Underreporting of errors in NLG output, and what to do about it

Emiel van Miltenburg,[1*] **Miruna Clinciu,**[2,3,4] **Ondřej Dušek,**[5] **Dimitra Gkatzia,**[6]
Stephanie Inglis,[7] **Leo Leppänen,**[8] **Saad Mahamood,**[9] **Emma Manning,**[10]
Stephanie Schoch,[11] **Craig Thomson,**[7] and **Luou Wen**[12]

[1]Tilburg University, [2]Edinburgh Centre for Robotics, [3]Heriot-Watt University,
[4]University of Edinburgh, [5]Charles University, Prague, [6]Edinburgh Napier University,
[7]University of Aberdeen, [8]University of Helsinki, [9]trivago N.V., [10]Georgetown University,
[11]University of Virginia, [12]University of Cambridge
Contact: c.w.j.vanmiltenburg@tilburguniversity.edu

Abstract

We observe a severe under-reporting of the different kinds of errors that Natural Language Generation systems make. This is a problem, because mistakes are an important indicator of where systems should still be improved. If authors only report overall performance metrics, the research community is left in the dark about the specific weaknesses that are exhibited by 'state-of-the-art' research. Next to quantifying the extent of error under-reporting, this position paper provides recommendations for error identification, analysis and reporting.

1 Introduction

This paper turned out very differently from the one we had initially intended to write. Our original intention was to write an overview of the different kinds of errors that appear in the output of different kinds of Natural Language Generation (NLG) systems, and to develop a general taxonomy of NLG output errors, based on the publications that have appeared at previous INLG conferences (similar to Howcroft et al. 2020; Belz et al. 2020). This, however, turned out to be impossible. The reason? There is a severe under-reporting of the different kinds of errors that NLG systems make. By this assertion, we mean that authors neither include any error analysis nor provide any examples of errors made by the system, and they do not make reference to different kinds of errors that may appear in the output. The latter is a lower bar than carrying out an error analysis, which requires a more systematic approach where several outputs are sampled and analysed for the presence of errors, which are then categorised (ideally through a formal procedure with multiple annotators). Section 3 provides more detailed statistics about error reporting in different years of INLG (and ENLG), and the amount of papers that discuss the kinds of errors that may appear in NLG output.

The fact that errors are under-reported in the NLG literature is probably unsurprising to experienced researchers in this area. The lack of reporting of negative results in AI has been a well-known issue for many years (Reiter et al., 2003). With the classic NLG example being the reporting of negative results for the STOP project on smoking cessation (Reiter et al., 2001, 2003). But even going in with (relatively) low expectations, it was confronting just to see how little we as a community look at the mistakes that our systems make.

We believe that it is both necessary *and* possible to improve our ways. One of the reasons why it is necessary to provide more error analyses (see §2.2 for more), is that otherwise, it is unclear what are the strengths and weaknesses of current NLG systems. In what follows, we provide guidance on how to gain more insight into system behavior.

This paper provides a general framework to carry out error analyses. First we cover the terminology and related literature (§2), after which we quantify the problem of under-reporting (§3). Following up on this, we provide recommendations on how to carry out an error analysis (§4). We acknowledge that there are barriers to a more widespread adoption of error analyses, and discuss some ways to overcome them (§5).

2 Background: NLG systems and errors

2.1 Defining errors

There are many ways in which a given NLG system can fail. Therefore it can be difficult to exactly define all the different types of errors that can possibly occur. Whilst error analyses in past NLG literature were not sufficient for us to create a taxonomy, we will instead propose high-level distinctions to help bring clarity within the NLG research community.

*This project was led by the first author. Remaining authors are presented in alphabetical order.

First and foremost, we define errors as countable instances of things that went wrong, as identified from the generated text.[1] We focus on text errors, where something is incorrect in the generated text with respect to the data, an external knowledge source, or the communicative goal. Through our focus on text errors, we only look at the *product* (what comes out) of an NLG system, so that we can compare the result of different kinds of systems (e.g., rule-based pipelines versus neural end-to-end systems), with error categories that are independent of the *process* (how the text is produced).[2]

By *error analysis* we mean the identification and categorisation of errors, after which statistics about the distribution of error categories are reported. It is an annotation process (Pustejovsky and Stubbs, 2012; Ide and Pustejovsky, 2017), similar to Quantitative Content Analysis in the social sciences (Krippendorff, 2018; Neuendorf, 2017).[3] Error analysis can be carried out during development (to see what kinds of mistakes the system is currently making), as the last part of a study (evaluating a new system that you are presenting), or as a standalone study (comparing different systems). The latter option requires output data to be available, ideally for both the validation and test sets. A rich source of output data is the GEM shared task (Gehrmann et al., 2021).

Text errors can be categorised in several different types, including factual errors (e.g. incorrect number; Thomson and Reiter 2020), and errors related to form (spelling, grammaticality), style (formal versus informal, empathetic versus neutral), or behavior (over- and under-specification). Some of these are universally wrong, while others may be 'contextually wrong' with respect to the task success or for a particular design goal. For example, formal texts aren't wrong *per se*, but if the goal is to produce informal texts, then any semblance of formality may be considered incorrect.

It may be possible to relate different kinds of errors to the different dimensions of text quality identified by Belz et al. (2020). What is crucial here, is that we are able to identify the specific thing which went wrong, rather than just generate a number that is representative of overall quality.

2.2 Why do authors need to report errors?

There is a need for realism in the NLG community. By providing examples of different kinds of errors, we can show the complexity of the task(s) at hand, and the challenges that still lie ahead. This also helps set realistic expectations for users of NLG technology, and people who might otherwise build on top of our work. A similar argument has been put forward by Mitchell et al. (2019), arguing for 'model cards' that provide, inter alia, performance metrics based on quantitative evaluation methods. We encourage authors to also look at the data and provide examples of where systems produce errors. Under-reporting the types of errors that a system makes is harmful because it leaves us unable to fully appreciate the system's performance.

