Evaluation rules!
On the use of grammars and rule-based systems for NLG evaluation

Emiel van Miltenburg[1], **Chris van der Lee**[1], **Thiago Castro-Ferreira**[1,2], and **Emiel Krahmer**[1]
[1]Tilburg center for Cognition and Communication, Tilburg University
[2]Federal University of Minas Gerais (UFMG), Brazil
C.W.J.vanMiltenburg@tilburguniversity.edu

Abstract

NLG researchers often use uncontrolled corpora to train and evaluate their systems, using textual similarity metrics, such as BLEU. This position paper argues in favour of two alternative evaluation strategies, using grammars or rule-based systems. These strategies are particularly useful to identify the strengths and weaknesses of different systems. We contrast our proposals with the (extended) WebNLG dataset, which is revealed to have a skewed distribution of predicates. We predict that this distribution affects the quality of the predictions for systems trained on this data. However, this hypothesis can only be thoroughly tested (without any confounds) once we are able to systematically manipulate the skewness of the data, using a rule-based approach.

1 Introduction

Recent years have seen many Natural Language Generation (NLG) researchers move away from rule-based systems, and towards neural end-to-end systems. These systems are typically evaluated using textual similarity metrics like BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005; Denkowski and Lavie, 2014), or ROUGE (Lin, 2004), on large corpora of crowd-sourced texts (e.g., the E2E dataset, Novikova et al. 2016; the WebNLG dataset, Gardent et al. 2017; or MS COCO, Lin et al. 2014). This evaluation strategy tells us to what extent the generated texts are similar to the reference data, but it is often difficult to determine exactly what that resemblance buys us. By now it is well-known that BLEU correlates poorly with human ratings (Elliott and Keller, 2014; Kilickaya et al., 2017; Reiter, 2018; Sulem et al., 2018; Mathur et al., 2020), but BLEU by itself also does not tell us anything about the strengths and weaknesses of a particular model, or model architecture. This paper argues that we need alternative (or at least additional) metrics to provide this kind of insight. We believe that rule-based approaches are well-suited for this task.

1.1 Not just BLEU; also uncontrolled data

BLEU is an easy target; it's a quick-and-dirty solution that ignores paraphrases and different-but-valid perspectives on the input data. But if we only look at the metrics, we miss the elephant in the room: the corpora we use to train NLG systems are the messy result of underspecified elicitation tasks, where annotators receive very little training as to what the outputs should look like (e.g., van Miltenburg 2016; van Miltenburg et al. 2017). Ideally, we should want training data that conforms to a clear set of guidelines. Having clean data is a means to control the quality of the output of an NLG system. By using crowdsourcing, we have ceded that control to the crowd.[1] The problem with crowdsourcing, and particularly with elicitation tasks to create NLG corpora, is that quality control is difficult. And even if we can control the quality of the data, it is very hard to control the diversity of the generated texts.[2] This makes it harder to study the ability of NLG systems to generalise from the training data to unseen instances. We will argue (in Section 4) that we need a more systematic approach to produce NLG test benches. We believe

[1]Although there are also benefits to having a more uncontrolled elicitation task. For example, having fewer constraints means that the resulting data will be more diverse.

[2]This is not just a problem in NLG. Freitag et al. (2020, and references therein) describe how human translators tend to produce *translationese*: translations that overly rely on the source text, resulting in less natural-sounding texts. This reduces the diversity of the evaluation data for Machine Translation (MT), which has strong effects on the evaluation metrics used in MT (Freitag et al., 2019). The authors go on to show that we can improve the correlation between modern evaluation metrics and human ratings, by improving the reference data (in this case: asking linguists to generate more fluent and diverse translations). But of course, this kind of exercise is expensive and time-consuming.

that a rule-based approach (combined with new or existing NLG data) would again be ideal.

1.2 The downside of end-to-end systems; opportunities for rule-based approaches

There are many good reasons to develop end-to-end systems. For example, Dušek et al. (2020) found that, in the E2E-challenge (Novikova et al., 2017), sequence-to-sequence models "scored higher than other architectures on word-overlap-based metrics and human-rated naturalness."[3] However, given the above, we can also see the move away from rule-based systems as a means to evade responsibility for whatever output our NLG systems produce. After all: the crowd decides what the output should look like. If we don't explicitly tell our NLG systems what to do (via rules), we should find other ways to control what the output should look like. And what better way to control and evaluate the output... than to use more rules? This paper presents some ways in which rules and rule-based systems can be used to improve current-day NLG research.[4]

