

# Automated Text Generation and Summarization for Academic Writing



Fernando Benites, Alice Delorme Benites, and Chris M. Anson

**Abstract** In this chapter, we discuss the implications of automatic text generation for academic writing. We first review the current state of the technology and how it is being used. We then discuss the implications of using automatic text generators for academic writing, including the need for users to be aware of the limitations of the technology and how to use it effectively. We also discuss how the use of automatic text generation can change the traditional stages of writing, and how the content generated by these systems is not justified by semantic or extra-linguistic criteria. We finally argue that notions that have been useful for explaining, analyzing, and teaching academic writing will need to be re-examined in the light of human-machine-interaction. (This abstract has been automatically generated using OpenAI and slightly post-edited; see this article's Appendix for an explanation.)

**Keywords** Natural language generation · Machine learning · Human-machine interaction · Academic writing · Text production · Language modeling

## 1 Introduction

In the twenty-first century, academic writing mostly takes place with a minimal setup of at least a computer, a text processor, and an internet connection. In this context, computers are often used to relieve human writers of specific tasks like correcting spelling mistakes, providing the results of library or internet searches, and

---

F. Benites (✉)

Institute for Data Science, University of Applied Sciences and Arts Northwestern Switzerland,  
Windisch, Switzerland  
e-mail: [fernando.benites@fhnw.ch](mailto:fernando.benites@fhnw.ch)

A. Delorme Benites

Zurich University of Applied Sciences, Winterthur, Switzerland  
e-mail: [alice.delormeбенites@zhaw.ch](mailto:alice.delormeбенites@zhaw.ch)

C. M. Anson

North Carolina State University, Raleigh, NC, USA  
e-mail: [chris\\_anson@ncsu.edu](mailto:chris_anson@ncsu.edu)

© The Author(s) 2023

O. Kruse et al. (eds.), *Digital Writing Technologies in Higher Education*,  
[https://doi.org/10.1007/978-3-031-36033-6\\_18](https://doi.org/10.1007/978-3-031-36033-6_18)

organizing scientific references into standardized quotations. Yet the author, who actually performs the task of choosing the words and the order they will be presented in, is still human. Or is it? Automated text generation has undergone significant advances in the last few years and is likely to redefine human–machine writing interaction in the near future.

Procedural text generation is actually not a new concept: in the seventeenth century, the German poet Georg Philipp Harsdörffer had the idea of designing a *volvelle*—a contraption of several circles whose combination produced words and sentences according to their respective position (<http://whitneyannetrettien.com/the-sis/>). Centuries later, in the era of computers, natural language generation (NLG) long relied on the same principles: combining words, very much like building blocks, using a set of rules in order to produce a text. For decades, automated systems have used templates, so that for each text to be produced, only some slots must be filled. These templates were very specific, as they gathered formulations designed for each language, for each domain, for each document type, and so on. As a result, maintaining such templates and keeping them up to date was a laborious and tedious task, and they performed better with highly standardized texts. This is why these text generation systems were employed mostly in domains such as weather reports (e.g. the *Pollen Forecast for Scotland* system [Turner et al., 2006]), sports news, and financial reports. The idea was to turn structured data, which was stored in databases, into text, hence automating the additional tedious work of organizing the data into a coherent text. The main goal of these NLG systems was to produce intelligible and relevant information only, regardless of the style or the repetitiveness of such texts. To that extent, such an approach might not seem compatible with the production of academic texts: academic writing is bound to language-specific and domain-specific conventions, but it also requires a certain amount of fluency and readability in order to engage readers. It works from the structure of the text up to the idiomaticity and how to express certain relations. Most importantly, the text should be written in a way to keep the reader interested and guide them through a discovery, or it should point attention towards key information. Even more, the ongoing competition for publication and acceptance of conference or even journal articles makes it unavoidable to consider questions such as style, rhetorical decisions, and even repetition (and its forbidden form, plagiarism—see Anson, 2022). This concern, and more generally the overall urge for intrinsic novelty in every academic publication, should discourage academic writers from using the aforementioned systems to produce their papers; however, it could be argued that such systems might act as “writing assistants” for more fluent, extended, and original text.

## 1.1 Core Idea of the Technology

To understand current developments in automatic text generational and natural language processing, it is helpful to trace their history in AI research. In the early 1980s, AI experimentation was partly designed to explore human language

processing to inform computer-based processing. Work at Yale Artificial Intelligence Labs, particularly by Roger Schank and colleagues (see Schank & Abelson, 1977), succeeded in generating texts that appeared to be written by humans, with a level of acceptable structure, coherence and cohesion, and lexical accuracy. One program, “Talespin,” was designed to create stereotypical “Aesop”-like stories with anthropomorphized characters and simple plots (Meehan, 1976, 1977). However, the errors generated in the automated story production process yielded insights into what information a computer needs to work effectively with natural language. In particular, the lack of sufficient world knowledge created significant problems, especially concerning plans, actions, preconditions, and logical outcomes. For example, early in the development of TaleSpin, the program produced stories such as the following:

Joe bear was hungry. He asked Irving Bird where some honey was. Irving Bird refused to tell him, so Joe offered to bring him a worm if he’d tell him where some honey was. Irving agreed. But Joe didn’t know where any worms were, so he asked Irving, who refused to say. So Joe offered to bring him a worm if he’d tell him where a worm was. Irving agreed, but Joe didn’t know where any worms were, so he asked Irving, who refused to say. So Joe offered to bring him a worm if he’d tell him where a worm was .... (Meehan, 1977, p. 91)

Meehan explains the source of the problem: “Don’t put a goal on the stack if it’s already there. Try something else. If there isn’t anything else, you can’t achieve that goal.”

