An Engineering Perspective on Data-to-text NLG

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

September 2, 2022

Abstract

I present a software engineering perspective on data-to-text, which is a form of Natural Language Generation (NLG). I discuss data-to-text from a perspective of requirements analysis, design and testing/evaluation as well as implementation. All of these are essential to building high-quality solutions, in data-to-text as well as in other types of software

1 Introduction

Natural Language Generation (NLG) systems generate texts in English and other human languages using natural language processing (NLP) and computational linguistics techniques. *Data-to-text* NLG systems generate texts from non-linguistic input data, such as spreadsheets and databases. For example a data-to-text system could produce a written weather forecast from numerical weather data Goldberg et al. (1994), or a written sports story from numerical data about a sports game [1].

In contrast, *Text-to-text* NLG systems generate texts from other texts. For example, a text-to-text system could generate a written summary of a medical consultation Knoll et al. (2022), or suggest alternative wording (paraphrase) to a human author [2]. *Prompt-to-text* systems generate texts from a short prompt. For example, a prompt-to-text system could generate a story from a prompt giving the first few sentences of the story [3].

[1] https://www.bbc.co.uk/news/technology-34204052

[2] https://www.wordtune.com/

[3] https://www.theguardian.com/commentisfree/2020/sep/08/robot-wrote-this-article-gpt-3

Most research papers on data-to-text focus on implementation and technology, but creating successful data-to-text solutions also requires a good understanding of requirements, design, and testing/evaluation. This paper presents a holistic 'software engineering' perspective on data-to-text that includes all of these aspects (including implementation).

1.1 Example use case: Business intelligence

The most successful commercial use case of data-to-text NLG (at the time of writing) is probably *business intelligence* (BI). BI systems help people understand and draw insights from complex data sets. Most BI systems use purely graphical presentations of data, but there is a growing realisation in the BI world that NLG textual summaries can supplement and enhance data visualistions.

For example, during the Covid pandemic, Arria and Tibco developed a dashboard which include both visual and text presentations of Covid data[4]. An example textual summary is

> As of Sunday 17th May 2020, there are at least 4,364,603 confirmed cases of COVID-19 worldwide. So far, 283,911 deaths have been recorded as a result of the virus. The number of cases reported in the United States is increasing to a rate of 0.01%, while Russia is starting to slow down, having a rate of -0.08%, compared to its average rate of -0.07% over the previous week.

This summary highlights key insights from the data, after reading this users can get more information from the data visualisations.

2 Requirements Analysis

The first, and probably most important, challenge in building a real-world NLG system is ensuring that it meets real use needs and expectations. There's not much point in building a fancy NLG (or AI) system that does something which no one in the real world is interested in! As in other areas of software, getting requirements wrong in a major cause of failure; getting requirements right is usually more important than getting the technology right.

Certainly in my own experiences I have seen many NLG projects fail because the developers did not properly understand what users and stakeholders wanted, needed, and expected. Some researchers have a tendancy to dream up plausible-sounding

[4] https://ehudreiter.com/2020/05/21/adding-narrative-to-a-covid-dashboard/

use cases without actually checking whether real users want this functionality; this kind of behaviour may lead to papers in academic venues, but it is unlikely to lead to useful real-world NLG systems.

There is of course an extensive literature on software requirements analysis. In this paper I will focus on aspects that are unique to NLG.

2.1 Text, Graphics, and Interaction

Data can be explained and summarised using data visualisations as well as NLG text summaries. So a key issue is when textual summaries 'add value' to visualisations. A related issue is whether and how users interact with an NLG system.

There certainly are specific contexts where NLG text summaries have been shown to be more useful than a competing data visualisation (van der Meulen et al., 2010), although its more common for NLG summaries to be used along with visualisation rather than completely replacing them (Gkatzia et al., 2016; Balloccu and Reiter, 2022). Commercial NLG Business Intelligence tools (subsection 1.1) usually supplement visualisations, they dont replace them. Law et al. (2005) point out that users may like and prefer visualisations even if they are not very effective as decision support tools.

The most obvious way to interact with a data-to-text NLG system is via a chatbot or voice assistant, this allows users to ask specific questions, including followup questions. However it is also possible for users to interact with an NLG system in other ways, for example by clicking in words in a generated text or by using a GUI to change the input set or adjust NLG parameters. If the NLG system is linked to an interactive data visualisation system, then interaction ideally will be similar and consistent for NLG and visualisation.

