numbers presented as digits (NUM-DIGIT), errors in spell-out numbers (NUM-WORD), and errors when a/an is used to mean the number 1.

From a semantic perspective, we can distinguish between *errors in copying numbers from the data* (eg, claiming that Smith scored 20 points when the box score data says that he scored 10 points) and *errors in calculating numbers which are not directly in the data* (eg, calculating the score at half-time, from the quarter-level scores given in the box office data). Both types of errors were common in our corpus.

4.2 Name errors

There were 317 Name errors in our corpus. TEAM, PLAYER, and DAY-WEEK (from Table 1) are all examples of a Name error. Many of these errors arose when NLG systems tried to create sentences for which they lacked data, such as the following:

> The Sixers' next game will be at **home** against the **New Orleans Pelicans** on **Wednesday**

Information about the next game is not present in the data used by the three systems which were fact-checked, so they simply guessed team and day of week, and usually guessed wrong. Of course we cannot expect a system to generate accurate texts that communicate information which is not present in the input data! But we can expect data-to-text systems to avoid sentences which communicate unavailable data.

As mentioned in subsection 3.2, sometimes a sentence could be characterised as having either a Name or a Number error. In such cases we asked annotators to make the correction which required the smallest number of changes.

4.3 Word errors

There were 334 Word errors in our corpus. They can be divided into two categories: errors in using words with clear unambiguous definitions (such as *out-scored* in Figure 1) and errors in words with fuzzy definitions (such as *strong* in Figure 1).

The most common error in a well-defined word is *double-double*. A double-double occurs when a player has ten or more (double-digits) in exactly two of the following categories: points, rebounds, assists, steals, and blocks. Note that if a player has ten or more in three of the categories, this is called a *triple-double* (3 statistics in double-digits) rather than a *double-double*. While double-double is easy to define via rules, there were 23 mistakes in our 60 corpus texts (Table 1) in the usage of this word; this seems to be a difficult concept for our systems to learn.

The most common error in a fuzzy word was *led*. *Led* appears in many contexts, for example we can say that a player *led the bench in scoring* or that a team *led at the half*.

The meaning of *led* is not clear-cut, and indeed on a few occasions the annotators disagreed on whether *led* was appropriate. This is because *led* (when used in descriptions of basketball games) can encompass rebounds, assists, steals and blocks

as well as points. For example, if one player has 25 points, 0 assists and 0 rebounds, and a second player has 22 points, 10 assists, and 10 rebounds, then the first player led in scoring, but it could be argued that the second player had a stronger performance overall, and therefore *led*. However, most of the incorrect usages of *led* marked by the annotators were in cases where all of the annotators agreed that *led* was inappropriate.

Some *ORDINAL* errors were related to this. For example, one sentence would state that a player *led*, and the subsequent sentence would state that a second player *was second*.

4.4 Context error

A Context occur occurs when a statement is literally true but misleading in context. There were 51 Context errors in our corpus, 50 of which involved PLAYERs. Typically the text would mislead the reader as to a player's status, especially which team he is playing for. An example from Figure 1 is:

> Marc Gasol scored 18 points, leading the Grizzlies. Isaiah Thomas added 15 points

Thomas scored 15 points but played for the other team (Suns). This is a Context error, since the context implies that Thomas played for the Grizzlies.

Such errors were common, the systems had a difficult time in learning when it is contextually appropriate to mention a person.

4.5 Not Checkable and Other

A Not Checkable error occurs when the annotator cannot check whether a fact is true or not. There were 37 such errors in our corpus. They usually occurred when complex statistics were given which were difficult and time-consuming for annotators to check. In order to keep the annotation task manageable, annotators were told not to look at more than 4 previous games. This made it impossible to check statements such as *he is averaging 19 points on the season so far* (from Figure 1), which requires looking at data from every previous game in the season.

We discouraged our annotators from using the Other category unless absolutely necessary, and in fact there was only one Other error in our corpus, which was the nonsensical statement *run with the first - round win of the season*.

4.6 Position analysis

In addition to analysing errors by category, we also wondered if there might be fewer errors at the beginning of the text, and more at the end. There was in fact a sharp increase in Name errors in the last sentence (Figure 3), but this was probably due to the fact that the last sentence usually described next games, and the systems did not have this information so they hallucinated. Other than this, we did not see an increase in errors later in the text. Figure 4 shows the distribution of Number errors in different positions, the other error types (excluding Name) have a similar distribution. For both of these figures, error counts are shown based upon which tenth of the summary (by token id) the error starts in.

