Data-to-text systems as writing environment

Adela Schneider and **Andreas Madsack**
Johanna Heininger and **Ching-Yi Chen** and **Robert Weißgraeber**
AX Semantics, Stuttgart, Germany
{firstname.lastname}@ax-semantics.com

Abstract

Today, data-to-text systems are used as commercial solutions for automated text production of large quantities of text. Therefore, they already represent a new technology of writing. This new technology requires the author, as an act of writing, both to configure a system that then takes over the transformation into a real text, but also to maintain strategies of traditional writing. What should an environment look like, where a human guides a machine to write texts? Based on a comparison of the NLG pipeline architecture with the results of the research on the human writing process, this paper attempts to take an overview of which tasks need to be solved and which strategies are necessary to produce good texts in this environment. From this synopsis, principles for the design of data-to-text systems as a functioning writing environment are then derived.

1 Introduction

Natural Language Generation (NLG) systems are computer systems that automatically generate texts in human languages, using advanced techniques from artificial intelligence and/or computational linguistics (Carlson, 2015). Non-academic NLG systems are used in different areas of text production and result in fundamental changes for content creation and publication processes: They form a new type of writing technology and create a new environment for humans in which texts are generated automatically, but humans still (co-)design the rules and specifications for this generation.

While NLG systems based on pre-trained large language models function more as writing assistants for authors on an individual level, the NLG systems that are the subject of this study have a different aim: They are configured to be able to produce large amounts of text automatically.

In this context, writing is regarded in a broader sense and means creating a blueprint for producing specific texts. So this new type of writing can be described as meta-writing: However, since the requirements of text structure, expression, and realisation of a communication goal cannot be solved on an abstract level only, many traditional writing tasks remain to be done by the author. Mahlow and Dale (2014) have described this new condition as follows: "Automated text production – when the author is not the writer". This observation raises the question, what a writing environment should be like in which a machine is guided by an author to write a text?

In this research, we use the framework of creating a writing environment to set out the requirements for an NLG system. So, the human writer is considered here as the agent, while the software functions as the environment. This setting is due to the fact that writing, in general, is primarily perceived as an individual action, even though some instances of writing are performed in collaboration. But of course, it is not the only possible framework. The interaction between humans and machines has recently been discussed, especially in the communicative field of AI, where both humans and the instances of AI are seen as agents and the aspect of collaboration is much more prominent (for the field of journalism: Lewis et al. (2019); for fiction writing: Manjavacas et al. (2017); Clark et al. (2018)).

And indeed, it may be that statistical approaches and deep-learning methods, in particular, bring the software's autonomy more to the fore. Autonomy is, after all, the distinctive property of the agent (Henrickson, 2018). This then would call for a reassessment of the situation, looking more closely at the requirements of collaboration within this described environment. However, data-to-text systems in real-world applications still require such a share of human configuration and control and the creative contribution share of the software, at least in the NLG systems focused on in this paper, is still so limited that it would not be adequate to claim

Proceedings of the First Workshop on Intelligent and Interactive Writing Assistants (In2Writing 2022), pages 1 - 10
May 26, 2022 ©2022 Association for Computational Linguistics

creative autonomy of the software in the process.

The environmental framework with its orientation towards the writing processes also offers the advantage of shifting the focus in the evaluation of NLG systems (Howcroft et al., 2020) from the evaluation of the output to an evaluation of the processes, that Gehrmann et al. (2022) recently postulated: "Evaluating NLG tasks only through the lens of outputs is thus insufficient and we should strife (sic) to deliver a more fine-grained breakdown (...)". For traditional writing it is set that the principle of having control over the writing and editorial processes is the most effective method of influencing text quality (Wyss, 2013; Perrin, 2001). And we assume it remains valid also for working with NLG systems. Thus, our approach could open up new perspectives for the evaluation of NLG systems.

What Perrin stated in 2002 for writing per se also applies to automated text production nowadays: "Writing is thus changing from a field of largely intuitive language design to a language technology that becomes aware of its compositional principles and purposefully uses its means, tools, and strategies" (translated from German (Perrin, 2002, page 7)).