2 Evaluation and cognitive capacities

Ideally, evaluation of NLG systems should be tied to the cognitive capacities those systems are claimed to possess (Schlangen, 2019, 2020).[5] For example, one could evaluate whether a system is able to produce a grammatically correct sentence. Abilities like these can be formalised as a set of rules (cf. Chomsky's generative program; Chomsky 1965), and we could simply check whether the output of an NLG system conforms to a predefined grammar. Xie et al. (2019) do this using Flickinger's (2000; 2011) English Resource Grammar, which offers broad coverage of the English language. The DELPH-IN catalogue offers

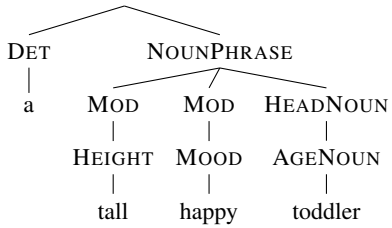

Figure 1: Parse tree for the phrase: *a tall happy toddler*.

an overview of HPSG grammars (Pollard and Sag, 1994) that are available for other languages.[6] In related work, Bangalore et al. (2000) use automated parsers for evaluation, but they compare the parse trees for the system outputs with those of the reference data, and compute an accuracy metric.

At this point, it is fair to say that not all languages are as well-resourced as English. Of course we can evaluate grammaticality if there is a relatively complete description of a language. But not everyone has that luxury. Moreover, as NLG researchers, we aren't *just* interested in grammaticality. Why should we care about grammars, then?

There are two ways to respond to this criticism. First, if you accept that NLG evaluation is a good use case for a broad-coverage parser, then that provides additional motivation to build a new or better parser (or to start talking to linguists in your area). Second, a grammar does not necessarily have to cover the *entire* language for it to be useful. It just needs to cover your domain of interest. One example grammar is provided by Van Miltenburg et al. (2018), who developed a context-free grammar to cover person-descriptions in the MS COCO dataset (a collection of images paired with image descriptions). The grammar has a set of production rules that describe as well as categorise the components of person-descriptions in the corpus. So the phrase 'the tall happy toddler' can be parsed as in Figure 1. Van Miltenburg (2020) improved this grammar, and used it to evaluate the extent to which image captioning systems are able to generate different kinds of person-descriptions. This is another example of a cognitive ability that can be examined using a pre-determined set of rules. As an additional bonus, a complete characterisation of a domain such as PERSON-DESCRIPTIONS allows us to reflect on what kinds of outputs are desirable or not, for the system to generate.

[3] Another advantage of neural end-to-end systems that is sometimes mentioned is development speed: if you have a training corpus, you can train an end-to-end system fairly quickly. But Reiter (2020) shows that this advantage is probably overstated (if not false). Elsewhere he remarks that 'effectively impossible to fix undesirable behaviour in a "deep learning" system' (Reiter, 2016), meaning such a system would have to be re-trained if any changes need to be made to its output. This makes maintenance very time-consuming.

[4] Code for this paper is available at: https://github.com/evanmiltenburg/EvaluationRules.

[5] This may not always be the case. Or at least: not directly. For example, consider the question whether a system is *user-friendly* or *pleasant to use*. Some high-level properties are fairly subjective, and may best be evaluated using human ratings. Still one could argue that these properties may be decomposed into a set of different abilities. For example: using the correct register, being able to translate jargon into layman's terms, generating unambiguous descriptions.

[6] http://moin.delph-in.net/GrammarCatalogue

3 Systems, models, and architectures

Before we continue, it is important to recognise the difference between systems, models and architectures. We consider **architectures** to be abstract descriptions of all the components that make up a system. A **system** is a specific instance that implements an architecture. When a system is trained on a particular dataset, we can say that it has constructed a **model** for how the task should be carried out. These distinctions are important, because different NLG researchers may be interested in either systems, models, or architectures. Theoretically oriented researchers may be more interested in the properties of different architectures, whereas more applied researchers may be interested in the properties of different systems or models. In our experience, shared tasks are often misconstrued as a competition to see who can deliver the best model. This misses the point, because ideally the results of a shared task teach us about the strengths and weaknesses of different architectures.[7]

4 Evaluating the ability to generalise

Machine learning datasets are used to determine whether systems are able to generalise from experience to unseen situations (Mitchell, 1997). To test this, researchers typically use separate training, development, and test sets. Different models are trained using the training set, the best model is selected using the development set, and then we evaluate its performance on the test set.