The programming for these tales takes a traditional form of rule-codes called planboxes, linguistically instantiated, that include details about plans, goals, actions, what a character knows, etc., as illustrated in the following:

Planbox 1: X tries to move Y to Z

preconditions:

X is self-movable  
 If X is different from Y,  
 then DPROX (X, X, Y)  
 and DO-GRASP (X, Y)  
 DKNOW (X, where is Z?)  
 DKNOW (X, where is X?)  
 DLINK (X, loc (z))

act: DO-PTRANS (x, y, loc (z))

postcondition:

Is Y really at Z? (DKNOW could have goofed)

postact: If X is different from Y, then DO-NEG-GRASP (X, Y)

Through multiple trials and errors, these rule codes can be refined, each iteration showing what else is required for the production of even simple tales with logical plots.

Another important requirement for natural language production and interpretation involves the role of inferencing. Consider these pairs of sentences:

Paula's dog slipped its collar on a busy street. The veterinary bills were obscene.  
René drank a fifth of vodka at the party. The morning was unpleasant.

To either interpret or produce these texts, a program first needs the semantic knowledge to understand the word "slip" in the context of a dog getting out of its collar (as opposed to "slip on the ice"). It then needs to infer that when a dog slips its collar, it can run from its owner and is likely to be injured in traffic on a busy street, and that such an injury will require the intervention of a veterinarian who bills the dog's owner. It also has to know that "obscene" can be used to describe not only something pornographic or grotesque but outrageous in a general and negative sense, and it must know that very high bills are unpleasant to most people. In the second pair, the program needs to know that a "fifth" is a liquor bottle size, and that drinking a fifth of vodka typically causes a highly displeasing physical reaction the next day. In both cases, this knowledge is not propositional in the statements but resides in world knowledge activated between the sentence pairs: it is *implied* in production and *inferred* in reception.

Schank and colleagues proposed categories of world knowledge required to understand and generate text. These included scripts (typical sets of actions, such as those that entail at a fast-food restaurant vs. an expensive fancy restaurant), props (such as menu boards vs. printed menus in leather folders), roles (such as order and cashier personnel vs. a maitre d', a head waiter, a bread waiter, and a sommelier), plans, and goals. By themselves, roles such as waiter, maid, carpenter, banker, etc., activate many assumptions that do not need to be stated in language but are inferred. Consider the following sentence:

The police officer held up her hand and stopped the car.

Any program working with natural language needs to know that the police officer has role-authority to cause driver to use the brakes to stop the car, not that she physically stopped it herself.

Schank and Abelson (1977) detail the kind of programming required to yield natural-language outputs that make sense. But the extent of knowledge required to generate or interpret text was, at the time, almost insurmountable for humans to program into a computer system. Consequently, this approach to NLG was replaced following the advent of artificial neural networks and modern natural language processing. Algorithms are now learning from textual data at a breathtaking pace, especially since the amounts of data being available on the Internet are increasing as quickly as the processing capacities of computers. Machine learning methods allow computers to observe the data and infer their own rules from it and, in essence, imitate what they have observed so far. In particular, self-supervised deep learning methods can not only extract word frequencies from large amounts of text, but also construct word correlations that allow the creation of very fluent texts. This technology is already widely developing for translation, and neural machine translation solutions like Google Translate or DeepL are now freely available to all Internet

users, offering fluent, idiomatic, and often accurate translations. The quick rise of such machine translation engines that are now omnipresent on websites, social media, and handheld devices hints at a similar explosion of automated text generation solutions in the near future, especially because the underlying technology of machine translation is text generation. But just as the use of machine translation entails pitfalls and requires a specific set of skills and knowledge to avoid them, using automated text generators for academic writing purposes will require a basic understanding of their affordances and a heightened awareness of their risks and potential (see Anson & Straume, 2022). Therefore, a closer look into these machine learning approaches is justified.