While some errors may be detected automatically, e.g., using information extraction techniques (Wiseman et al., 2017) or manually defined rules (Dušek et al., 2018), others are harder or impossible to identify if not reported. We rely on researchers to communicate the less obvious errors to the reader, to avoid them going unnoticed and causing harm for subsequent users of the technology.

Reporting errors is also useful when comparing different implementation paradigms, such as pipeline-based data-to-text systems versus neural end-to-end systems. It is important to ask where systems fall short, because different systems may have different shortcomings. One example of this is the E2E challenge, where systems with similar human rating scores show very different behavior (Dušek et al., 2020).

Finally, human and automatic evaluation metrics, or at least the ones that generate some kind of intrinsic rating, are too coarse-grained to capture relevant information. They are general evaluations of system performance that estimate an average-case performance across a limited set of abstract dimensions (if they measure anything meaningful at all; see Reiter 2018). We don't usually know the worst-case performance, and we don't know what kinds of errors cause the metrics or ratings

[1] We use the term 'text' to refer to any expression of natural language. For example, sign language (as in Mazzei 2015) would be considered 'text' under this definition.

[2] By focusing on countable instances of things that went wrong in the output text, we also exclude issues such as bias and low output diversity, that are global properties of the collection of outputs that a system produces for a given amount of inputs, rather than being identifiable in individual outputs.

[3] There has been some effort to automate this process. For example, Shimorina et al. (2021) describe an automatic error analysis procedure for shallow surface realisation, and Stevens-Guille et al. (2020) automate the detection of repetitions, omissions, and hallucinations. However, for many NLG tasks, this kind of automation is still out of reach, given the wide range of possible correct outputs that are available in language generation tasks.

to be sub-optimal. Additionally, the general lack of extrinsic evaluations among NLG researchers (Gkatzia and Mahamood, 2015) means that in some cases we only have a partial understanding of the possible errors for a given system.

2.3 Levels of analysis

As noted above, our focus on errors in the output text is essential to facilitate framework-neutral comparisons between the performance of different systems. When categorizing the errors made by different systems, it is important to be careful with terms such as *hallucination* and *omission*, since these are process-level (pertaining to the system) rather than product-level (pertaining to the output) descriptions of the errors.[4] Process-level descriptions are problematic because we cannot reliably determine how an error came about, based on the output alone.[5] We can distinguish between at least two causes of errors, that we define below: system problems and data problems. While these problems should be dealt with, we do not consider these to be the subject of error analysis.

System problems can be defined as the malfunctioning of one or several components in a given system, or the malfunctioning of the system as a whole. System problems in rule/template-based systems could be considered as synonymous to 'bugs,' which are either semantic and/or syntactic in nature. If the system has operated in a mode other than intended (e.g., as spotted through an error analysis), the problem has to be identified, and then corrected. Identifying and solving such problems may require close involvement of domain experts for systems that incorporate significant domain knowledge or expertise (Mahamood and Reiter, 2012). Van Deemter and Reiter (2018) provide further discussion of how errors could occur at different stages of the NLG pipeline system. System problems in end-to-end systems are harder to identify, but recent work on interpretability/explainability aims to improve this (Gilpin et al., 2019).

Data problems are inaccuracies in the input that are reflected in the output. For example: when a player scored three goals in a real-world sports game, but only one goal is recorded (for whatever reason) in the data, even a perfect NLG system will generate an error in its summary of the match. Such errors may be identified as factual errors by cross-referencing the input data with external sources. They can then be further diagnosed as data errors by tracing back the errors to the data source.

3 Under-reporting of errors

We examined different *NLG conferences to determine the amount of papers that describe (types of) output errors, and the amount of papers that actually provide a manual error analysis.

3.1 Approach

We selected all the papers from three SIGGEN conferences, five years apart from each other: INLG2010, ENLG2015, and INLG2020. We split up the papers such that all authors looked at a selection of papers from one of these conferences, and informally marked all papers that discuss NLG errors in some way. These papers helped us define the terms 'error' and 'error analysis' more precisely.

In a second round of annotation, multiple annotators categorised all papers as 'amenable' or 'not amenable' to an error analysis. A paper is amenable to an error analysis if one of its primary contributions is presenting an NLG system that produces some form of output text. So, NLG experiments are amenable to an error analysis, while survey papers are not.[6] For all amenable papers, the annotator indicated whether the paper (a) mentions any errors in the output and (b) whether it contains an error analysis.[7] We encouraged discussion between annotators whenever they felt uncertain (details in Appendix A). The annotations for each paper were subsequently checked by one other annotator, after which any disagreements were adjudicated through a group discussion.

[4] Furthermore, terms like *hallucination* may be seen as unnecessary anthropomorphisms, that trivialise mental illness.

[5] A further reason to avoid process-level descriptors is that they are often strongly associated with one type of approach. For example, the term 'hallucination' is almost exclusively used with end-to-end systems, as it is common for these systems to add phrases in the output text that are not grounded in the input. In our experience, pipeline systems are hardly ever referred to as 'hallucinating.' As such, it is better to avoid the term and instead talk about concrete phenomena in the output.

[6] Examples of other kinds of papers that are not amenable include evaluation papers, shared task proposals, papers which analyze patterns in human-produced language, and papers which describe a component in ongoing NLG work which does not yet produce textual output (e.g. a ranking module).

[7] As defined in Section 2, errors are (countable) instances of something that is wrong about the output. An 'error mention' is a reference to such an instance or a class of such instances. Error analyses are formalised procedures through which annotators identify and categorise errors in the output.