Unfortunately, our theoretical understanding of above issues (combining text and graphics, interaction) is very limited. From a practical perspective, standard user-study techniques, such as prototyping and wizard-of-oz, can be used to understand how users want to interact with a system, and indeed what functionality best supports users in their workflows. For example, Knoll et al. (2022) describe how such techniques were used to elicit key requirements for a system which summarised medical consultations, including the need to generate such summaries in real-time as the consultation progressed.

2.2 Language and content

Of course we need to ensure that generated texts communicate the data insights that users need to know, in appropriate language. Appropriate language and insights are largely based on utility (what will help the user), but other factors come into play as well. For example, in addition to being clear and easily readable, the language used in generated texts may need to follow corporate writing guidelines and be influenced by regulatory concerns. Similarly texts may need to communicate insights motivated by legal concerns (eg, fear of malpractice lawsuits) even if they dont actually help the user much.

A related issue is which *quality criteria* (Belz et al., 2020) are most important to users; this is different contexts. For example, correctness (accuracy) is of critical importance in NLG systems which generate medical texts which are directly shown to clinician (Portet et al., 2009), but is less critical in contexts where a human clinician first checks and post-edits the text (Knoll et al., 2022). Being interesting and engaging is not usually very important for clinical decision support, but it is very important in sports media, where generated stories need to attract eyeballs.

Of course it is essential to work with users and subject matter experts to understand appropriate language and content. *Corpus analysis* can also be very useful, this involves analysing human-written texts in the domain to understand the content and language they use. If only a small number of texts is available, this is usually done manually by a developer, who discusses his or her observations with users and domain experts. If a larger number of texts are available, we can use machine learning techniques to build models of how insights and words are chosen (Reiter et al., 2005); if these models are interpretable (eg, decision trees), then we can discuss them with users and domain experts and indeed combine what we learn from corpus analysis with what stakeholders directly tell us. Reiter et al. (2003) discusses the pros and cons of corpus analysis vs working with experts, and how to combine these.

One challenge with both of the above techniques (corpus analysis and talking to stakeholders) is that they may not give good coverage of unusual edge cases. Users often do not mention these without prompting, and an unusual edge case may not appear in a corpus. So developers may need to explicitly work with stakeholders to understand how to handle edge cases.

Another challenge is that different experts and corpus writers may have different views on appropriate language and content. From a practical perspective, it may be easiest to focus on a single expert (or analyse corpus texts from a single writer) and build a system which imitates this specific individual instead of a collection of experts and writers.

2.3 Configuration and control

NLG solutions often need to be configured for different users. For example, a weather-forecast generator will need to be configured for different meteorological organisations (eg UK Met Office vs Environment Canada), and a sports story generator will need to be configured for different sports and leagues. These configurations are not static, for example a weather organisation may decide to issue a new type of weather forecast, and a sports media company may decide to use a different provider for sports data.

A related issue is configuring the language used by an NLG solution for different clients. For example, in investment reporting one company may want reports to talk about "equities" while another wants reports to talk about "stocks". This may sounds trivial, but if customers and users really care about this sort of linguistic choice, then the NLG system needs to allow it to be configured and controlled.

The academic community has mostly ignored configuration issues, which is a shame because this is a major requirement for many commercial NLG systems and products. From a requirements perspective, the goal is to understand the types of configuration that need to be supported. Of course its useful to discuss this with users and stakeholders. Its also often useful to look at what changes have happened historically. For example, if a sports media company has changed its provider of sports data 5 times in the past 10 years, then it is reasonable to expect such changes to happen in the future as well.

2.4 Safety and Ethics

As in other areas of AI and NLP, it is essential that NLG systems be safe and ethical!

Probably the main challenge is to ensure that texts are accurate and do not mislead people into doing dangerous things. For example, a medical NLG system should not make inappropriate medical suggestions which could kill a patient (Bickmore et al., 2018)! We also do not want NLG systems to generate texts which show racial or other biases, or use potentially offensive language such as profanity. There has been a fair amount of research on safety issues in the dialogue community (Dinan et al., 2022), much of which can be applied to NLG.

van Deemter and Reiter (2018) discuss whether NLG systems should lie to users. While at first thought the answer seems to be "obviously not", van Deemter and Reiter point out that some "deviation from the truth" is inevitable in complex data-to-text systems. They also point out some contexts where explicit lying may be ethical, for example not telling someone who is at risk of heart attacks that his/her grandchild is very sick and may die.