4.1 Requirements to measure generalisation

Using different splits is necessary, but not sufficient for NLG tasks. We can see this when we look at the generation of weather forecasts, a popular topic in the NLG community (e.g. Gkatzia et al. 2016 and references therein). It is not good enough to only have a corpus where all inputs have the same weather but different place names. NLG models trained on such a corpus would only learn to produce a fixed weather template, where they should copy in the name from the input. An evaluation is only meaningful if there are clear differences in *all* (combinations of) variables, between training and test set. At the same time, the training data should also not contain so much variation that it's impossible to detect any pattern. It is an open question how much systematicity (and redundancy) there should be in the training data for NLG systems to learn how to perform any language generation task. Finally, it is important to have specific information about the output. For example: how many different ways are there to verbalise the same predicates, entities, numbers, dates and times? Without this information, it is impossible to say anything about the complexity of the task.

4.2 Are current datasets sufficient?

We don't believe current datasets are sufficient to measure the extent to which systems are able to generalise from the training data, although some datasets do come close. WebNLG, for example, is a state-of-the-art dataset. It offers an excellent overview table (Table 1 in Gardent et al. 2017) describing properties of the input (e.g. number of different predicates, number of combinations of RDF triples, relations between the different triples) and output (e.g. number of sentences verbalising different amounts of triples).

Still missing from the description of the WebNLG corpus is the distribution of different predicates. Figure 2 shows the frequency distribution of different labels in the training set (computed using the XML files from the enriched WebNLG dataset; Castro Ferreira et al. 2018). The plot reveals that the data is heavily skewed, with 76 predicates (out of 246) occurring fewer than 10 times, while the most frequent predicate ('country') occurs 2150 times. End-to-end systems will probably perform worse on the tail of the distribution (where example outputs are scarce) than on the head (where examples are plentiful).

On the output side, it is not clear from the original WebNLG corpus how many different possible lexicalisations there are for each predicate.[8] This is difficult to study with unstructured text output, but luckily the enriched WebNLG dataset converted the outputs into templates (see Table 2 below), which we can count. Table 1 shows a selection of predicates with different ratios of unique-to-total number of templates. One can imagine that it's much easier for a model to predict the template for a predicate with a ratio of 0.12, than for a predicate with a ratio of 1.00. After all: a lower ratio means that there are more examples for each unique template. The easiest situation would be one where there is a

[7] For further discussion of shared tasks and leaderboards in NLP, see: Parra Escartín et al. 2017; Nissim et al. 2017; Rogers 2019; Ethayarajh and Jurafsky 2020.

[8] We limit ourselves to predicates here, but note that predicates are not the only part of the input that needs to be lexicalised.

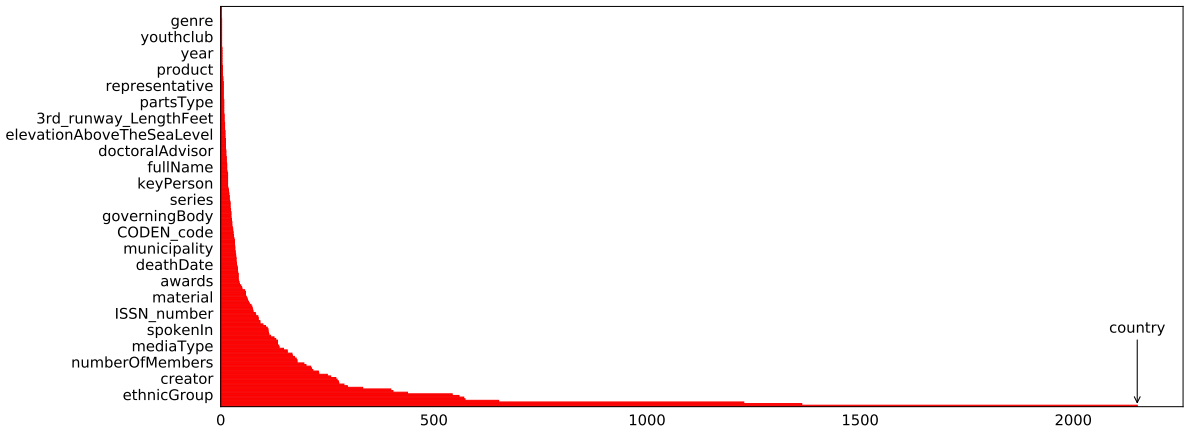

Figure 2: Frequency of all predicates in the training split of the WebNLG corpus. The x-axis shows the frequency of the labels (range: 1–2150, where 'country' is the most frequent label). To improve readability, the y-axis only shows a selection of different labels at fixed intervals.