4.7 System analysis

Last but not least, we wanted to look at the error profiles for the three systems we included in the error corpus. Two of the systems used RNN-based encoders (Wiseman et al., 2017; Puduppully et al., 2019a) and the third used a hierarchical transformer (Rebuffel et al., 2020). Table 2 shows the errors each system had within each category. The hierarchical transformer made fewer Number errors than both RNN based systems but more Context errors. It is unclear why the hierarchical encoder of (Rebuffel et al., 2020) made more Context errors, although it may be learning to better group entities with their attributes, at the expense of ordering between entities.

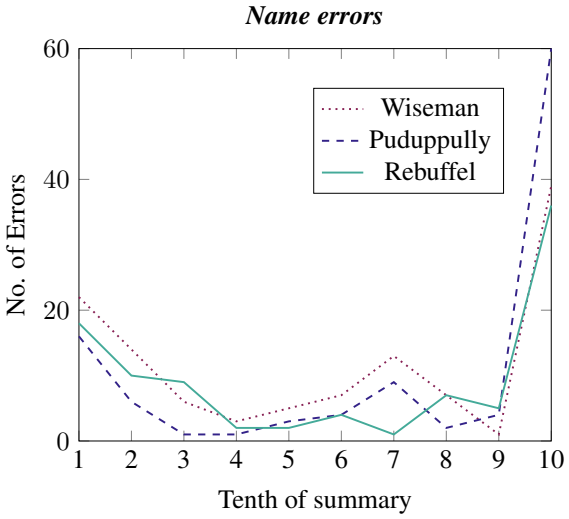

Figure 3: Name errors in different tenths of the summary.

System	encoder	NAME	NUMBER	WORD	CONTEXT	NOT_CHECK	OTHER
Wiseman	RNN	5.9	10.4	6.7	0.3	1.0	0.0
Puduppully	RNN	5.3	7.9	5.1	0.6	0.4	0.0
Rebuffel	transformer	4.7	5.5	5.0	1.7	0.5	0.1

Table 2: Breakdown of error types per-text, by system. 20 texts were included in the training corpus for each system.

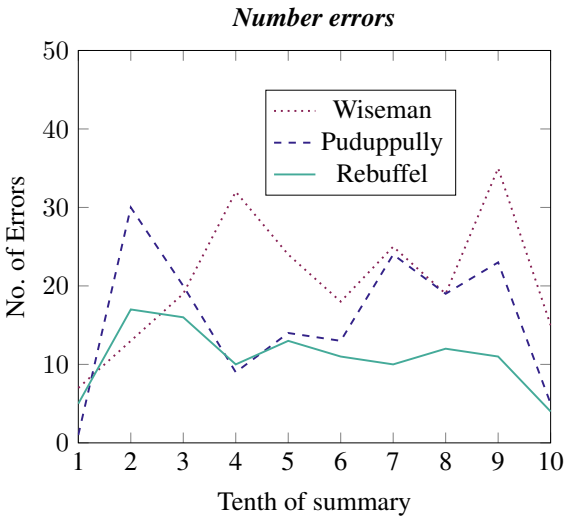

Figure 4: Number errors in different tenths of the summary.

5 Submissions

5.1 Automatic approaches

5.1.1 Charles-UPF

Charles University and Pompeu Fabra University submitted a system which detects errors using a three-stage process

1. A rule-based NLG system is used to generate sentences with facts that can be derived from the game data.

2. For each sentence in the NLG texts, a subset of the sentences in (1) is chosen based on semantic similarity to the target sentence.

3. A language model is used to identify errors. The input to the model is both the target sentence and the sentences in (2). The model is trained on synthetic data as well as the training data.

Note that the Charles-UPF system checks sentences separately, so it cannot detect errors that depend on document-level context, including Context errors and usage of '**only other**' (subsection 6.1).

5.1.2 Eurecom

The Eurecom system follows an approach inspired by earlier work on computational fact-checking (Karagiannis et al., 2020). It focuses on identifying Number errrors, and also Word errors where the word maps in a straightforward way to the game data, such as errors in the usage of '**defeated**'. A three-step process is used

1. *Claim identification:* Factual claims are extracted from the NLG text.