Venue	Total	Amenable	Error mention	Error analysis	Percentage with error analysis
INLG2010	37	16	6	0	0%
ENLG2015	28	20	4	1	5%
INLG2020	46	35	19	4	11%

Table 1: Annotation results for different SIGGEN conferences, showing the percentage of amenable papers that included error analyses. Upon further inspection, most error mentions are relatively general/superficial.

3.2 Results

Table 1 provides an overview of our results. We found that only five papers at the selected *NLG conferences provide an error analysis,[8] and more than half of the papers fail to mention any errors in the output. This means that the INLG community is systematically under-informed about the weaknesses of existing approaches. In light of our original goal, it does not seem to be a fruitful exercise to survey all SIGGEN papers if so few authors discuss any output errors. Instead, we need a culture change where authors discuss the output of their systems in more detail. Once this practice is more common, we can start to make generalisations about the different kinds of errors that NLG systems make. To facilitate this culture change, we give a set of recommendations for error analysis.

4 Recommendations for error analysis

We provide general recommendations for carrying out an error analysis, summarized in Figure 1.

4.1 Setting expectations

Before starting, it is important to be clear about your goals and expectations for the study.

Goal Generally speaking, the goal of an error analysis is to find and quantify system errors statistically, to allow a thorough comparison of different systems, and to help the reader understand the shortcomings of your system. But your personal goals and interests may differ. For example, you may only be interested in *grammatical* errors, and less so in factual errors.

Expected errors When starting an error analysis, you may already have some ideas about what kinds of errors might appear in the outputs of different systems. These ideas may stem from the literature (theoretical limitations, or discussions of errors), from your personal experience as an NLG researcher, or it might just be an impression you have from talking to others. You might also have particular expectations about what the distribution of errors will look like.

Both goals and expectations may bias your study, and cause you to overlook particular kinds of errors. But if you are aware of these biases, you may be able to take them into account, and later check if the results confirm your original expectations. Hence, it may be useful to preregister your study, so as to make your thoughts and plans explicit (Haven and Grootel, 2019; van Miltenburg et al., 2021).

4.2 Set-up

Given your goals and expectations, there are several design choices that you have to make, in order to carry out your study.

Systems and outputs Since error analysis is relatively labor-intensive, it may not be feasible to look at a wide array of different systems. In that case, you could pre-select a smaller number of models, either based on automatic metric scores, or based on specific model features you are interested in. Alternatively, you could see to what extent the model outputs overlap, given the same input. If two models produce exactly the same output, you only need to annotate that output once.

Number of outputs Ideally, the number of outputs should be based on a power analysis to provide meaningful comparisons (Card et al., 2020; van der Lee et al., 2021), but other considerations, such as time and budget, may be taken into account.

Sample selection Regarding the selection of examples to analyze, there are three basic alternatives: The most basic is *random sampling* from the validation/test outputs. Another option is selecting *specific kinds of inputs* and analysing all corresponding outputs. Here, inputs known to be difficult/adversarial or inputs specifically targeting system properties or features may be selected (Ribeiro et al., 2020). Finally, examples to analyze may also be selected based on *quantitative values*: automatic metric scores or ratings in a preceding general human evaluation. This way, error anal-

[8] Summaries of these error analyses are in Appendix B.

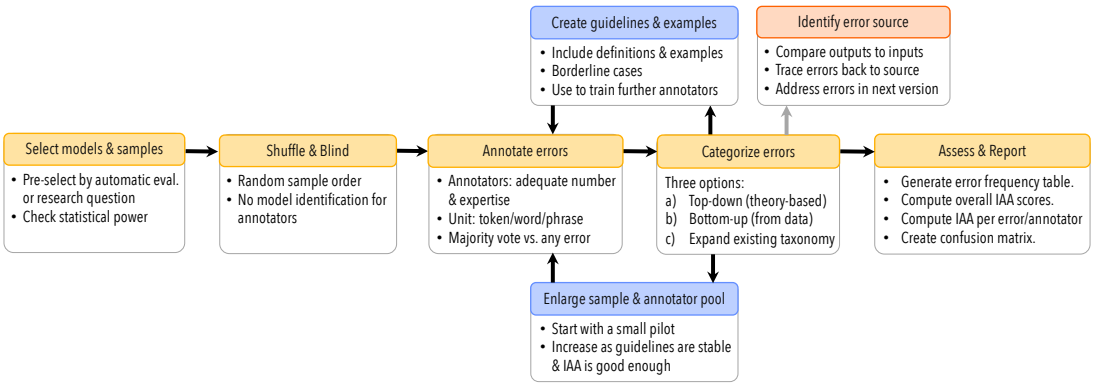

Figure 1: Flowchart depicting recommended analysis steps, as described in Section 4.

ysis can provide explanation for the automatic or human scores. The most suitable option depends on your specific use case: While random selection gives the least biased picture of the model performance, selecting specifically hard and/or low-rated samples may be more efficient. Also note that the sample selection should always be independent of any samples you may have previously examined during system development, to find out how to improve the performance of baseline models.

Presentation The order of examples for analysis should be randomized (to reduce possible order effects), and if multiple system variants are considered, the annotators must not know which system produced which output (to minimise annotator bias).

Annotators and the annotation process The annotation process can be split into two parts: identifying the errors (§4.3), and categorising the errors (§4.4). These can either be carried out sequentially (first identify, then categorize) or simultaneously (asking annotators to both identify and categorize errors at the same time). The choices you make here also impact annotator requirements, and the evaluation of the annotation procedure.