Predicate	Unique	Total	Ratio
fullName	3	3	1.00
product	3	3	1.00
architecturalStyle	17	18	0.94
order	15	16	0.94
discipline	8	9	0.89
champions	18	21	0.86
locationCity	7	9	0.78
demonym	25	32	0.78
foundedBy	2	3	0.67
birthName	6	9	0.67
leaderTitle	37	63	0.59
address	4	7	0.57
areaCode	17	36	0.47
countySeat	7	15	0.47
language	47	119	0.39
city	14	36	0.39
status	3	11	0.27
affiliation	3	11	0.27
capacity	5	26	0.19
capital	11	61	0.18
location	22	177	0.12

Table 1: Number of templates for predicates in the '1-triples' subset of the WebNLG corpus. Columns show the predicate, number of unique templates, total number of templates, and the ratio of unique-to-total number of predicates. This table shows a selection across the entire range of different ratios.

single template, with many examples. Evaluation of NLG systems trained on this corpus should ideally take this uniqueness ratio into account (e.g. by computing performance for different subsets of the data, with different uniqueness ratios).

4.3 What do we need?

Our discussion so far points us in the direction of more carefully planned corpora, with clear input distributions. Ideally the output language should also be controlled, so that it conforms to a set of guidelines for what appropriate output should look like. To really test the degree to which system performance depends on these variables (i.e. the distributions of input and output), having just one big training corpus isn't good enough. Rather, there should be different versions of the same corpus, so that we can manipulate different aspects of the training data, to see how each of those variables affects the outcome (performance on the test set).

5 How can we get there? Rules!

Corpora constructed solely through human labor are not good enough, because they do not give us enough control over the data to carry out systematic experiments. For example: we only know the amount of different lexicalisations for a predicate *after* data collection has finished. We present two alternative approaches to generate evaluation data.

5.1 From systems to synthetic datasets

One way to construct a controlled corpus, is to use existing NLG systems to produce a large collection of texts in a particular domain. We can then train

end-to-end systems on this data to produce similar texts. This has three major advantages, compared to the use of crowd-sourced data:

1. It is efficient and cost-effective to produce large corpora, since no human annotators are needed.
2. We can easily create different sub-corpora with very specific distributions of the input data, which would allow us to estimate the extent to which systems are able to generalise from low-frequent training examples.
3. It allows us to automatically evaluate the quality of the output in ways that are not possible (or very labor-intensive) with human-generated data.

The generate-and-train approach has recently been applied by Oraby et al. (2018, using the PERSONAGE system; Mairesse and Walker 2010) to create a synthetic corpus of utterances in the RESTAURANT domain, where the authors controlled the *personality* of the utterances. Oraby et al. showed that it is possible for neural NLG models to distinguish style and content, and that models trained on their data were able to generate meaningful output with the desired personality traits.

The idea of training NLG-systems based on the output of other NLG systems is controversial. Ehud Reiter argues on his blog that this is just reverse-engineering existing systems (Reiter, 2017). This is a valid concern if the goal is to build an NLG system to be used in some application context. However, we are not concerned with any applications. Rather, we are interested in the core properties of different end-to-end architectures, and particularly the way those properties relate to learnability: to what extent a particular architecture can learn to generate natural language, based on a corpus with particular, controlled properties?

Feasibility

A natural question at this point is how to find existing systems to generate synthetic NLG corpora. One answer is simply to look for systems using SimpleNLG, since this is probably the most used realisation engine for NLG in academia. It may be possible to build a corpus generation tool that incorporates all different systems. To find these systems and assess the feasibility of our proposal, we used the *Publish or Perish* software[9] to retrieve all publications on Google Scholar that cite the original SimpleNLG paper (Gatt and Reiter, 2009).[10] We found 361 publications referring to SimpleNLG on Google Scholar, coming from a wide array of different venues. We are still in the process of analysing the results, but our impression is that only a small proportion of the reported systems is useful. Many are either unavailable, form part of a larger pipeline, or use proprietary/personal data (e.g. BT-Nurse; Hunter et al. 2012).

Triple: ⟨SAGE_Publications, founder, Sara_Miller_McCune⟩ **Text**: Sara Miller McCune founded SAGE Publications. **Template**: ENT-1 founded ENT-2. **Mapping**: ENT-1: Sara Miller McCune ENT-2: SAGE Publications

Table 2: Example from the extended WebNLG corpus.

5.2 From datasets to rules, and back again

Another approach is to construct our own template-driven corpus generator, based on existing datasets. Table 2 shows part of an entry from the extended WebNLG corpus. The triple was expressed by a participant through the text 'Sara Miller McCune founded SAGE Publications.' Castro Ferreira et al. (2018) semi-automatically converted these texts to templates. Additionally, the dataset also shows how different entities can be realised. This gives us all the ingredients to develop a rule-based system that can generate a corpus matching specific criteria (or indeed a collection of corpora that allow us to determine the ability to which end-to-end systems are able to generalise).