As a starting point to achieve such an awareness and methodology for this new kind of writing, including a system of rules, strategies, and cues that guide action, we want to make the action steps, tools and decisions within the processes explicit:

1. In order to approach this, we take a look at the structure and design of NLG systems, because from these the special requirements and conditions are derived to which the user is subject with their text generation task. (*The different categories of NLG systems* and *Overview of the NLG pipeline*)

2. To identify the factors that are conducive to the production of (good) texts, we will outline how the human writing process is organized (*A model of the human writing process*). In doing so, we will refer to the results of writing process research as well as to the approaches to the development of modern writing software. (*Requirements for writing software*)

3. With these findings in mind, we try to take a closer look at automated text production with NLG systems. How can the phases of NLG systems be coordinated with the human writing process? And how should the parameters of the various phases be designed so that texts can be produced with good quality? (*NLG systems in real life: writing on a meta level*)

4. As a result, we will formulate the requirements for the design of NLG systems that take into account the human writing process (*Principles for designing NLG systems*). These requirements ensure creating an environment in which the production of complex written texts is possible. The texts generated in this way should use the full potential of language and not just provide simple data descriptions

2 The different categories of NLG systems

In the basic reference work on NLG it is characterized as ‘the subfield of artificial intelligence and computational linguistics that is concerned with the construction of computer systems that can produce understandable texts in English or other human languages from some underlying non-linguistic representation of information‘ (Reiter et al., 2000).

There are already a number of implemented applications for the data-to-text approach in different areas. They range from the media sector, where they have been a much-discussed topic as "robot journalism", to medical reports, business and finance reports or product descriptions in e-commerce. NLG systems are useful when large amounts of text are needed or information is only available in formats that are not easily understood (such as measurement data from medical examinations), and verbalisation facilitates or enables understanding.

In this study, a further classification concerning the organization of NLG systems is to be discussed. On the one hand there are the so-called *pipeline solutions* that modularize the procedures and then execute the tasks (one after the other). The *end-to-end solutions* on the other hand leave the modular approach behind and aim for *end-to-end* generation based on the successes of deep learning. They can be trained with (data, text) tuples that can be efficiently collected at scale (Castro Ferreira et al., 2019; Harkous et al., 2020). Large pre-trained language models such as GPT-3 or BERT can be integrated into all of these solutions.

At present, end-to-end solutions are not yet suitable for commercial production of great amount of texts because they have two fundamental limitations: First, they are very domain-bound, so they can only generate texts for very limited segments. In addition, they lack semantic fidelity, this means how accurately the generated text conveys the meaning (Harkous et al., 2020). As described, end-to-end systems based on deep learning combine all NLG steps in one function. This means that the only possible intervention is to select or edit the results (Gehrmann, 2020). Due to this too tight restriction of interaction these approaches fall out of consideration for this research. Modular data-to-text systems, on the other hand, offer more points of connection and reflect parallels between humans and systems in the text generation process.

Since this study analyses the application under real-life conditions, the focus is on implementable solutions, not on academic NLG projects. In the commercial sector, rule-based pipeline solutions are established first and foremost, which differ in handling, architectures and purposes. Some of the solutions are offered as self-service, requiring limited or no programming skills. The leading companies in this fields are ARRIA NLG, Narrative Science, AX Semantics, Yseop and Automated Insights (Dale, 2020).

3 Overview of the NLG pipeline

There are different ways to structure the tasks and decisions of text generation. The most cited model for this is the NLG architecture constructed by Ehud Reiter and Robert Dale that performs tasks in sequence related to document planning, sentence planning and linguistic realization (Reiter et al., 2000).

Module	*Content task*	*Structure Task*
Document planning	Content determination	Document structuring
Microplanning	Lexicalisation Referring expression Generation	Aggregation
Realisation	Linguistic realisation	Structure realisation

Table 1: Overview over the most important modules and tasks in the NLG pipeline (Reiter et al., 2000)

The function of the *Document Planner* is to specify the text's content and structure based on domain and application knowledge about what information fits the specified communication goal and other generating objectives. In this module decisions are made about which information will be included (*Content determination*) and in what order this information will appear (*Document structuring*).