Number of annotators Generally speaking, having more annotators reduces the prevalence of the individual bias (Artstein and Poesio, 2005). This is particularly relevant if we want to detect all the errors in the output data. Having more annotators means that we are less likely to overlook individual instances of errors. Once those errors are identified, it may make more sense to rely on a smaller set of well-trained annotators to categorise the different errors. In the ideal situation, all errors are annotated by (at least) two judges so as to be able to detect and resolve any disagreements afterwards. If this is not possible, then you should double-annotate a large enough sample to reliably estimate inter-annotator agreement.[9]

Role of the annotators Ideally, the annotators should be independent of the authors reporting the error analysis (Neuendorf, 2017), to ensure that the results are not influenced by any personal biases about the systems involved, and that the annotations are indeed based on the guidelines themselves rather than on discussions between the authors. If this is not feasible, then the authors should at least ensure that they remain ignorant of the identity of the system that produced the relevant outputs.

Degree of expertise Depending on the complexity of the annotation guidelines, the error analysis may require expertise in linguistics (in the case of a theory-driven error categorisation scheme), or the relevant application area (with a context-driven error categorisation scheme). For example, Mahamood and Reiter (2012) worked with nurses to identify errors in reports generated for parents of neonatal infants. Taking into consideration the costly process of selecting domain expert annotators and the importance of quality control, non-domain experts might be also considered, ensuring their qualification through (intensive) training (Artstein and Poesio, 2005; Carlson et al., 2001).[10]

Compensation and treatment of workers If annotators are hired, either directly or via a crowdsourcing platform such as MTurk, they should be

[9] See Krippendorff 2011 for a reference table to determine the sample size for Krippendorff's α. Similar studies exist for Cohen's κ, e.g. Flack et al. 1988; Sim and Wright 2005.

[10] At least on the MTurk platform, Requesters can set the entrance requirements for their tasks such that only Workers who passed a qualifying test may carry out annotation tasks.

compensated and treated fairly (Fort et al., 2011). Silberman et al. (2018) provide useful guidelines for the treatment of crowd-workers. The authors note that they should at least be paid the minimum wage, they should be paid promptly, and they should be treated with respect. This means you should be ready to answer questions about the annotation task, and to streamline the task based on worker feedback. If you use human participants to annotate the data, you likely also need to apply for approval by an Institutional Review Board (IRB).

Training Annotators should receive training to be able to carry out the error analysis, but the amount of training depends on the difficulty of the task (which depends, among other factors, on the *coding units* (see §4.3), and the number of error types to distinguish). They should be provided with the annotation guidelines (§4.5), and then be asked to annotate texts where the errors are known (but not visible). The solutions would ideally be created by experts, although in some cases, solutions created by researchers may be sufficient (Thomson and Reiter, 2020). It should be decided in advance what the threshold is to accept annotators for the reaming work, and, if they fail, whether to provide additional training or find other candidates. Note that annotators should also be compensated for taking part in the training (see previous paragraph).

4.3 Identifying the errors

Error identification focuses on discovering all errors in the chosen output samples (as defined in the introduction). Previously, Popović (2020) asked error annotators to identify issues with comprehensibility and adequacy in machine-translated text. Similarly, Freitag et al. (2021) proposed a manual error annotation task where the annotators identified and highlighted errors within each segment in a document, taking into account the document's context as well as the severity of the errors.

The major challenge in this annotation step is how to determine the units of analysis; should annotators mark individual tokens, phrases, or constituents as being incorrect, or can they just freely highlight any sequence of words? In content analysis, this is called *unitizing*, and having an agreed-upon unit of analysis makes it easier to process the annotations and compute inter-annotator agreement (Krippendorff et al., 2016).[11] What is the right unit may depend on the task at hand, and as such is beyond the scope of this paper.[12]

A final question is what to do when there is disagreement between annotators about what counts as an error or not. When working with multiple annotators, it may be possible to use majority voting, but one might also be inclusive and keep all the identified errors for further annotation. The error categorization phase may then include a category for those instances that are not errors after all.

4.4 Categorizing errors

There are three ways to develop an error categorisation system:

1. **Top-down** approaches use existing theory to derive different types of errors. For example, Higashinaka et al. (2015a) develop an error taxonomy based on Grice's (1975) Maxims of conversation. And the top levels of Costa et al.'s (2015) error taxonomy[13] are based on general linguistic theory, inspired by Dulay et al. (1982).

2. **Bottom-up** approaches first identify different errors in the output, and then try to develop coherent categories of errors based on the different kinds of attested errors. An example of this is provided by Higashinaka et al. (2015b), who use a clustering algorithm to automatically group errors based on comments from the annotators (verbal descriptions of the nature of the mistakes that were made). Of course, you do not have to use a clustering algorithm. You can also manually sort the errors into different groups (either digitally[14] or physically).

3. **Expanding on existing taxonomies**: here we make use of other researchers' efforts to categorize different kinds of errors, by adding, removing, or merging different categories. For example, Costa et al. (2015) describe how different taxonomies of errors in Machine Translation build on each other. In NLG, if you are working on data-to-text, then you could take Thomson and Reiter's (2020) taxonomy as a starting point. For image captioning,

[11] Though note that Krippendorff et al. do provide a metric to compute inter-annotator agreement for annotators who use units of different lengths.

[12] One interesting solution to the problem of unitization is provided by Pagnoni et al. (2021), who do not identify individual errors, but do allow annotators to "check all types that apply" at the sentence level. The downside of this approach is that it is not fine-grained enough to be able to count individual instances of errors, but you do get an overall impression of the error distribution based on the sentence count for each type.

[13] Orthography, Lexis, Grammar, Semantic, and Discourse.

[14] E.g. via a program like Excel, MaxQDA or Atlas.ti, or a website like https://www.well-sorted.org.

there is a taxonomy provided by van Miltenburg and Elliott (2017).