3 Design and Architecture

Architectural design is crucial for data-to-text NLG, as it is for other types of software systems. From an architectural perspective, an NLG system can be "end-to-end", generating output texts from inputs in a single step, or it can be broken up into modules which handle different tasks and types of processing. The "end-to-end" vs modularised distinction is not technology-dependent; for example, we can build end-to-end systems with templates as well as machine-learning.

At least in my experience, end-to-end approaches do not work well for complex data-to-text systems which produce lengthy outputs, especially in production contexts where systems need to be thoroughly tested and maintained over time. Its easier to test and maintain systems if they are modularised, and its also easier to reuse components of a modularised system. This of course is true of all software systems, not just NLG.

3.1 Analytics and Narrative Expression

Looking specifically at data-to-text, when we split a system up into modules, the most important split is between *analytics* (proposing insights to communicate to the user) and *narrative expression* (generating a narrative which communicates these insights). The types of processing needed in analytics is very different from what is needed in narrative expression, as are the skills needed by developers working on these modules.

The interface between analytics and narrative expressions is *messages* which encode the insights produced by the analytics part of the NLG system. These are often represented as classes in an object-oriented programming language such as Java or Python. Messages can link to each other, for example one message (insight) may have a causal relationship with another message.

In a sports NLG system, for example, messages would communicate insights ranging from obvious (who won the game) to more subtle (eg, a notably performance from a player who usually does not shine). Note that different insights are interesting and important to different audiences. For example, in a World Cup match between France and Germany, French fans will be more interested in insights about players on the French team. Understanding which insights are important in different contexts is part of requirements analysis.

3.2 Analytics Architecture

Analytics components of NLG systems are diverse and domain dependent; the kind of analytics needed to generate a weather forecast text from weather data is very different from what is needed in business intelligence, which in turn is very different from what is needed in clinical decision support.

Nevertheless, in many contexts it is useful to make a distinction between a *signal analysis* component which looks for patterns in the data (and filters out noise), and a *data interpretation* component which derives domain-specific insights from this data. For example, in a clinical context, a signal analysis module might detect that a patient's temperature is steadily rising over time, while a data interpretation module could generate from this pattern the insight that the patient may have an infection.

Signal analysis modules tend to use generic pattern detection algorithms (perhaps fine-tuned to a domain), with the biggest challenge often being dealing with noise (temperature in real patients does not steadily rise, it jump around depending on what the patient is doing at the moment). Data interpretation modules use domain-specific reasoning (such as the fact that infections can increase a patient's body temperature) to infer insights from the patterns; ideally this logic should be based on a proper evidence base, but sometimes developers need to rely on intuition of domain experts.

3.3 Narrative architecture

NLG systems that generate texts which are longer than a sentence usually make a distinction between document-level processing (deciding which messages/insights to express and how to structure them in a document) and sentence-level processing (creating sentences which express messages).

Document-level processing is usually known as *document planning*. Most NLG users want generated texts to be stories or narratives about events (such as sports matches, or changes in sales in a business intelligence context), and it is the job of the document planner to select appropriate messages and link them together into something which reads like a narrative. Of course the narratives should conform to expectations, eg a sports story should start be saying who won the match.

Sentence-level processing is often decomposed into *microplanning* (deciding which words and syntactic structures to use in a sentence) and *surface realisation* (creating a readable and grammatically correct sentence). However this demcomposition is not universal, and neural NLG systems in particular often combine these steps.

3.4 Architectures for Neural NLG

Most neural NLG systems separate analytics from narrative processing. Even so-called 'end-to-end' neural systems generally start from semantic 'meaning representations' which are similar to what I have called messages, ie that they include the results of analytics. At the narrative level, neural NLG systems which generate texts that are longer than a few sentences usually include some kind of document planning facility (Puduppully and Lapata, 2021). However 'end-to-end' neural systems usually do combine microplanning and surface realistion; many template and rule-based NLG systems also do this.

A general problem with neural approaches, especially end-to-end ones, is that they can generate texts that are factually incorrect (*hallucination*) or that leave out key information (*omission*). This is a major problem if requirements analysis shows that correctness and other content-level factors are very important in the target use case.