The task of the *Microplanning* component is to take the results of the Document Planning module and refine it to produce a more detailed text specification. At this point, sentences and paragraphs are planned (*Aggregation*) and the linguistic elements to be used to express the information are determined (*Lexicalisation*), i.e. which specific words or certain phrases are to be used. Within the *Referring expression generation*, it is decided which properties are used to describe an object unit, for example, a person's proper name and profession. It is therefore necessary to determine which properties are important so that the reader can identify the object.

In the process of *Surface Realisation*, the system converts abstract representations of sentences into grammatically well-formed text (*Linguistic realisation*) and ensures that the abstract structures of sections and paragraphs are assembled as a document in the appropriate format.

4 A model of the human writing process

From the best-known model that illustrates how human writing functions at the cognitive level – the so called Flower-Hayes-Model (Flower and Hayes, 1981) – three important features can be derived that are characteristic of the human individual writing process:

1. There are distinguishable cognitive processes.
2. These processes are organized recursively.
3. Text passages that have already been written have an influence on further text production.

The three distinguishable cognitive processes are controlled by a monitor. This central executive directs attention and switches from one sub-process to another.

The first process is *planning* of a text, where information is collected and thoughts are made about the form and structure of the text. What should the text achieve? Whom does it address? What aspects, data, information should appear in it? It comprises three types of sub-operations: First there is *generating*, in which the writer retrieves information relevant to the writing task from long-term memory. Then there is *organising*, during which the most useful of the retrieved elements are arranged in a plan; finally the writer sets further goals to guide the writing (*goal setting*).

After the *planning* follows the phase of linguistic implementation (*translating*). While many ideas in the planning phase are not really linguistically available, a kind of translation process now takes place during which the thoughts are translated into language: One now decides on the concrete vocabulary.

The third main process is *reviewing* with its two sub-operations *editing* and *revising*. Now, the writer re-reads the text and aims to improve the quality of the written text by changing the text at the time it was written to correct errors, or fit the plans (*editing*). Or they intentionally revise the text to look for problems and errors at all levels of the text (*revising*).

4.1 Recursiveness in writing

One of the most important findings of Flower and Hayes, which is also confirmed by later analyses of the writing processes, is the observation that writing is recursive: The writer jumps back and forth between the processes, again and again. There is no sequential proceeding in which one process is completed and then the next begins.

In principle, it is possible to activate any process at any time, but it can be seen that the frequency and duration of the processes change throughout a writing session. The activation of translating remains constant while that of planning decreases and that of revision increases (Olive, 2004). In the concrete act of writing, the *recursive procedure* shows itself in different facets:

- There is no fixed sequence of the individual operations. It seems that the individual writer develops certain patterns of sequences that remain relatively stable (Olive, 2004).
- Individual activities always refer to each other and overlap.
- All processes can be repeated as often as required.
- Each formulation can be the trigger for a subsequent revision, which results in a new formulation, which in turn can be a trigger for another new formulation.

Text passages written previously have influence on the further text and the arrangement of the processes. Reading and rereading the actual text is an important mental process in which the idea of the text is compared with the actual implementation. The deviations either lead to immediate changes in the written text or to a modification of the idea of the text - which, of course, in the further course of time influences both the text that is still being written and corrections of the pre-existing text passages.

5 Requirements for writing software

In general, technology and writing have always been interdependent: the writing tool and the writing medium influence writing in terms of how the problems at hand can be solved. In most writing settings today, the pen, pencil or typewriter has ceased to be the tool, and paper is no longer the medium. Rather, computers, tablets and smartphones with input functions and screens are the extended writing environment today (Mahlow and Dale, 2014).

The writing environment in the narrower sense is the associated software. There have been and still are approaches to investigate which conditions serve the authors to write without interference and receive the appropriate support during the writing process.

The investigation of the results of writing process research played an important role in this context (Sharples, 1999). It was criticised that the writing tool and the medium were not included in former research. The most important results of the critique are, first, Sharples' (Sharples, 1999) re-evaluation of recursiveness and writing phases and second, the description of certain objects (external mental representations) as a bridge between the writer's ideas and the emerging text. He emphasises the biphasic nature of two activities within the writing process: *engagement* - this means the actual writing, where new material is created and *reflection*, the thinking (about the writing) where the generated material is revised. The two processes are separate and cannot occur simultaneously, forming cycles of engagement and reflection in writing (Sharples, 1999).