The problem of error ambiguity To be able to categorize different kinds of errors, we often rely on the *edit-distance heuristic*. That is: we say that the text contains particular kinds of errors, because fixing those errors will give us the desired output. With this reasoning, we take the mental 'shortest path' towards the closest correct text.[15] This at least gives us a set of 'perceived errors' in the text, that provides a useful starting point for future research. However, during the process of identifying errors, we may find that there are multiple 'shortest paths' that lead to a correct utterance, resulting in error ambiguity. For example, if the output text from a sports summary system notes that Player A scored 2 points, while in fact Player A scored 1 point and Player B scored 2 points, should we say that this is a number error (2 instead of 1) or a person error (Player A instead of B)?[16] Here, one might choose to award partial credit ($1/n$ error categories), mark both types of errors as applying in this situation (overgeneralising, to be on the safe side), or count all ambiguous cases to separately report on them in the overall frequency table.

4.5 Writing annotation guidelines

Once you have determined an error identification strategy and developed an error categorisation system, you should describe these in a clear set of annotation guidelines. At the very least, these guidelines should contain relevant definitions (of each error category, and of errors in general), along with a set of examples, so that annotators can easily recognize different types of errors. For clarity, you may wish to add examples of borderline cases with an explanation of why they should be categorized in a particular way.

Pilot The development of a categorisation system and matching guidelines is an iterative process. This means that you will need to carry out multiple pilot studies in order to end up with a reliable set of guidelines,[17] that is easily understood by the annotators, and provides full coverage of the data. Pilot studies are also important to determine how long the annotation will take. This is not just practical to plan your study, but also essential to determine how much crowd-workers should be paid per task, so that you are able to guarantee a minimum wage.

4.6 Assessment

Annotators and annotations can be assessed during or after the error analysis.[18]

During the error analysis Particularly with crowd-sourced annotations it is common to include gold-standard items in the annotation task, so that it is possible to flag annotators who provide too many incorrect responses. It is also possible to carry out an intermediate assessment of inter-annotator agreement (IAA), described in more detail below. This is particularly relevant for larger projects, where annotators may diverge over time.

After the error analysis You can compute IAA scores (e.g., Cohen's κ or Krippendorff's α, see: Cohen 1960; Krippendorff 1970, 2018), to show the overall reliability of the annotations, the pairwise agreement between different annotators, and the reliability of the annotations for each error type. You can also produce a confusion matrix; a table that takes one of the annotators (or the adjudicated annotations after discussion) as a reference, and provides counts for how often errors from a particular category were annotated as belonging to any of the error categories (Pustejovsky and Stubbs, 2012). This shows all disagreements at a glance.

Any analysis of (dis)agreement or IAA scores requires there to be overlap between the annotators. This overlap should be large enough to reliably identify any issues with either the guidelines or the annotators. Low agreement between annotators may be addressed by having an adjudication round, where the annotators (or an expert judge) resolve any disagreements; rejecting the work of unreliable annotators; or revising the task or the annotation guidelines, followed by another annotation round (Pustejovsky and Stubbs, 2012).

4.7 Reporting

We recommend that authors should provide a table reporting the frequency of each error type, along with the relevant IAA scores. The main text should at least provide the overall IAA score, while IAA

[15] Note that we don't know whether the errors we identified are actually the ones that the system internally made. This would require further investigation, tracing back the origins of each different instance of an error.

[16] Similar problems have also been observed in image captioning, by Van Miltenburg and Elliott (2017).

[17] As determined by an inter-annotator agreement that exceeds a particular threshold, e.g. Krippendorff's $\alpha \geq 0.8$.

[18] And in many cases, the annotators will already have been assessed during the training phase, using the same measures.

scores for the separate error categories could also be provided in the appendix. For completeness, it is also useful to include a confusion matrix, but this can also be put in the appendix. The main text should provide a discussion of both the frequency table, as well as the IAA scores. What might explain the distribution of errors? What do the examples from the *Other*-category look like? And how should we interpret the IAA score? Particularly with low IAA scores, it is reasonable to ask why the scores are so low, and how this could be improved. Reasons for low IAA scores include: unclear annotation guidelines, ambiguity in the data, and having one or more unreliable annotator(s). The final annotation guidelines should be provided as supplementary materials with your final report.

5 (Overcoming) barriers to adoption

One reason why authors may feel hesitant about providing an error analysis is that it takes up significantly more space than the inclusion of some overall performance statistics. The current page limits in our field may be too tight to include an error analysis. Relegating error analyses to the appendix does not feel right, considering the amount of work that goes into providing such an analysis. Given the effort that goes into an error analysis, authors have to make trade-offs in their time spent doing research. If papers can easily get accepted without any error analysis, it is understandable that this additional step is often avoided. How can we encourage other NLG researchers to provide more error analyses, or even just examples of errors?

Improving our standards We should adopt reporting guidelines that stress the importance of error analysis in papers reporting NLG experiments. The NLP community is already adopting such guidelines to improve the reproducibility of published work (see Dodge et al.'s (2019) reproducibility checklist that authors for EMNLP2020 need to fill in). We should also stress the importance of error reporting in our reviewing forms; authors should be rewarded for providing insightful analyses of the outputs of their systems. One notable example here is COLING 2018, which explicitly asked about error analyses in their reviewing form for NLP engineering experiments, and had a 'Best Error Analysis' award.[19,20]

[19] https://coling2018.org/paper-types/

[20] http://coling2018.org/index.html%3Fp=1558.html

Making space for error analyses We should make space for error analyses. The page limit in *ACL conferences is already expanding to incorporate ethics statements, to describe the broader impact of our research. This suggests that we have reached the limits of what fits inside standard papers, and an expansion is warranted. An alternative is to publish more journal papers, where there is more space to fit an error analysis, but then we as a community also need to encourage and increase our appreciation of journal submissions.

Spreading the word Finally, we should inform others about how to carry out a proper error analysis. If this is a problem of exposure, then we should have a conversation about the importance of error reporting. This paper is an attempt to get the conversation started.

6 Follow-up work

What should you do after you have carried out an error analysis? We identify three directions for follow-up studies.