With the templates and entity realisation options in hand, we can choose to make full use of all possible templates and realisations for all predicates and entities, or we can select only specific templates/realisations to have a particular distribution of the data. Here are the aspects that we imagine may be interesting to manipulate:

- The number of different templates/entities in the train, validation, and test sets. (Note that templates and entities may be manipulated separately from each other.)

[9] Search carried out on the 17th of August, 2020, using the macOS GUI edition, version 7.25.2877.7516. Software available from: https://harzing.com/resources/publish-or-perish

[10] This approach excludes many systems using SimpleNLG in a different language, e.g. Brazilian Portuguese (de Oliveira and Sripada, 2014), Dutch (de Jong, 2018), German (Bollmann, 2011; Braun et al., 2019), French (Vaudry and Lapalme, 2013), Galician (Cascallar-Fuentes et al., 2018), Italian (Mazzei et al., 2016), or Spanish (Ramos-Soto et al., 2017).

- The frequency with which those different templates/entities each occur. Is there a uniform distribution, or do some templates/entities occur more than others?
- The overlap in terms of templates/entities between the train, validation, and test sets. Here we may also choose to generate multiple different test sets to accompany the same training set, to make evaluation more efficient.
- The ordering principles, and the number of different orders in which triples are realised in the output texts. (E.g. maintaining the input order, ordering triples alphabetically or based on their content.)
- The segmentation principles, and the number of sentences that are used to realise a set of triples. (E.g. three triples per sentence; only one triple per sentence; segmentation depending on the predicate, the entities, or both.)
- The amount of noise in the dataset. By default, there is no noise in the synthetic dataset, but we could add synthetic noise (i.e. knowingly introduce errors), to see how systems deal with the presence of noise in the data. This is similar to Dušek et al. (2019), who systematically *removed* noise from the E2E dataset, to gauge the impact of erroneous meaning representations.

Using data generated in this manner, we could answer questions like the following:

- How do skewness and diversity (of templates, referring expressions) influence the quality and diversity of the generated outputs?
- How many minority examples are necessary before end-to-end models consider these a valid alternative to majority examples?
- What kinds of generation rules are learnable by end-to-end systems? Which architectures are more apt to pick up on different kinds of systematic patterns in the data?

Feasibility

There are two main challenges for this approach. The first challenge concerns **multiple predicates**. It is easy to see how textual output for single-predicate inputs can automatically be generated (just fill in the empty slots in the template), but for inputs with multiple predicates the problem is more complex. The realisation of multiple predicates is not necessarily equal to the realisation of two single predicates, plus some text to link the two (e.g. the conjunction *and*). Indeed, Perez-Beltrachini et al. (2016) purposefully selected combinations of predicates that might lead to more concise solutions. E.g. combining ⟨Alan_Bean, occupation, test_pilot⟩ with ⟨Alan_Bean, nationality, USA⟩ leads to the insertion of an adjective: *Alan Bean was an **American** test pilot*.

Normally it would mean a large amount of human labor to find any systematicity in the corpus. To build a good NLG system, we need to know how to order the predicates; how to relate the predicates to each other; how to split up the information in different sentences; and how to realise sentences combining multiple predicates. However, for evaluation purposes, the exact answers to these questions aren't necessarily important.[11] What matters is that there is some output that conforms to a particular set of rules. The evaluation is just there to see if end-to-end systems are able to learn those rules. The exception here is when the ability to learn a specific kind of rule is in question. For example: can neural NLG systems learn to insert adjectives like *American* in the example above?

The second challenge concerns **the distribution of the original corpus**. As Table 1 shows, some predicates occur only three times. This limits the different kinds of corpora that we are able to produce. For example, it is not possible with just the WebNLG data alone to generate a corpus where there are more than three different lexicalisations for the FULLNAME predicate. Moreover, it is not even possible to generate more than nine different predicate-entity combinations (3 entities times 3 predicate-realisations). One way to address the distribution issue is to (semi-)automatically generate more examples by extracting triples from DBpedia (Auer et al., 2007; Lehmann et al., 2015), and verbalising them using the predicate's lexicalization templates available in the enriched WebNLG dataset and a grammar-based NLG system (e.g., Mille et al. 2019). As mentioned earlier, the data does not need to be perfect; it just needs to be consistent, so that learners are (in theory) able to infer rules from the data.

[11] One might have multiple rules corresponding to different answers to each question. It would then be possible to experiment with different amounts of examples generated using the different rules.