2. *Property identification*: The claims in (1) are expanded into full property specifications; for example the claim *18 points* is expanded with the name of the player who is supposed to have scored these points.

3. *Claim verification*: The game data is queried using the property specifications in (2); incorrect claims are flagged.

5.1.3 National Institute of Japanese Literature

The NIJL system used different approaches for different types of errors:

- *Word and Name errors*: A set of rules was used to identify Word and Name errors in the NLG texts. These rules were tuned to the structure of game summaries, with different rule used for lead, middle, and end sections of the summaries. The rules referred to the human reference texts as well as the game data.

- *Number errors*: A classifier was used to predict what relation each number represented. A co-reference tool was used to resolve referring expressions such as '*he*'.

The NIJL system was the only submission which used the human-written reference texts as well as game data when looking for accuracy errors; all other submissions just used the game data.

5.2 Hybrid approaches

5.2.1 Laval University

The Laval University approach was a hybrid system, which combined automatic analysis and human annotation.

1. *Pre-annotation*: a set of rules and classifiers are used to highlight potential accuracy errors in the NLG text. Row-column lookup on source data is used to identify potential Name and Number errors, and a multi-class, multi-label classifier is trained for Word, Context, and Not Checkable errors.

2. *Human annotation*: a single human annotator then annotated errors in the NLG text, using the pre-annotation of (1) to help them.

The human annotation was much quicker than the protocol of Thomson and Reiter (2020), because of the pre-annotation step.

We present two results for Laval University: a 'metric' result which is based purely on the results of the pre-annotation process, and a 'hybrid' result which is based on the full approach described above.

6 Results

The submissions were evaluated by computing their recall and precision against the gold-standard mistake list (GSML) which was based on the human annotated texts in the test set (section 3). 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 in the gold-standard annotation (precision). We calculated this at the level of both mistakes and tokens.

Table 3 shows the recall and precision of our submissions against the gold-standard manually annotated texts, for the 30 texts in the test set. We can see that the Laval University hybrid approach did best. Amongst the automatic evaluations, the Charles-UPF system had the best recall and precision.

Tables 4 to 8 show recall/precision of the submissions for different types of mistakes, as well as overall. We can see that the automatic techniques (Tables 5 to 8) were unable to detect Context and Other errors, and only the Laval University (metric) system could detect Not Checkable errors (but at low precision and recall). We can also see that none of the automatic systems did well at detecting Word errors; the best system, Charles-UPF, had around 50% precision and recall. 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 and Number errors and has a recall of less than 40% for these types of errors (Thomson and Reiter, 2020). This is considerably worse than the best-performing submissions to our shared task.

	Mistake		Token	
Team	recall	precision	recall	precision
Laval University*	0.841	0.879	0.668	0.859
Charles-UPF	0.691	0.756	0.550	0.769
NIJL	0.523	0.494	0.349	0.505
Laval University	0.503	0.334	0.410	0.397
Eurecom	0.080	0.311	0.046	0.202

Table 3: Results of the Accuracy Evaluation Shared Task for all submissions. The * denotes the hybrid evaluation for Laval University. All other submissions were metrics.

	Mistake		Token	
Team	recall	precision	recall	precision
Name	0.920	0.938	0.929	0.919
Number	0.862	0.881	0.832	0.854
Word	0.679	0.731	0.561	0.685
Context	0.750	0.400	0.733	0.367
Not checkable	0.237	0.391	0.073	0.615
Other	0.000	-	0.000	-
Overall	0.841	0.879	0.668	0.859

Table 4: Laval University (hybrid evaluation) 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

Table 5: Charles-UPF (metric) per-type results.

Team	Mistake		Token	
	recall	precision	recall	precision
Name	0.000	-	0.000	-
Number	0.205	0.329	0.198	0.203
Word	0.014	0.095	0.006	0.095
Context	0.000	-	0.000	-
Not checkable	0.000	-	0.000	-
Other	0.000	-	0.000	-
Overall	0.080	0.311	0.046	0.202

Table 6: Eurecom (metric) per-type results.