Errors in inputs An additional step can be added during the identification of errors which focuses on observing the system inputs and their relation to the errors. Errors in the generated text may occur due to semantically noisy (Dušek et al., 2019) or incorrect system input (Clinciu et al., 2021); for instance, input data values might be inaccurate or the input might not be updated due to a recent change (e.g., new president). To pinpoint the source of the errors, we encourage authors to look at their input data jointly with the output, so that errors in inputs can be identified as such.

Building new evaluation sets Once you have identified different kinds of errors, you can try to trace the origin of those errors in your NLG model, or posit a hypothesis about what kinds of inputs cause the system to produce faulty output. But how can you tell whether the problem is really solved? Or how can you stimulate research in this direction? One solution, following McCoy et al. (2019), is to construct a new evaluation set based on the (suspected) properties of the errors you have identified. Future research, knowing the scope of the problem from your error analysis, can then use this benchmark to measure progress towards a solution.

Scales and types of errors Error types and human evaluation scales are closely related. For ex-

ample, if there are different kinds of grammatical errors in a text, we expect human grammaticality ratings to go down as well. But the relation between errors and human ratings is not always as transparent as with grammaticality. Van Miltenburg et al. (2020) shows that different kinds of semantic errors have a different impact on the perceived overall quality of image descriptions.[21] Future research should aim to explore the connection between the two in more detail, so that there is a clearer link between different kinds of errors and different quality criteria (Belz et al., 2020).

7 Conclusion

Having found that NLG papers tend to underreport errors, we have motivated why authors should carry out error analyses, and provided a guide on how to carry out such analyses. We hope that this paper paves the way for more in-depth discussions of errors in NLG output.

Acknowledgements

We would like to thank the anonymous reviewers for their insightful feedback. Dimitra Gkatzia's contribution was supported under the EPSRC projects CiViL (EP/T014598/1) and NLG for Low-resource Domains (EP/T024917/1). Miruna Clinciu's contribution is supported by the EPSRC Centre for Doctoral Training in Robotics and Autonomous Systems at Heriot-Watt University and the University of Edinburgh. Miruna Clinciu's PhD is funded by Schlumberger Cambridge Research Limited (EP/L016834/1, 2018-2021). Ondřej Dušek's contribution was supported by Charles University grant PRIMUS/19/SCI/10. Craig Thomson's work is supported under an EPSRC NPIF studentship grant (EP/R512412/1).

References

Imen Akermi, Johannes Heinecke, and Frédéric Herledan. 2020. Tansformer based natural language generation for question-answering. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 349–359, Dublin, Ireland. Association for Computational Linguistics.

Ron Artstein and Massimo Poesio. 2005. Bias decreases in proportion to the number of annotators. *Proceedings of FG-MoL*.

Cristina Barros and Elena Lloret. 2015. Input seed features for guiding the generation process: A statistical approach for Spanish. In *Proceedings of the 15th European Workshop on Natural Language Generation (ENLG)*, pages 9–17, Brighton, UK. Association for Computational Linguistics.

David Beauchemin, Nicolas Garneau, Eve Gaumond, Pierre-Luc Déziel, Richard Khoury, and Luc Lamontagne. 2020. Generating intelligible plumitifs descriptions: Use case application with ethical considerations. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 15–21, Dublin, Ireland. Association for Computational Linguistics.

Anya Belz, Simon Mille, and David M. Howcroft. 2020. Disentangling the properties of human evaluation methods: A classification system to support comparability, meta-evaluation and reproducibility testing. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 183–194, Dublin, Ireland. Association for Computational Linguistics.

Emily M. Bender, Leon Derczynski, and Pierre Isabelle, editors. 2018. *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, Santa Fe, New Mexico, USA.

Dallas Card, Peter Henderson, Urvashi Khandelwal, Robin Jia, Kyle Mahowald, and Dan Jurafsky. 2020. With little power comes great responsibility. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9263–9274, Online. Association for Computational Linguistics.

Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. 2001. Building a discourse-tagged corpus in the framework of Rhetorical Structure Theory. In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue - Volume 16*, SIGDIAL '01, page 1–10.

Miruna-Adriana Clinciu, Dimitra Gkatzia, and Saad Mahamood. 2021. It's commonsense, isn't it? demystifying human evaluations in commonsense-enhanced NLG systems. In *Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval)*, pages 1–12, Online. Association for Computational Linguistics.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Ângela Costa, Wang Ling, Tiago Luís, Rui Correia, and Luísa Coheur. 2015. A linguistically motivated taxonomy for machine translation error analysis. *Machine Translation*, 29(2):127–161.

Kees van Deemter and Ehud Reiter. 2018. Lying and computational linguistics. In Jörg Meibauer, editor,

[21] Relatedly, Freitag et al. (2021) ask annotators to rate the severity of errors in machine translation output, rather than simply marking errors.

The Oxford Handbook of Lying. Oxford University Press.

Jesse Dodge, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A. Smith. 2019. Show your work: Improved reporting of experimental results. 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 2185–2194, Hong Kong, China. Association for Computational Linguistics.

Heidi C. Dulay, Marina K. Burt, and Stephen D. Krashen. 1982. *Language two*. New York : Oxford University Press.

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

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2018. Findings of the E2E NLG challenge. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 322–328, Tilburg University, The Netherlands. Association for Computational Linguistics.

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2020. Evaluating the State-of-the-Art of End-to-End Natural Language Generation: The E2E NLG Challenge. *Computer Speech & Language*, 59:123–156.

Virginia F Flack, AA Afifi, PA Lachenbruch, and HJA Schouten. 1988. Sample size determinations for the two rater kappa statistic. *Psychometrika*, 53(3):321–325.