5.3 General feasibility

Ideally we would be able to control the data in such a way, that changes to individual variables happen *ceteris paribus*; i.e. with all other variables staying the same. But there are practical considerations we need to take into account:

- The number of times you can train a model, is limited by the size and complexity of that model. If it takes a long time to train the model, then it is not feasible to do this for tens or hundreds of different versions of the same training data.[12]
- This issue is further compounded by the fact that many models are randomly initialised. For a good estimation of system performance, the system needs to be trained multiple times on the exact same data.

There is no universal solution to this problem, but it does help to have specific hypotheses about which factors might affect system performance, and to focus on those.

5.4 Assessing model performance

Next to increased control over the training data, the approaches proposed in this section have an additional benefit: because all training data has been generated using a rule-based approach, we can use those same rules to evaluate which rules were learned by the system, and which ones weren't. This is also the approach we described in Section 2. We could even split up the evaluation, to measure which templates, entity realisations, ordering rules, and segmentation rules the system acquired.

One aspect we did not address yet is how to parse imperfect outputs. There are no guarantees that the output of end-to-end systems will conform to any of the rules through which the corpus was generated. Using a strict approach, we could say that faulty output just doesn't count; if it is flawed, the system simply did not fully learn the relevant rules. But perhaps we would also like to give partial credit to systems that *almost* learned how to perform the generation task. We leave this as a question for future research.[13]

[12] So next to environmental issues caused by computationally heavy approaches to NLP (Strubell et al., 2019), we can also say that such approaches are an obstacle to properly evaluate new systems.

[13] But note that the texts (and probably system outputs as well) are very predictable. This makes it interesting to explore whether metrics based on edit distance could work here, even though they have been shown to be inadequate 'in the wild.'

5.5 Predictions

Since we intend to explore this approach in the future, and to encourage others to explore this space as well, we make a number of predictions:

1. Templates with a lower unique-to-total ratio (see Table 1) are easier to learn.
2. The number of examples needed to successfully learn a template, depends on the amount of alternative templates that could also verbalise the same predicate, the amount of predicates in the corpus, and the size of the corpus.
3. It is easier to learn how to realise a predicate, if the arguments for that predicate are diverse. (If a predicate always occurs with the same arguments, they may be considered part of the template by the model.)
4. When combining multiple triples, conjunction (*Susan is an astronaut **and** she is American*) is easier to learn than insertion (*Susan is an **American** astronaut*).

This list is not exhaustive; certainly many more predictions could be made about different combinations of the parameters we described above. But these hypotheses should serve as a starting point for future research. Initially we may want to see empirically whether the predictions hold up for popular architectures. A different avenue of research could be based on this evidence, to develop formal proofs about the properties of families of NLG architectures. We believe both are needed to inform NLG research and practice.

6 Limitations

The output of rule-based systems is often said to be less fluent or natural than the output of end-to-end systems, and this claim is corroborated by the results of the E2E-challenge (Dušek et al., 2020). It may thus be expected that any synthetically generated corpus will be less natural than human-produced data, and the texts will probably have other shortcomings, too. However, the proposal in this paper is focused on determining what systems can or cannot learn from corpora with different properties. This means that, to some extent, the naturalness or fluency of the synthetic data does not really matter. What matters is that we learn how those different properties of the data affect the output of data-driven systems. We can then use those systems in other areas, knowing what they are capable of and what their limitations are. At

that point, we need a different kind of evaluation (although rules are still valuable to check whether system output conforms to particular guidelines).

One might reasonably object here that the quality of the corpus *does* matter. How can you be sure that your synthetic data has the desired properties? Wouldn't that require some form of evaluation as well? We believe this concern could be addressed through unit-tests in the corpus generation code base. Because our proposal involves rule-based generation, the output should always be predictable.

7 Conclusion

We discussed the merits of (grammar) rules and rule-based systems in the context of NLG evaluation. Our conclusion is that there are clear benefits for practitioners who want to learn more about the architectures that they use for real-life applications. A concern that some may have, is that the real world is messy. Why should we solve toy problems like reverse-engineering rule-based systems? Our answer is two-fold. First, since our proposals involve synthetic data, we can make the data as clean or messy as we want. But because we have full control over the data, the evaluation will be much more informative about what systems can or cannot do. Second, a rule-based perspective is useful because it forces us to engage with the data. Looking at the WebNLG data, and all of the different templates that exist for each of the different predicates, one cannot help but ask: is this diversity really useful? Or should we try to reduce the diversity (e.g. formulating guidelines), to ensure the best possible outputs for our NLG systems? Messiness can be good or bad, and it is up to us to explore the impact of variation in NLG data.

Acknowledgments

We thank the anonymous reviewers for their feedback, which helped us refine the arguments laid out in this paper.