At the time of writing, there is interest in architectures which combine neural and rule-based techniques, especially amongst people who are trying to build real-world production NLG systems; this is partially seen as a way of reducing hallucination and omission problems. For example, Arun et al. (2020) describe an architecture which includes a neural NLG system, a backup rule-based NLG system, and a module which checks the output of the neural system for potential problems; if the checking module finds problems, then the system switches to the backup rule-based NLG system. Kale and Rastogi (2020) descibe an architecture where a simple rule-based system generates an initial draft narrative (which includes correct content but may not be expressed well), and a neural NLG system rewrites the draft to improve its coherence and readability.

High quality training data is a big help in building production-quality neural NLG systems Arun et al. (2020). Similar to code, data sets should be explicitly designed (contents, structure, acquistion technique) so that they are appropriate for an application.

4 Implementation

There are many ways to implement data-to-text systems. In this section I will describe some implementation options at the time of writing; I will generally follow the modularisation described in the architecture section. Gatt and Krahmer (2018) present a good survey of many NLG implementation techniques as of 2018.

Technology of course is rapidly evolving, so I focus below on general principles instead of on specific libraries or tools.

4.1 Analytics

As mentioned anove, analytics components of data-to-text systems are very diverse and domain-dependent. One generic issue is that data-to-text systems need messages that fit naturally into narratives, which can require a different type of analytics than that needed for predictive modelling (for example). This is sometimes called *articulate analytics*. For example, Sripada et al. (2003) point out that in the context of analysing trends for data-to-text systems, linear interpolation may be better than linear regression because interpolation is more easily expressed in a narrative.

4.2 Document Planning

Document planning involves choosing the messages to be communicated in a narrative and structuring and ordering these messages at a document level. The primary goal is to create a narrative or story about the data, which is what users usually say they want.

The output of document planning is generally a tree structure, whose internal nodes represent document-level structures such as paragraphs, and whose leaf nodes are messages. Sometimes document plans include ’rhetorical relations’ between messages, for example one message being a cause or consequence of another message.

There are many challenges and subtleties in producing good narratives (Reiter et al., 2008), but they are not well understood. Of course there is an extensive literature on narratology and also the psychology of reading, but it is not easy to apply this work to data-to-text document planning (Thomson et al., 2018). More research on this topic is badly needed, including machine learning approaches.

In most real-world NLG systems, document planning is done using scripts or schemas which are coded by developers and often designed to mimic document structures seen in corpus texts.

4.3 Microplanning

Microplanning involves making linguistic choices about which words and syntatic structures should be used to express messages and document structure. There is an extensive research literature on specific microplanning tasks, including

- *Lexicalisation*: Choosing words to express data; for example ’the stock market *soared*’ vs ’the stock market increased’ (Smiley et al., 2016). (Reiter et al., 2005) show that good word choice can have a major impact on text quality. Recently there has been a lot of interest in using machine learning techniques for word choice (Zhang et al., 2018; Chen and Yao, 2019).

- *Reference*: There was been extensive research on choosing referring expressions, for example deciding whether to refer of this paper as *Professor Reiter*, *the author*, or *him*. An excellent survey is presented in Section 2.5 of Gatt and Krahmer (2018). Same et al. (2022) point out that different approaches work best in different contexts.

- *Aggregation*: Aggregation involves combining and eliding sentences, for example producing *John went to the store. John bought an apple.* into *John went to the store and bought an apple*. Various models have been proposed for aggregation, including (Harbusch and Kempen, 2009) and White and Howcroft (2015). One complication is that aggregation adds semantic and pragmatic connotations. For example, *John went to the store and bought an apple* implies the apple was bought in the store; if the system does not know this, then it should not use this aggregation.

Microplanning (sometimes combined with surface realisation) can also be done as a single step, without decomposing it into a set of tasks; many neural NLG systems take this approach. For example, the WebNLG challenge (Gardent et al., 2017; Castro Ferreira et al., 2020) solicits systems which map a semantic web meaning representation into sentences; this is usually done by creating and training neural language models. This approach is also used by Arun et al. (2020) in their commercial weather-forecast generator.

4.4 Surface Realisation

Surface realisers generate grammatically correct sentences which express messages using guidance (eg, lexical/word choice) from the microplanner. There are a number of libraries and toolkits available for surface realisation. These include

- *Template systems*: The simplest template systems just fill in slots in template sentences. More complex systems like Jinja[5] allow scripting logic within templates, and NLG template systems like RosaeNLG [6] also allow some syntactic processing and markups, eg for word inflections.