From these results, guidelines for the development of writing environment software were derived (Sharpies and Pemberton, 1990) with elaborating the following aspects:

- Because one cannot think about the structure of the text while writing, it is necessary to have a macrostructure (a kind of plan of the text), but this cannot be kept in our working memory. One needs an external representation of these macrostructures (Sharples, 1999).
- It must be possible to store mental representations of information (which can be in linear

language or in other forms such as networks, mind maps, drawings or structures).

- Writers need to be able to switch quickly between tasks (i.e. between notes, outline and linear text or spell check) this facilitates the interleaving of tasks.
- Writers need to switch freely between different parts of the document as well, and should simultaneously be able to choose an appropriate level of focus. So, they should have an overview display and then be able to zoom in. At the same time, it has to be possible at all levels to delete or merge parts of the text or to change the order.

Today there are a handful of software tools that take this non-linear writer-centred approach as their starting point (such as PageFour, Liquid Story Binder, RoughDraft (discontinued), Ulysses, Scrivener), but they tend to be used for specific professionalised, often narrative, writing (Bray, 2013).

However, functions are built into conventional text processors as well that support individual phases of the writing process, such as the outline view, comment functions, text and grammar checks (Piotrowski and Mahlow, 2009).

6 NLG systems in real life: writing on a meta level

At this point, the phases of the human text production process and the modules of the NLG pipeline architecture are juxtaposed in order to find out which principles can be derived for an NLG system that is not designed for experts, but as a writing environment for the (automated) production of large numbers of texts.

6.1 NLG: document planner – human writing: conceptual planning

The characteristic of this phase lies in the significance of alignment with the overall goals of the text: What are the interests of the target audience? What are the communication goals? This provides orientation for the selection of content and the structuring of the resulting text.

The result depends on what goal is to be achieved with the texts and in which environment the text should be published. The editorial strategies as well as the narrative angles for the stories are developed.

In individual writing, the writer derives such text assignments and keeps them either in long-term memory or in the form of a text brief or sample text. How detailed such specifications are worked out depends on the text assignment and the experience of the writer.

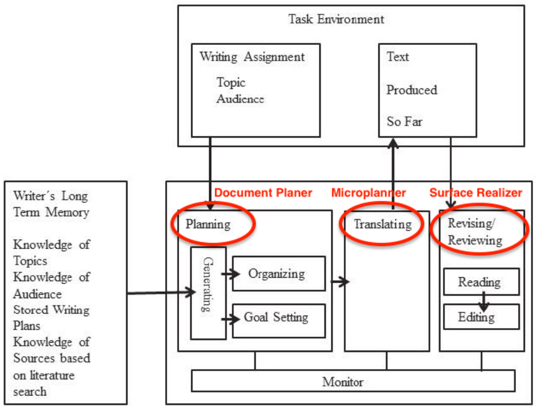

Figure 1: The Flower-Hayes-Model, Flower and Hayes (1981)

In NLG systems the output of the document planner is a document plan which is a structured and ordered representation of messages. Often it is realized in form of a tree, whose leaf nodes are messages and whose internal nodes specify document elements such as paragraphs and sections and discourse relations between the elements. The representations of this plan are partly structural in nature, partly they are already connected with verbal elements (Reiter et al., 2000; Gatt and Krahmer, 2018).

Up to now humans were in most cases also responsible for designing handcrafted rules during the planning phase of automated text production, but there are some examples for developments of modelling genres with Machine Learning and statistics as long as there is a corpus of manual written text available for this specific case (Reiter and Williams, 2010).