Karën Fort, Gilles Adda, and K. Bretonnel Cohen. 2011. Last words: Amazon Mechanical Turk: Gold mine or coal mine? *Computational Linguistics*, 37(2):413–420.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, errors, and context: A large-scale study of human evaluation for machine translation.

Sebastian Gehrmann, Tosin Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Aremu Anuoluwapo, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna Clinciu, Dipanjan Das, Kaustubh D. Dhole, Wanyu Du, Esin Durmus, Ondřej Dušek, Chris Emezue, Varun Gangal, Cristina Garbacea, Tatsunori Hashimoto, Yufang Hou, Yacine Jernite, Harsh Jhamtani, Yangfeng Ji, Shailza Jolly, Mihir Kale, Dhruv Kumar, Faisal Ladhak, Aman Madaan, Mounica Maddela, Khyati Mahajan, Saad Mahamood, Bodhisattwa Prasad Majumder, Pedro Henrique Martins, Angelina McMillan-Major, Simon Mille, Emiel van Miltenburg, Moin Nadeem, Shashi Narayan, Vitaly Nikolaev, Rubungo Andre Niyongabo, Salomey Osei, Ankur Parikh, Laura Perez-Beltrachini, Niranjan Ramesh Rao, Vikas Raunak, Juan Diego Rodriguez, Sashank Santhanam, João Sedoc, Thibault Sellam, Samira Shaikh, Anastasia Shimorina, Marco Antonio Sobrevilla Cabezudo, Hendrik Strobelt, Nishant Subramani, Wei Xu, Diyi Yang, Akhila Yerukola, and Jiawei Zhou. 2021. The gem benchmark: Natural language generation, its evaluation and metrics.

Leilani H. Gilpin, David Bau, Ben Z. Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. 2019. Explaining explanations: An overview of interpretability of machine learning. In *Proceedings - 2018 IEEE 5th International Conference on Data Science and Advanced Analytics, DSAA 2018*.

Dimitra Gkatzia and Saad Mahamood. 2015. A Snapshot of NLG Evaluation Practices 2005-2014. In *Proceedings of the 15th European Workshop on Natural Language Generation (ENLG)*, pages 57–60.

H. P. Grice. 1975. *Logic and Conversation*, pages 41 – 58. Brill, Leiden, The Netherlands.

Tamarinde L. Haven and Dr. Leonie Van Grootel. 2019. Preregistering qualitative research. *Accountability in Research*, 26(3):229–244. PMID: 30741570.

Ryuichiro Higashinaka, Kotaro Funakoshi, Masahiro Araki, Hiroshi Tsukahara, Yuka Kobayashi, and Masahiro Mizukami. 2015a. Towards taxonomy of errors in chat-oriented dialogue systems. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 87–95, Prague, Czech Republic. Association for Computational Linguistics.

Ryuichiro Higashinaka, Masahiro Mizukami, Kotaro Funakoshi, Masahiro Araki, Hiroshi Tsukahara, and Yuka Kobayashi. 2015b. Fatal or not? finding errors that lead to dialogue breakdowns in chat-oriented dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2243–2248, Lisbon, Portugal. Association for Computational Linguistics.

David M. Howcroft, Anya Belz, Miruna-Adriana Clinciu, Dimitra Gkatzia, Sadid A. Hasan, Saad Mahamood, Simon Mille, Emiel van Miltenburg, Sashank Santhanam, and Verena Rieser. 2020. Twenty years of confusion in human evaluation: NLG needs evaluation sheets and standardised definitions. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 169–182, Dublin, Ireland. Association for Computational Linguistics.

Nancy Ide and James Pustejovsky, editors. 2017. *Handbook of linguistic annotation*. Springer. ISBN 978-94-024-1426-4.

Taichi Kato, Rei Miyata, and Satoshi Sato. 2020. BERT-based simplification of Japanese sentence-ending predicates in descriptive text. In *Proceedings*

of the 13th International Conference on Natural Language Generation, pages 242–251, Dublin, Ireland. Association for Computational Linguistics.

Klaus Krippendorff. 1970. Bivariate agreement coefficients for reliability of data. *Sociological methodology*, 2:139–150.

Klaus Krippendorff. 2011. Agreement and information in the reliability of coding. *Communication Methods and Measures*, 5(2):93–112.

Klaus Krippendorff. 2018. *Content Analysis: An Introduction to Its Methodology*, fourth edition edition. SAGE Publications, Inc.

Klaus Krippendorff, Yann Mathet, Stéphane Bouvry, and Antoine Widlöcher. 2016. On the reliability of unitizing textual continua: Further developments. *Quality & Quantity*, 50(6):2347–2364.

Zewang Kuanzhuo, Li Lin, and Zhao Weina. 2020. SimpleNLG-TI: Adapting SimpleNLG to Tibetan. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 86–90, Dublin, Ireland. Association for Computational Linguistics.

Chris van der Lee, Albert Gatt, Emiel van Miltenburg, and Emiel Krahmer. 2021. Human evaluation of automatically generated text: Current trends and best practice guidelines. *Computer Speech & Language*, 67:101–151.

Saad Mahamood and Ehud Reiter. 2012. Working with clinicians to improve a patient-information NLG system. In *INLG 2012 Proceedings of the Seventh International Natural Language Generation Conference*, pages 100–104.

Alessandro Mazzei. 2015. Translating Italian to LIS in the rail stations. In *Proceedings of the 15th European Workshop on Natural Language Generation (ENLG)*, pages 76–80, Brighton, UK. Association for Computational Linguistics.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Emiel van Miltenburg and Desmond Elliott. 2017. Room for improvement in automatic image description: an error analysis. *CoRR*, abs/1704.04198.

Emiel van Miltenburg, Chris van der Lee, and Emiel Krahmer. 2021. Preregistering NLP research. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 613–623, Online. Association for Computational Linguistics.