References

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Srinivas Bangalore, Owen Rambow, and Steve Whittaker. 2000. Evaluation metrics for generation. In *INLG'2000 Proceedings of the First International Conference on Natural Language Generation*, pages 1–8, Mitzpe Ramon, Israel. Association for Computational Linguistics.

Marcel Bollmann. 2011. Adapting SimpleNLG to German. In *Proceedings of the 13th European Workshop on Natural Language Generation*, pages 133–138, Nancy, France. Association for Computational Linguistics.

Daniel Braun, Kira Klimt, Daniela Schneider, and Florian Matthes. 2019. SimpleNLG-DE: Adapting SimpleNLG 4 to German. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 415–420, Tokyo, Japan. Association for Computational Linguistics.

Andrea Cascallar-Fuentes, Alejandro Ramos-Soto, and Alberto Bugarín Diz. 2018. Adapting SimpleNLG to Galician language. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 67–72, Tilburg University, The Netherlands. Association for Computational Linguistics.

Thiago Castro Ferreira, Diego Moussallem, Emiel Krahmer, and Sander Wubben. 2018. Enriching the WebNLG corpus. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 171–176, Tilburg University, The Netherlands. Association for Computational Linguistics.

Noam Chomsky. 1965. *Aspects of the Theory of Syntax*. MIT Press.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 376–380, Baltimore, Maryland, USA. Association for Computational Linguistics.

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. 2020. Evaluating the state-of-the-art of end-to-end natural language generation: The e2e nlg challenge. *Computer Speech & Language*, 59:123 – 156.

Desmond Elliott and Frank Keller. 2014. Comparing automatic evaluation measures for image description. In *Proceedings of the 52nd Annual Meeting of*

the Association for Computational Linguistics (Volume 2: Short Papers), pages 452–457, Baltimore, Maryland. Association for Computational Linguistics.

Kawin Ethayarajh and Dan Jurafsky. 2020. Utility is in the eye of the user: A critique of nlp leaderboards.

Dan Flickinger. 2000. On building a more efficient grammar by exploiting types. *Nat. Lang. Eng.*, 6(1):15–28.

Dan Flickinger. 2011. Accuracy vs. robustness in grammar engineering. In E. M. Bender and J. E. Arnold, editors, *Language from a cognitive perspective: Grammar, usage, and processing*, pages 31–50. Stanford: CSLI Publications.

Markus Freitag, Isaac Caswell, and Scott Roy. 2019. APE at scale and its implications on MT evaluation biases. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 34–44, Florence, Italy. Association for Computational Linguistics.

Markus Freitag, David Grangier, and Isaac Caswell. 2020. Bleu might be guilty but references are not innocent.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. The WebNLG challenge: Generating text from RDF data. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 124–133, Santiago de Compostela, Spain. Association for Computational Linguistics.

Albert Gatt and Ehud Reiter. 2009. SimpleNLG: A realisation engine for practical applications. In *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009)*, pages 90–93, Athens, Greece. Association for Computational Linguistics.

Dimitra Gkatzia, Oliver Lemon, and Verena Rieser. 2016. Natural language generation enhances human decision-making with uncertain information. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 264–268, Berlin, Germany. Association for Computational Linguistics.

James Hunter, Yvonne Freer, Albert Gatt, Ehud Reiter, Somayajulu Sripada, and Cindy Sykes. 2012. Automatic generation of natural language nursing shift summaries in neonatal intensive care: Bt-nurse. *Artificial Intelligence in Medicine*, 56(3):157 – 172.

R.F. de Jong. 2018. Simplenlg-nl : Natural language generation for dutch.

Mert Kilickaya, Aykut Erdem, Nazli Ikizler-Cinbis, and Erkut Erdem. 2017. Re-evaluating automatic metrics for image captioning. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 199–209, Valencia, Spain. Association for Computational Linguistics.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision – ECCV 2014*, pages 740–755, Cham. Springer International Publishing.

François Mairesse and Marilyn A. Walker. 2010. Towards personality-based user adaptation: psychologically informed stylistic language generation. *User Modeling and User-Adapted Interaction*, 20(3):227–278.

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

Alessandro Mazzei, Cristina Battaglino, and Cristina Bosco. 2016. SimpleNLG-IT: adapting SimpleNLG to Italian. In *Proceedings of the 9th International Natural Language Generation conference*, pages 184–192, Edinburgh, UK. Association for Computational Linguistics.

Simon Mille, Stamatia Dasiopoulou, and Leo Wanner. 2019. A portable grammar-based nlg system for verbalization of structured data. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, SAC '19, page 1054–1056, New York, NY, USA. Association for Computing Machinery.

Emiel van Miltenburg. 2016. Stereotyping and bias in the flickr30k dataset. In *Proceedings of Multimodal Corpora: Computer vision and language processing (MMC 2016)*, pages 1–4.