At this point, it is worth considering how to transfer the author's implicit knowledge about the communication goal, text genre and document structure into explicit knowledge, such as rules, which can be applied to text generation. Many approaches are possible for the production of such a machine-processable document plan by the writer: The necessary elements can be requested via a kind of questionnaire or forms can be filled out, based on briefing forms (Reiter and Williams, 2010). Since in both areas a form of (internal) representation is created, that is still not translated into words, and for the reasons outlined above, namely that human

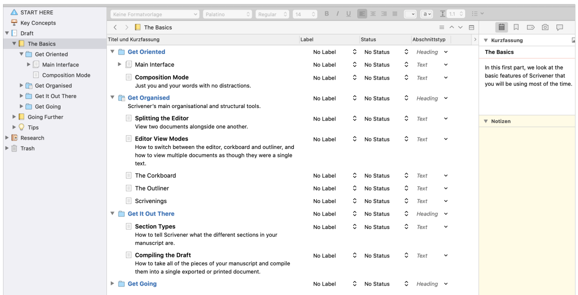

Figure 2: This is the main view in the writing software *Scrivener*. This is an example how a graphic representation with verbal elements can look. On the right there is the option to label and annotate the text parts. (Source: https://www.literatureandlatte.com/scrivener/overview)

mental representations of the document structure are often visual, graphic solutions are a suitable choice. A good example for this is the main page of the writing software *Scrivener (Literature and Latte)* (see Figure 2).

Also in this phase, knowledge and information is inserted either by collecting data and doing research by the human author or by working with the database in NLG systems. In NLG systems data has to be filtered, mapped and combined to achieve the information needed. The results are semantic representations of information which are often expressed in logical or database languages (Gatt and Krahmer, 2018). Commonly, in these systems the authors link particular data situations into abstract meaning which then can be used to trigger specific statements, phrases or document planning decisions. During production, data situations of the various data sets are then evaluated by the system and possible choices determined and executed upon. Especially compared to end-to-end neural systems, this makes sure that all aspects in the output are grounded in the underlying data.

6.2 NLG: microplanning (aggregation) – human writing: text structuring

In this step, it is decided in which order information should appear in a text. As with planning what content is to be included, the orientation towards the reader group and the communication goal also applies here.

In addition, there are some basic rhetorical rules and conventions for the individual text genres. For example, there is a rhetoric rule to place more general information at the top, while the details appear further back. In journalistic text forms on the other

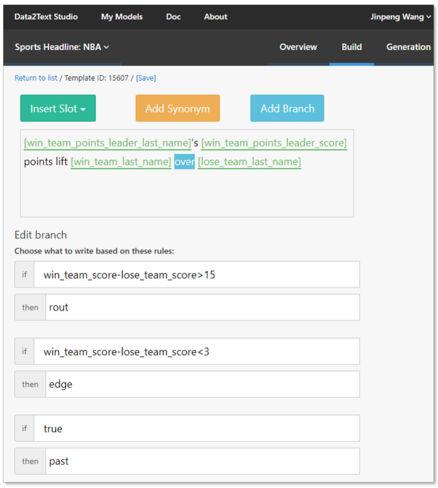

Figure 3: This is a view of the logical structures of the statements and the first step to translating into language. (Data2Text Studio Interface, source: (Dou et al., 2018))

hand, the news, i.e. the special points, are mentioned first, while more general information comes later. There are some recent approaches to use machine learning techniques for content structuring, but since the text structure is very domain-oriented, its design is still produced on the basis of handwritten rules.

This is where requirements for different levels of focus (Sharpies, 1992) come into play: It is advisable to be able to name or label the sentences and to represent them graphically so that they can be arranged by drag and drop, for example. Via the graphical representation, one can then access the assigned sentence and the appropriate data in order to be able to make changes at this level.

6.3 NLG: micro planner – human writing: translating

In this phase, the resulting nonverbal knowledge is translated into actual language. Now decisions have to be made about the words used and the syntax of the text.

In NLG systems, one would basically have to transfer the non-linguistic concepts developed in the document planning phase directly into lexical elements. However, this is not easy for various reasons.

First, the aspect of vagueness, which is tolerated in natural language, plays a major role here. Statements that are transferred as closely as possible directly from the data into words lead to a precision that is quickly perceived as unnatural in natural language. A certain degree of vagueness is necessary for expression in human languages.

The second basic difficulty with this transferring task is that there are always several different ways to verbally describe a piece of information or an event. So there is not one solution for this task, but always multiple ones (Gatt and Krahmer, 2018). For example, Reiter et al. (2005) discussed time expressions in the context of weather-forecast generation. A direct transfer of these time stamps into a record leads to the described overprecision (*At 3:14 it was raining*). Reiter et al. (2005) are also pointing out that e.g. a timestamp 00:00 could be expressed as *late evening*, *midnight*, or simply *evening*. Not surprisingly, humans show considerable variation in their lexical choices.