Emiel van Miltenburg, Wei-Ting Lu, Emiel Krahmer, Albert Gatt, Guanyi Chen, Lin Li, and Kees van Deemter. 2020. Gradations of error severity in automatic image descriptions. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 398–411, Dublin, Ireland. Association for Computational Linguistics.

Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. 2019. Model cards for model reporting. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 220–229.

Adrian Muscat and Anja Belz. 2015. Generating descriptions of spatial relations between objects in images. In *Proceedings of the 15th European Workshop on Natural Language Generation (ENLG)*, pages 100–104, Brighton, UK. Association for Computational Linguistics.

Kimberly A. Neuendorf. 2017. *The Content Analysis Guidebook*. SAGE Publications. Second edition, ISBN 9781412979474.

Jason Obeid and Enamul Hoque. 2020. Chart-to-text: Generating natural language descriptions for charts by adapting the transformer model. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 138–147, Dublin, Ireland. Association for Computational Linguistics.

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4812–4829, Online. Association for Computational Linguistics.

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

James Pustejovsky and Amber Stubbs. 2012. *Natural Language Annotation for Machine Learning: A guide to corpus-building for applications*. " O'Reilly Media, Inc.".

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

Ehud Reiter, Roma Robertson, A. Scott Lennox, and Liesl Osman. 2001. Using a randomised controlled clinical trial to evaluate an NLG system. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 442–449, Toulouse, France. Association for Computational Linguistics.

Ehud Reiter, Roma Robertson, and Liesl M. Osman. 2003. Lessons from a failure: Generating tailored smoking cessation letters. *Artificial Intelligence*, 144(1):41–58.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Anastasia Shimorina, Yannick Parmentier, and Claire Gardent. 2021. An error analysis framework for shallow surface realisation. *Transactions of the Association for Computational Linguistics*, 9.

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.

Julius Sim and Chris C Wright. 2005. The Kappa Statistic in Reliability Studies: Use, Interpretation, and Sample Size Requirements. *Physical Therapy*, 85(3):257–268.

Symon Stevens-Guille, Aleksandre Maskharashvili, Amy Isard, Xintong Li, and Michael White. 2020. Neural NLG for methodius: From RST meaning representations to texts. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 306–315, Dublin, Ireland. Association for Computational Linguistics.

Mariët Theune, Ruud Koolen, and Emiel Krahmer. 2010. Cross-linguistic attribute selection for REG: Comparing Dutch and English. In *Proceedings of the 6th International Natural Language Generation Conference*.

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

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.

A Annotation

A.1 Procedure and definitions

We annotated all papers from INLG2010, ENLG2015, and INLG2020 in two rounds. Round 1 was an informal procedure where we generally checked whether the papers mentioned any errors at all (broadly construed, without defining the term 'error'). Following this, we determined our formal annotation procedure, based on the example papers: first check if the paper is amenable. If so, check if it (a) mentions any errors in the output or (b) contains an error analysis. We used the following definitions:

Amenable A paper is amenable to an error analysis if one of its primary contributions is presenting an NLG system that produces some form of output text. So, NLG experiments are amenable to an error analysis, while survey papers are not.

Error Errors are (countable) instances of something that is wrong about the output.

Error mention An 'error mention' is a reference to such an instance or a class of such instances.

Error analysis Error analyses are defined as formalised procedures through which annotators identify and categorise errors in the output.

A.2 Discussion points

The most discussion took place on the topic of amenability. Are papers that just generate prepositions (Muscat and Belz, 2015) or attributes for referring expressions (Theune et al., 2010) amenable to error analysis? And what about different versions of SimpleNLG? (E.g., Kuanzhuo et al. 2020.) Although these topics feel different from, say, data-to-text systems, we believe it should be possible to carry out an error analysis in these contexts as well. In the end, amenability for us is just an artificial construct to address the (potential) criticism that we cannot just report the amount of error analyses as a proportion of all *NLG papers. As such, our definition for amenability is just a quick heuristic. Determining whether a paper really benefits from an error analysis is a more complex issue, that depends on many contextual factors.

B Papers containing error analyses

Below is a brief summary of the error analyses that we found in our annotation study.

1. Barros and Lloret (2015) investigate the use of different seed features for controlled neural NLG. They analysed all the outputs of their model, and categorised them based on existing lists of common grammatical errors and drafting errors.

2. Akermi et al. (2020) explore the use of pre-trained transformers for question-answering. They

conducted a human evaluation study, asking 20 native speakers to indicate the presence of errors in the outputs of a French and English system. These errors were categorised as: *extra words, grammar, missing words, wrong preposition, word order*.

3. Beauchemin et al. (2020) aim to generate explanations of *plumitifs* (dockets), based on the text of the dockets themselves. Following the identification of different errors (defined by the authors as "the lack of realizing a specific part (accused, plaintiff or list of charges paragraphs), instead of evaluating the textual generation," they trace the source of the error back to either an earlier information extraction step, or to the generation procedure.

4. Kato et al. (2020) present a BERT-based approach to simplify Japanese sentence-ending predicates. They took a bottom-up approach to classify the 140 cases where their model could not generate any acceptable cases. The authors then relate the error types to different stages of the generation process, and to the general architecture of their system.

5. Obeid and Hoque (2020) present a neural NLG model for automatically providing natural language descriptions of information visualisations (i.e., charts). They manually assessed 50 output examples, and highlighted the different errors in the text. The authors find that, despite their efforts to prevent it, their model still suffers from hallucination. They identify two kinds of hallucination: either the model associates an existing value with the wrong data point, or it simply predicts an irrelevant token.

A **notable exception** is the paper by Thomson and Reiter (2020), who carry out an error analysis of existing output data from three different systems. This paper was not considered amenable, because it does not present an NLG system of its own, and thus it was not included in our counts. But even if we were to count this paper among the error analyses, the trend remains the same: very few papers discuss errors in NLG output.