- *Grammatical realisers:* Grammatical realisers such as SimpleNLG[7] generate sentences from grammatical representations. The distinction between a template system and a grammatical system is fuzzy (van Deemter et al., 2005), but the input to grammatical realisers usually includes explicit grammatical structures such as clauses, while the input to template systems generally does not.

From a more theoretical perspective, there has been considerable interest in training realisers using machine learning techniques (Mille et al., 2020); because languages have evolved instead of being explicitly designed, grammars contain a huge number

[5] `https://palletsprojects.com/p/jinja/`

[6] `https://rosaenlg.org/`

[7] `https://github.com/simplenlg/simplenlg`

of special cases and exceptions which are probably best learned from corpora. However, at the time of writing, work on neural approaches seems to increasingly focus on a combined microplanning and realisation task, not on just realisation.

4.5 Data

If we use neural or machine learning techniques to build data-to-text systems, then training data is needed. Arun et al. (2020), (Dušek et al., 2019) and others point out that high-quality training data is critical in reducing the number of mistakes made by neural NLG systems, including hallucination (generating factually incorrect texts) and omissions (generating texts that omit key messages and insights).

In many NLP applications, quantity of training data seems more important than quality, which is why people tend to use very large datasets which include low-quality texts, such as internet crawls or reddit dumps. However, this strategy does not seem to work well for data-to-text, where there is a real danger that systems will learn inappropriate behaviour from low-quality data, which in turn will lead to occasional generation of unacceptable texts.

5 Evaluation and Testing

If we developing new NLG techniques (text-to-text as well as data-to-text), we need to be able to evaluate how effective these techniques are in producing good quality texts; if we are developing new NLG products, we need to be able to test our product to ensure that its texts are of acceptable quality. Evaluating and testing NLG systems can be challenging, not least because we need to define what constitutes a 'good' text. Gehrmann et al. (2022) provide a good survey of many NLG evaluation techiques,

5.1 Quality criteria

A key task in requirements analysis (subsection 2.2) is understanding the important quality criteria in an application and use case. (Howcroft et al., 2020) surveys criteria used by the research community, and Belz et al. (2020) analyse these criteria in terms of the underlying factors of type of quality (correctness, goodness, features), aspect of text being evaluated (form, content, both), and frame of reference (just text, text and input data, text, external frame). From this perspective, some key quality criteria are

- *grammaticality* (correctness of form): is a text grammatically and otherwise correct from a linguistic perspective?

- *readability* (goodness of form): is a text easy to read and understand?
- *accuracy* (correctness of content): is the information in a text correct?
- *helpfulness* (goodness of content): is the information in a text useful and helpful?
- effectiveness (goodness from external frame of reference, also sometimes called *extrinsic* evaluation): does the text achieve its communicative goal, such as helping users make good decisions?

When evaluating an NLG *system* (as opposed to an individual output text), we need to combine ratings of individual output texts into a system rating. The most straightforward technique, which is popular in academic contexts, is for the system rating to be based on the average ratings of its output texts. However in many commercial contexts, and in all contexts where safety is an issue, it may make more sense for the system rating to be based on the minimum ratings of any of its output texts; software testing (subsection 5.4) also focuses on 'worst case' behaviour.

Of course non-functional criteria such as time taken to generate texts can also be important.

Usually the most important criteria for specific NLG applications is effectiveness, eg do weather forecasts produced by NLG help users make good weather-related decisions, and do sports stories produced by NLG attract readers (who might then look at advertising). However effectiveness is difficult to measure, not least because its contextual (eg for weather forecasts, may depend on reader and weather situation). Also effectiveness measures are specific to a use case, so may not be ideal for evaluating a technology (instead of a system).

In general, different criteria are important in different contexts. For example, in a medical clinical decision support system, worst-case accuracy may be of paramount importance; developers must be able to guarantee that the output texts are always factually correct! In addition to the impact on patient care, incorrect texts could lead to malpractice lawsuits. On the other hand, sports stories need to be easy to read and engaging/interesting; some ungrammatical texts may be acceptable, and while factual accuracy is important, readers do seem willing to accept texts that occasionally contain incorrect information [8].