Another consequence of this direct transmission would be the uniformity of expression, which is usually poorly tolerated in a text. If, for example, in weather texts a rise in temperature occurs several time and is expressed as follows:

```
[time]+ [temp. rise in degrees]
+ {the temperature rose by}
```

The weather report for a day would look like this: *In the morning the temperature rose by 4 degrees. In the afternoon the temperature rose by 5 degrees. In the the evening the temperature rose by -2 degrees*.

First the verbal expression *rise* for a negative rise would be *fall*. And in addition, such a formal structure would be identified very quickly and classified as unreadable. For this reason, several linguistic expressions must be available for a single pre-linguistic event, which are then selected by the system either randomly or based on a formulated condition derived, for example, from the communication goal or the rhetorical strategy. These linguistic variations also serve to ensure sufficient variance in the production of serial texts (see Figure 4).

The formulation of a larger set of different expressions for the data events is a task that in NLG systems still has to be performed by writers and is basically subject to the same principles as in the human language process.

Unlike planning, the phase of *translate* is not related to spatial-visual functions of memory, but rather to phonological working memory: In principle, it is as if the writer now hears the words they write (Olive, 2004; Kellogg et al., 2007). An abstract representation such as a plan or a formula does not provide support during this phase. For this reason, the user is always shown a real-time preview of what a possible instance of the statement would actually look like. Only in this form a statement can be *heard*.

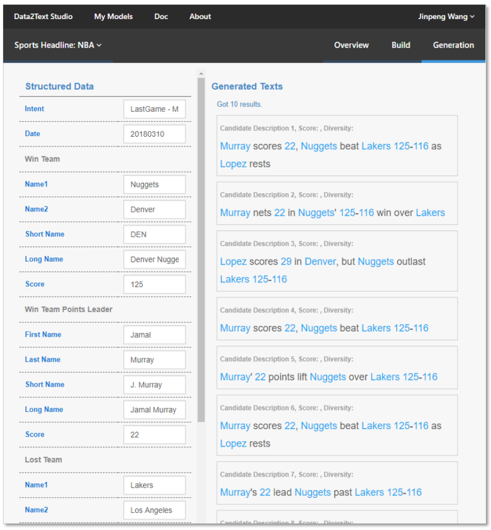

Figure 4: This is a preview of multiple generated text for one data set to guarantee variance. (Data2Text Studio Interface, source: (Dou et al., 2018))

In this manner the user first develops an abstract formulaic representation of the text, then takes the intermediate step via preview and subsequently inserts the corrections into the formula (as an example of a separated preview table see Figures 3 and 4).

The sequence of this procedure, however, narrows the linguistic range of expression in comparison to the conventional formulation of an event. At this point, it is more suitable to give the writer the opportunity to phrase the sentence on the basis of a specific data set as if they were only producing an individual text. And only in a second step express the formula for this expression by providing the software with the labels and logics that it needs for further processing and that it cannot itself recognise on the basis of the text produced.

At this stage, the application of an AI-based component is feasible. They can deliver suggestions based on e.g. keywords or paraphrases of the sentences created by the writer. Just as described earlier, the self-written text and the suggestions of the software take over the function of the *already written text passages*, which in turn can lead to new ideas for the next sentence or to revisions of previous text parts.

6.4 NLG: surface realizer – human writing: reviewing

Linguistic Realisation is concerned with mapping the phrase specifications to the specific words and syntactic constructs which the target language provides such as making subject and verb agree, capitalizing the first letter of a sentence or building the correct plural of a noun. Most decisions in this stage are related to grammar (Reiter et al., 2000).

There are three approaches for implementing this task into NLG systems (Gatt and Krahmer, 2018): Human-written templates that are easy to control, but require a lot of time and effort and offer only limited variability for texts; rule-based systems that make their choices on the basis of the grammar of the language; and statistic related solutions that rely on corpus data.

In the human writing process, an important part of these tasks is already accomplished in the verbalisation phase, but the validation of linguistic and grammatical accuracy takes place in the review phase. For checking syntax and grammar in the native language, the author usually relies on their linguistic intuition and looks up rules in case of doubt. In principle, however, they immediately recognise whether a concrete sentence is syntactically correct or not.