Emiel van Miltenburg. 2020. How do image description systems describe people? a targeted assessment of system competence in the people domain. In *Proceedings of LANTERN*. Association for Computational Linguistics.

Emiel van Miltenburg, Desmond Elliott, and Piek Vossen. 2017. Cross-linguistic differences and similarities in image descriptions. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 21–30, Santiago de Compostela, Spain. Association for Computational Linguistics.

Emiel van Miltenburg, Desmond Elliott, and Piek Vossen. 2018. Talking about other people: an endless range of possibilities. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 415–420, Tilburg University, The Netherlands. Association for Computational Linguistics.

Tom M. Mitchell. 1997. *Machine learning*. McGraw-Hill.

Malvina Nissim, Lasha Abzianidze, Kilian Evang, Rob van der Goot, Hessel Haagsma, Barbara Plank, and Martijn Wieling. 2017. Last words: Sharing is caring: The future of shared tasks. *Computational Linguistics*, 43(4):897–904.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017. The E2E dataset: New challenges for end-to-end generation. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 201–206, Saarbrücken, Germany. Association for Computational Linguistics.

Jekaterina Novikova, Oliver Lemon, and Verena Rieser. 2016. Crowd-sourcing NLG data: Pictures elicit better data. In *Proceedings of the 9th International Natural Language Generation conference*, pages 265–273, Edinburgh, UK. Association for Computational Linguistics.

Rodrigo de Oliveira and Somayajulu Sripada. 2014. Adapting SimpleNLG for Brazilian Portuguese realisation. In *Proceedings of the 8th International Natural Language Generation Conference (INLG)*, pages 93–94, Philadelphia, Pennsylvania, U.S.A. Association for Computational Linguistics.

Shereen Oraby, Lena Reed, Shubhangi Tandon, Sharath T.S., Stephanie Lukin, and Marilyn Walker. 2018. Controlling personality-based stylistic variation with neural natural language generators. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 180–190, Melbourne, Australia. Association for Computational Linguistics.

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

Carla Parra Escartín, Wessel Reijers, Teresa Lynn, Joss Moorkens, Andy Way, and Chao-Hong Liu. 2017. Ethical considerations in NLP shared tasks. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, pages 66–73, Valencia, Spain. Association for Computational Linguistics.

Laura Perez-Beltrachini, Rania Sayed, and Claire Gardent. 2016. Building RDF content for data-to-text generation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1493–1502, Osaka, Japan. The COLING 2016 Organizing Committee.

Carl. Pollard and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. Head-driven Phrase Structure Grammar. University of Chicago Press.

Alejandro Ramos-Soto, Julio Janeiro-Gallardo, and Alberto Bugarín Diz. 2017. Adapting SimpleNLG to Spanish. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 144–148, Santiago de Compostela, Spain. Association for Computational Linguistics.

Ehud Reiter. 2016. Natural language generation and machine learning. Ehud Reiter's blog, https://ehudreiter.com/2016/12/12/nlg-and-ml/.

Ehud Reiter. 2017. You need to understand your corpora! The Weathergov example. Ehud Reiter's blog, https://ehudreiter.com/2017/05/09/weathergov/.

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

Ehud Reiter. 2020. Is building neural nlg faster than rules nlg? no one knows, but i suspect not. Ehud Reiter's blog, https://ehudreiter.com/2020/05/11/is-building-neural-nlg-faster/.

Anna Rogers. 2019. How the transformers broke nlp leaderboards. Posted on the *Hacking Semantics* blog: https://hackingsemantics.xyz/2019/leaderboards/.

David Schlangen. 2019. Language tasks and language games: On methodology in current natural language processing research. *CoRR*, abs/1908.10747.

David Schlangen. 2020. Targeting the benchmark: On methodology in current natural language processing research. *CoRR*, abs/2007.04792.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.

Elior Sulem, Omri Abend, and Ari Rappoport. 2018. BLEU is not suitable for the evaluation of text simplification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 738–744, Brussels, Belgium. Association for Computational Linguistics.

Pierre-Luc Vaudry and Guy Lapalme. 2013. Adapting SimpleNLG for bilingual English-French realisation. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 183–187, Sofia, Bulgaria. Association for Computational Linguistics.

Huiyuan Xie, Tom Sherborne, Alexander Kuhnle, and Ann Copestake. 2019. Going beneath the surface: Evaluating image captioning for grammaticality, truthfulness and diversity. In *Proceedings of Rigorous Evaluation of AI Systems 2019, collocated with The seventh AAAI Conference on Human Computation and Crowdsourcing*.