5.2 Human evaluation

The best NLG evaluations usually involve people. In rough terms, we can ask people to do the following

[8] https://ehudreiter.com/2022/04/03/humans-make-mistakes-too/

- *Rate or rank texts*: for example, give texts a Likert rating for 'this text ie easy to read'. van der Lee et al. (2021) discuss this type of evaluation in detail, and give recommendations on best practice.

- *Find and annotate problems in a text*: for example, identify all factual errors in a text. This type of evaluation is newer, but often seems to give more reliable results. Examples include Freitag et al. (2021) and Thomson and Reiter (2020).

- *Extrinsic effectiveness*: measuring impact of an NLG on an actual real-world outcome, such as better decision making (Portet et al., 2009) and behaviour change (Braun et al., 2018). A/B testing techniques can be used for this.

Key issues in all types of human evaluation are choice of appropriate subjects, solid experimental design, ethics, and replicability. *Subject choice* is very important and is discussed in most of the above-cited papers; we need subjects who take the evaluation task seriously (which can be an issue with crowdworkers) and usually we want subjects who are representative of the user community. In some cases we want subjects with domain expertise.

There are many aspects to *experimental design*. A few key ones for NLG include appropriate baselines (ideally systems should be compared to state-of-the-art, not artificially weak baselines), good coverage (scenarios in experiments should be varied and cover the space of possibilities, they should not be near-duplicates), and proper statistical analysis and hypothesis testing. Sometimes *ecological validity* (doing an experiment in a realistic setting, eg testing a clinical NLG system in a hospital ward (Hunter et al., 2012)) is also very important.

Whenever working with human subjects, researchers need to ensure ethical concerns are addressed. This is especially important when evaluating real-world effectiveness. For example, if we want to assess the effectiveness of an NLG system in encouraging safer driving (Braun et al., 2018), we need to ensure that the system will not impair driving, by distracting the user or giving inappropriate advice.

Finally, human evaluations should be *reproducible*; ie other researchers should be able to repeat the experiment and get similar results (Belz et al., 2021). A prerequisite for reproducibility is for researchers to publish a detailed description of their experiments; the HEDS datasheet (Shimorina and Belz, 2022) may be useful from this perspective,

5.3 Automatic evaluation

Automatic evaluation techniques use algorithms, often called *metrics*, to assess the quality of a generated text. In many but not all cases, the algorithms compare the generated texts to one or more ’reference texts’ written by humans.

A large number of metrics have been proposed; Gehrmann et al. (2022) and Celikyilmaz et al. (2020) describe the more popular ones in NLG, and Kocmi et al. (2021) give a detailed analysis of the validity of popular metrics for machine translation.

In principle, metrics should only be used if they have a good correlation with high quality human evaluations. Reiter (2018) presents a meta-survey of studies on how well the popular BLEU metric correlates with human evaluations, and concludes the correlation is reasonable for machine translation but not for data-to-text. Mathur et al. (2020) point out some problems with correlation studies which mean that small difference in metric scores such as BLEU may not be meaningful.

As with human evaluations, metric evaluations should be reproducible by other researchers. Surveys (Mieskes et al., 2019; Belz et al., 2021) suggest that even seemingly small differences in metric code or in ancilliary operations such as preprocessing can have a big difference in outcome. Hence it is essential for researchers to record their experiments in great detail in order to support replicability, and for metric developers to provide standardised versions of metrics which include preprocessing (Post, 2018).

5.4 Software testing and quality assurance

When building real-world NLG systems, we need to test them to ensure that they are fit for purpose. Software testing is related to evaluation, but tends to focus on edge cases and worst-case behaviour. In other words, while most academic evaluation examines how well a system performs on average, software testing identifies specific contexts and data sets where a system does not do what it is supposed to do.

Most software testing techniques rely on comparing a system’s behaviour or outputs to expected behaviour/outputs on *test cases* which cover unusual and edge (borderline) cases as well as straightforward cases. Unfortunately, it is difficult to apply such techniques to NLG systems which are non-deterministic, ie randomly vary their outputs; this includes rule-based systems with a random choice component (eg, randomly choosing between synonyms in order to vary the language in a text) as well as neural NLG systems.

Testing NLG systems can be a real headache and ’pain point’ for commercial developers, but so far academic researchers have paid little attention to this topic.