Team	Mistake		Token	
	recall	precision	recall	precision
Name	0.594	0.787	0.641	0.811
Number	0.442	0.351	0.427	0.340
Word	0.357	0.137	0.207	0.146
Context	0.000	-	0.000	-
Not checkable	0.500	0.190	0.200	0.407
Other	0.000	-	0.000	-
Overall	0.503	0.334	0.410	0.397

Table 7: Laval University (metric) per-type results.

6.1 Error analysis: The blind spot of metric submissions

To explore our intuition that complex errors were harder for the automatic systems to find, we performed a preliminary error analysis on the 84 mistakes (of 622) that were missed by all automatic submissions (the blind spot). We categorised each mistake based on the type of sentence that contained it:

Simple: Only 27 of the mistakes were simple, such as an incorrect attribute for an entity, or an incorrect name for a set of attributes. An example is '***Buddy Hield*** *led the second unit with a season - high 29 points , along with one assist , one rebound and one steal*', where the statistics belonged to Eric Gordon.

Comparison: 26 of the mistakes involved the comparison of how two teams fared in a quarter/half, or how their statistics compared in the game. An examples is '*The Nets got off to a quick start in this one, out-scoring the Kings 28-28 right away in the first quarter.*', where the tie of 28 points in the first quarter is incorrectly described. Many of these mistakes also involved getting the X-Y numbers wrong.

Only other: 14 of the mistakes were in clauses like '*The only other Net to reach double figures in points was Ben McLemore*'. This requires models

Team	Mistake		Token	
	recall	precision	recall	precision
Name	0.358	0.974	0.258	0.974
Number	0.696	0.419	0.672	0.419
Word	0.350	0.301	0.245	0.310
Context	0.000	-	0.000	-
Not checkable	0.000	-	0.000	-
Other	0.000	-	0.000	-
Overall	0.523	0.494	0.349	0.505

Table 8: National Institute of Japanese Literature (metric) per-type results.

and metrics to determine:

- That Ben McLemore had double-figures and was a Nets player.
- Which other Nets had double-figures.
- That all such players have been mentioned previously.

Data outwith game: 11 of the mistakes required data from outwith the game being summarised, including averages over prior games (8 mistakes), and the upcoming game schedule (3 mistakes).

Player groups: 6 mistakes incorrectly described a group of players, such as a duo.

45% of blind spot mistakes involved Word, Context, and Not-Checkable errors, compared to only 30% overall in the GSML. In addition, only 8% of blind spot mistakes were cardinal numbers, despite these constituting 33% of the GSML. It is important that we do not miss blind spot mistakes as whilst they are only 14% of the current GSML, this proportion could increase as systems become better at avoiding simple errors.

7 Conclusion

Neural data-to-text systems need to be able to produce accurate texts in order to be genuinely useful in most NLG use cases. An essential prerequisite to improving accuracy is being able to measure and evaluate accuracy.

We believe that the evaluation techniques submitted to our shared task represent a major advance in the state of the art. We encourage participants and others to continue developing better-performing techniques for this key evaluation task.

Acknowledgments

We are very grateful to all of the participants in the shared task, for their hard work in exploring very diverse approaches to the problem of finding accuracy errors! Several other teams were unable to complete a submission by our deadline; they had very interesting ideas and we encourage them to continue working on their ideas. We are also very grateful to Samira Shaikh and the members of the Aberdeen CLAN group for their help and advice. Last but not least, we are very grateful for the hard work of our Mechanical Turk annotators, in creating our training and test data. Craig Thomson's work is supported under an EPSRC NPIF studentship grant (EP/R512412/1).

References

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

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

Georgios Karagiannis, Mohammed Saeed, Paolo Papotti, and Immanuel Trummer. 2020. Scrutinizer: A mixed-initiative approach to large-scale, data-driven claim verification. *Proc. VLDB Endow.*, 13(11):2508–2521.

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

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

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

M. S. Silberman, B. Tomlinson, R. LaPlante, J. Ross, L. Irani, and A. Zaldivar. 2018. Responsible research with crowds: Pay crowdworkers at least minimum wage. *Commun. ACM*, 61(3):39–41.

Craig Thomson and Ehud Reiter. 2020. A gold standard methodology for evaluating accuracy in data-to-text systems. In *Proceedings of INLG 2020*.

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

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