It is less simple for them to assess correctness on the basis of abstract representations. For this reason, a separate review process for linguistic expression and correctness always has to be carried out on the basis of a sample of generated texts. In order to strategically adjust this review, it should be possible to compile this sample group on the basis of different criteria, such as the selection of specific evaluation data sets.

It is noteworthy that NLG systems offer significant advantages in the review process over conventional word processors. Since they retain much more detailed linguistic information about the text, they can perform more targeted correcting operations than word processors. Thus, they fulfil the requirements that Piotrowski and Mahlow (2009) have formulated as to how a software must look like that supports the writer in their editing: (1) Specific views for highlighting linguistic phenomena, and (2) functions to perform operations on linguistic units.

With NLG systems every change made in the text is automatically grammatically adjusted to ensure congruency: For example, changing the number of the subject initiates changing the number of the finite verb and vice versa.

7 Principles for designing NLG systems

The following principles for the design of an NLG system can be derived from the observations presented above:

1. **Build modular systems in alignment with the writing processes:** The modular design of conventional NLG systems suits the writer in that it can be used to provide them with the material and environment to support the specific stage of the writing process. Set up separate views for each main process, which are restricted to the processes in terms of their functionality.

2. **Keep tasks flexible**: To comply with the recursiveness of human writing, it must be possible to edit each task at any time. On the one hand, this means that it must be possible to switch between tasks without any obstacles. And secondly, all changes within a task must be immediately passed on to all instances of the system.

3. **Provide external (non-verbal) representations**: In each phase, the writer must be able to draw on material that are not yet available as linear text. This includes not only overviews of the planning or outlines, but also the option of notes on the existing data material, formulated conditions, templates or text sections.

4. **In the planning view give preference to visual information**: This ranges from representations of the structure to illustrations of logics and data material.

5. **Facilitate the possibilities for linguistic expression**: The writer should always be able to write concrete sentences (without having to include formulas or other abstractions). Provide vocabulary or synonyms and ensure that the writer has the option of formulating multiple variations for the same statement.

6. **Display instances of real text**: The instance of a real text remains an important variable in the process. Only when real text is visible and editable linguistic creativity and grammatical correction can be adequately implemented. Even though this type of automated

text production has different requirements as the production of an individually written text.

7. **Enable linguistics-based editing**: In rule-based data-to-text NLG systems, there is enough meta-information about the grammatical structure of the text that can be used for this task.

8 Conclusion

We showed that there are considerable similarities between the NLG modules and the writing phases of humans in terms of the tasks and decisions involved, which is a significant prerequisite for establishing these systems as a new extended writing technology.

The analysis of these processes is of particular relevance in that quality assurance for data-to-text systems – whose goal is the mass generation of texts – is only attainable by optimizing the processes, since an evaluation of the entire output is not feasible.

However, it also became clear that the human writing process has special features that need to be taken into account when designing NLG systems, especially the consistent and fast change between the processes and the distinctive cognitive activities that require access to different components of the human working memory (e.g. visio-spatial or phonological loop). To neglect these characteristics would mean confining the human involved in a linear process and to strict rules of formal language (i.e. code) to produce natural language texts. This kind of environment would impede the capacity of human writing and, with it, the quality of the text generated. In other words, it would stand in the way of a further successful development of the *technology of writing* which is to be expected in the course of adapting NLG systems in text production.

References

Nancy Bray. 2013. Writing with scrivener: A hopeful tale of disappearing tools, flatulence, and word processing redemption. *Computers and Composition*, 30(3):197–210.

Matt Carlson. 2015. The robotic reporter. *Digital Journalism*, 3(3):416–431.

Thiago Castro Ferreira, Chris van der Lee, Emiel van Miltenburg, and Emiel Krahmer. 2019. Neural data-to-text generation: A comparison between pipeline and end-to-end architectures.

Elizabeth Clark, Anne Spencer Ross, Chenhao Tan, Yangfeng Ji, and Noah A. Smith. 2018. Creative writing with a machine in the loop: Case studies on slogans and stories. In *23rd International Conference on Intelligent User Interfaces*, IUI '18, page 329–340, New York, NY, USA. Association for Computing Machinery.