6 Conclusion

Academic researchers on NLG and data-to-text usually focus on implementation technology, but building good NLG systems also requires good requirements analysis, design, and testing/evaluation. There are lots of interesting and important challenges in these areas, including

- understanding how and when texts add value to visualisations (*Requirements*)
- principles for designing good data sets for creating data-to-text systems (*Design*)
- techniques for testing non-deterministic NLG systems (*Evaluation/Test*)

Of course there are many more challenges, for example we badly need a better understanding of articulate analytics, narrative generation, high-quality human evaluation, etc.

As well as being important for real-world system building, these challenges are also intellectually interesting and in some cases (such as combining texts and visualisations) should shed light on fundamental cognitive science issues.

I strongly encourage academic researchers to take a broad view of NLG research which includes the above topics!

References

Ankit Arun, Soumya Batra, Vikas Bhardwaj, Ashwini Challa, Pinar Donmez, Peyman Heidari, Hakan Inan, Shashank Jain, Anuj Kumar, Shawn Mei, Karthik Mohan, and Michael White. 2020. Best practices for data-efficient modeling in NLG:how to train production-ready neural models with less data. In *Proceedings of the 28th International Conference on Computational Linguistics: Industry Track*, pages 64–77, Online. International Committee on Computational Linguistics.

Simone Balloccu and Ehud Reiter. 2022. Comparing informativeness of an nlg chatbot vs graphical app in diet-information domain. In *Proceedings of INLG 2022*.

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

Anya Belz, 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.

Timothy W Bickmore, Ha Trinh, Stefan Olafsson, Teresa K O'Leary, Reza Asadi, Nathaniel M Rickles, and Ricardo Cruz. 2018. Patient and consumer safety risks when using conversational assistants for medical information: An observational study of siri, alexa, and google assistant. *J Med Internet Res*, 20(9):e11510.

Daniel Braun, Ehud Reiter, and Advaith Siddharthan. 2018. Saferdrive: An nlg-based behaviour change support system for drivers. *Natural Language Engineering*, 24(4):551–588.

Thiago Castro Ferreira, Claire Gardent, Nikolai Ilinykh, Chris van der Lee, Simon Mille, Diego Moussallem, and Anastasia Shimorina, editors. 2020. *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*. Association for Computational Linguistics, Dublin, Ireland (Virtual).

Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. 2020. Evaluation of text generation: A survey.

Guanyi Chen and Jin-Ge Yao. 2019. A closer look at recent results of verb selection for data-to-text NLG. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 158–163, Tokyo, Japan. Association for Computational Linguistics.

Emily Dinan, Gavin Abercrombie, A. Bergman, Shannon Spruit, Dirk Hovy, Y-Lan Boureau, and Verena Rieser. 2022. SafetyKit: First aid for measuring safety in open-domain conversational systems. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4113–4133, Dublin, Ireland. 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.

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

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating training corpora for NLG micro-planners. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 179–188, Vancouver, Canada. Association for Computational Linguistics.

Albert Gatt and Emiel Krahmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *J. Artif. Int. Res.*, 61(1):65–170.

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

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.

E. Goldberg, N. Driedger, and R.I. Kittredge. 1994. Using natural-language processing to produce weather forecasts. *IEEE Expert*, 9(2):45–53.

Karin Harbusch and Gerard Kempen. 2009. Generating clausal coordinate ellipsis multilingually: A uniform approach based on postediting. In *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009)*, pages 138–145, Athens, Greece. 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.

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.

Mihir Kale and Abhinav Rastogi. 2020. Template guided text generation for task-oriented dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6505–6520, Online. Association for Computational Linguistics.

Tom Knoll, Francesco Moramarco, Alex Papadopoulos Korfiatis, Rachel Young, Claudia Ruffini, Mark Perera, Christian Perstl, Ehud Reiter, Anya Belz, and Aleksandar Savkov. 2022. User-driven research of medical note generation software. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 385–394, Seattle, United States. Association for Computational Linguistics.

Tom Kocmi, Christian Federmann, Roman Grundkiewicz, Marcin Junczys-Dowmunt, Hitokazu Matsushita, and Arul Menezes. 2021. To ship or not to ship: An extensive evaluation of automatic metrics for machine translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 478–494.

Anna Law, Yvonne Freer, Jim Hunter, Robert Logie, Neil Mcintosh, and John Quinn. 2005. A comparison of graphical and textual presentations of time series data to support medical decision making in the neonatal intensive care unit. *Journal of clinical monitoring and computing*, 19:183–94.