Robert Dale. 2020. Natural language generation: The commercial state of the art in 2020. *Natural Language Engineering*, 26(4):481–487.

Longxu Dou, Guanghui Qin, Jinpeng Wang, Jin-Ge Yao, and Chin-Yew Lin. 2018. Data2Text studio: Automated text generation from structured data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 13–18, Brussels, Belgium. Association for Computational Linguistics.

Linda Flower and John R. Hayes. 1981. A Cognitive Process Theory of Writing. *College Composition and Communication*, 32(4):365–387.

Albert Gatt and Emiel Krahmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation.

Sebastian Gehrmann. 2020. *Human-AI collaboration for natural language generation with interpretable neural networks*. Ph.D. thesis, Harvard University.

Sebastian Gehrmann, Elizabeth Clark, and Thibault Sellam. 2022. Repairing the cracked foundation: A survey of obstacles in evaluation practices for generated text. *ArXiv*, abs/2202.06935.

Hamza Harkous, Isabel Groves, and Amir Saffari. 2020. Have your text and use it too! end-to-end neural data-to-text generation with semantic fidelity. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2410–2424, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Leah Henrickson. 2018. Tool vs. agent: attributing agency to natural language generation systems. *Digital Creativity*, 29(2-3):182–190.

David M. Howcroft, Anya Belz, Miruna 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 *INLG*.

Ronald T. Kellogg, Thierry Olive, and Annie Piolat. 2007. Verbal, visual, and spatial working memory in written language production. *Acta Psychologica*, 124(3):382–397.

Seth Lewis, Andrea Guzman, and Thomas Schmidt. 2019. Automation, journalism, and human–machine communication: Rethinking roles and relationships of humans and machines in news. *Digital Journalism*, 7:1–19.

Cerstin Mahlow and Robert Dale. 2014. Production Media: Writing as Using Tools in Media Convergent Environments. In Eva-Maria Jakobs and Daniel Perrin, editors, *Handbook of Writing and Text Production*, volume 10 of *Handbooks of Applied Linguistics*, pages 209–230. De Gruyter Mouton, Berlin, Germany.

Enrique Manjavacas, Folgert Karsdorp, Ben Burtenshaw, and Mike Kestemont. 2017. Synthetic literature: Writing science fiction in a co-creative process. In *Proceedings of the Workshop on Computational Creativity in Natural Language Generation (CC-NLG 2017)*, pages 29–37, Santiago de Compostela, Spain. Association for Computational Linguistics.

Thierry Olive. 2004. Working memory in writing: Empirical evidence from the dual-task technique. *European Psychologist*.

D. Perrin. 2002. *Schreiben: von intuitiven zu professionellen Schreibstrategien*. Westdt. Verlag.

Daniel Perrin. 2001. *Wie Journalisten schreiben. Ergebnisse angewandter Schreibprozessforschung*. UVK Verlag.

Michael Piotrowski and Cerstin Mahlow. 2009. Linguistic editing support. In *Proceedings of the 9th ACM Symposium on Document Engineering*, pages 214–217.

Ehud Reiter, Robert Dale, and Zhiwei Feng. 2000. *Building natural language generation systems*, volume 33. MIT Press.

Ehud Reiter, Somayajulu Sripada, Jim Hunter, Jin Yu, and Ian Davy. 2005. Choosing words in computer-generated weather forecasts. *Artificial Intelligence*, 167:137–169.

Ehud Reiter and Sandra Williams. 2010. Generating texts in different styles. In *The Structure of Style*, pages 59–75. Springer.

Mike Sharpies. 1992. Representing writing: External representations. *Computers and Writing: State of the Art*.

Mike Sharpies and Lyn Pemberton. 1990. Starting from the writer: Guidelines for the design of user-centred document processors. *Computer Assisted Language Learning*, 2(1):37–57.

Mike Sharples. 1999. *How we write. Writing as creative design*. Routledge.

Vinzenz Wyss. 2013. *Qualitätsmanagement in Redaktionen*, number 1 in Aktuell : Studien zum Journalismus, pages 89–105. Neuberger, Christoph and Meier, Klaus, Baden-Baden.