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.

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

Simon Mille, Anya Belz, Bernd Bohnet, Thiago Castro Ferreira, Yvette Graham, and Leo Wanner. 2020. The third multilingual surface realisation shared task (SR'20): Overview and evaluation results. In *Proceedings of the Third Workshop on Multilingual Surface Realisation*, pages 1–20, Barcelona, Spain (Online). Association for Computational Linguistics.

François Portet, Ehud Reiter, Albert Gatt, Jim Hunter, Somayajulu Sripada, Yvonne Freer, and Cindy Sykes. 2009. Automatic generation of textual summaries from neonatal intensive care data. *Artificial Intelligence*, 173(7):789–816.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Ratish Puduppully and Mirella Lapata. 2021. Data-to-text generation with macro planning. *Transactions of the Association for Computational Linguistics*, 9:510–527.

Ehud Reiter. 2018. A Structured Review of the Validity of BLEU. *Computational Linguistics*, 44(3):393–401.

Ehud Reiter, Albert Gatt, François Portet, and Marian van der Meulen. 2008. The importance of narrative and other lessons from an evaluation of an NLG system that summarises clinical data. In *Proceedings of the Fifth International Natural Language Generation Conference*, pages 147–156, Salt Fork, Ohio, USA. Association for Computational Linguistics.

Ehud Reiter, Somayajulu Sripada, Jim Hunter, Jin Yu, and Ian Davy. 2005. Choosing words in computer-generated weather forecasts. *Artificial Intelligence*, 167(1):137–169. Connecting Language to the World.

Ehud Reiter, Somayajulu G Sripada, and Roma Robertson. 2003. Acquiring correct knowledge for natural language generation. *Journal of Artificial Intelligence Research*, 18:491–516.

Fahime Same, Guanyi Chen, and Kees Van Deemter. 2022. Non-neural models matter: a re-evaluation of neural referring expression generation systems. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5554–5567, Dublin, Ireland. Association for Computational Linguistics.

Anastasia Shimorina and Anya Belz. 2022. The human evaluation datasheet: A template for recording details of human evaluation experiments in NLP. In *Proceedings of the 2nd Workshop on Human Evaluation of NLP Systems (HumEval)*, pages 54–75, Dublin, Ireland. Association for Computational Linguistics.

Charese Smiley, Vassilis Plachouras, Frank Schilder, Hiroko Bretz, Jochen Leidner, and Dezhao Song. 2016. When to plummet and when to soar: Corpus based verb selection for natural language generation. In *Proceedings of the 9th International Natural Language Generation conference*, pages 36–39, Edinburgh, UK. Association for Computational Linguistics.

Somayajulu G. Sripada, Ehud Reiter, Jim Hunter, and Jin Yu. 2003. Generating english summaries of time series data using the gricean maxims. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, page 187–196, New York, NY, USA. Association for Computing Machinery.

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

Craig Thomson, Ehud Reiter, and Somayajulu Sripada. 2018. Comprehension driven document planning in natural language generation systems. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 371–380, Tilburg University, The Netherlands. Association for Computational Linguistics.

Kees van Deemter, Emiel Krahmer, and Mariët Theune. 2005. Squibs and discussions: Real versus template-based natural language generation: A false opposition? *Computational Linguistics*, 31(1):15–24.

Kees van Deemter and Ehud Reiter. 2018. Lying and computational linguistics. *The Oxford Handbook of Lying*.

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*, 67:101151.

Marian van der Meulen, Robert H. Logie, Yvonne Freer, Cindy Sykes, Neil McIntosh, and Jim Hunter. 2010. When a graph is poorer than 100 words: A comparison of computerised natural language generation, human generated descriptions and graphical displays in neonatal intensive care. *Applied Cognitive Psychology*, 24:77–89.

Michael White and David M. Howcroft. 2015. Inducing clause-combining rules: A case study with the SPaRKy restaurant corpus. In *Proceedings of the 15th European Workshop on Natural Language Generation (ENLG)*, pages 28–37, Brighton, UK. Association for Computational Linguistics.

Dell Zhang, Jiahao Yuan, Xiaoling Wang, and Adam Foster. 2018. Probabilistic verb selection for data-to-text generation. *Transactions of the Association for Computational Linguistics*, 6:511–527.