Automatizing and standardizing writing processes is not new in academic writing. Numerous phrasebooks and collections of stereotypical formulations, templates, and writing guides have been published over the years in an attempt to speed up the writing process. Such ready-made formulaic blocs can seldom help with other writing issues, such as overcoming writing block or anxiety, citing related works more rapidly (e.g., turning citations into readable text), rephrasing and paraphrasing, and summarizing findings. These are areas of interest to AI-based programmers and AI-application users. In fact, several attempts have been made to create an algorithm that can write a scientific abstract or even a full paper on its own; one such paper was even submitted for publication (Thunström & Steingrimsson, 2022).<sup>1</sup>

Besides the creation of new content, academic writing also encompasses a variety of summarizing tasks: writing a literature review, for example, can be considered a multi-source text summarization activity. It is also quite usual to summarize one's own text in a short abstract that will help potential readers to decide whether a paper's content is relevant for their research or not. This type of single-source text summarization is particularly current in the academic context. Automatizing such tasks could prove useful, especially since summarization is less bound to novelty and originality than academic text production in general. Yet automatic text summarization presents other challenges: summing up facts, abstracting, and generalizing might require general, contextual information that the system does not possess. In the worst case, this could lead to the system stating new and inaccurate facts. Further, deciding which elements of a text are to be mentioned in a summary and which ones can be left out usually relies on our human understanding of the text's content, and could pose a problem for an automatic system.<sup>2</sup> To that extent, while summarization is an inherent part of academic writing, automatic text generation and automatic text summarization are usually considered two distinct yet related fields. Both are rather large fields, which is why we will provide only a brief overview in this chapter. There is, of course, already abundant research focusing on various related fields (such as

---

<sup>1</sup> <https://www.scientificamerican.com/article/we-asked-gpt-3-to-write-an-academic-paper-about-itself-mdash-then-we-tried-to-get-it-published/>, accessed 19.08.2022, remark: since the references are wrong it probably won't pass the first review. However, we recommend the reading of the paper, since it shows how a fluent text can hide such mistakes.

<sup>2</sup> When "supervised," of course, such systems can still benefit writers who spend a fraction of the time it would take to write an abstract simply ensuring that the output of the system is accurate.

chatbots, machine translation, question and answer generation, and next word prediction); for extended surveys and reviews, see Yu et al. (2022), Celikyilmaz et al. (2020), and El-Kassas et al. (2021).

## 2 Functional Specifications

### 2.1 Rule-Based Systems vs. Neural/Statistical Methods

A very early version of a text generation system can be seen in chatbots such as ELIZA, which was developed in the 1960s. Many generations since then, the systems employ different methods that can be divided into rule-based or neural/statistical-based. The rule-based methods are triggered by words found in a given sentence: they replace a variable (missing word) in the template with a value according to the context and return this filled template. Neural/statistical methods work differently: they learn correlations between words so that they can either find the right context (intent classification) or predict the words that should more likely come next. When using intent classification, they can find the right values to fill predefined templates or even generate a response directly. Statistical methods usually work with a set of rules extracted from a learning corpus, whereas neural methods rely on neural networks architectures, also trained on selected corpora. Neural networks generalize better to unseen input data, but they can also derail and create nonsensical content (Fyfe, 2022). They are the current state of the art, which is why a closer look at their inner workings will help to explain the stakes of automatic text generation.

### 2.2 Neural Networks

Artificial neural networks (ANNs) are inspired by biological neurons (McCulloch & Pitts, 1943). In that regard, they build an abstract representation of these: the signal from one neuron to another neuron can either be intensified or repressed. A popular base building block (neuron) of ANNs is the Perceptron (Rosenblatt, 1957), which sums the input signals and decides they should be repressed or passed through. Moreover, the connections between the neurons create a network; consequently, it is only with enough input strength of the connected neurons that a given neuron is activated and passes the input signal through. In that sense, each neuron acts as a gatekeeper. Each connection is also referred to as a parameter in ANNs (there are, in fact, generally two parameters for each connection, a weight and a bias). Each parameter is usually set to a random number and needs to be adjusted through training. By presenting examples with input values and output values, a neuron based on the input can produce an output; the difference between true and produced output (error/cost function) is used to adjust the parameter values and therefore learn. However, a neuron alone cannot differentiate complex problems, given the logic rule with two

input parameters (Minsky & Papert, 1969). For example, let us imagine the following set of parameters:

- a: = I am eating
- b: = I am talking
- c: = I am polite

Within the context of a formal dinner I am invited to, if both parameters a and b are true (I am eating, and I am talking), then parameter c is false, since it is usually considered not very polite to speak with a full mouth. On the other hand, one could usually expect that guests engage in conversations and that they at least try the food that is served to them. As a result, not eating and not talking (i.e., a and b being both false) will also make me impolite, and variable c would be false again. If I either eat or talk (a or b being true), then I am polite and c is true. A single neuron cannot solve problems like this, also called a non-linear separable problem.

In order to solve such problems, neural networks have to use multiple neurons, usually structured in layers (multilayer Perceptron). The more layers, the more complex the problems that can be solved. In general, there is no rule determining how many layers a problem of certain complexity needs. However, the more layers a network has, the more calculations it needs to adjust each parameter. Therefore, very large networks are expensive in terms of time, computing power, money, and ultimately their carbon footprint. Nevertheless, there are techniques to train large networks with fewer resources. One of them is to have the system not learn all samples at once, but in batches, where each batch encompasses a certain number of examples. A hyperparameter called the learning rate adjusts how much the new batch influences the network's parameters in order to accommodate the new examples, but also how many of the examples of the previous batches can be discarded (and thus partially forgotten by the network). It is very difficult to estimate how big a network needs to be and how to train such large networks so that they keep everything correct. The problem is aggravated when the data are not perfect, which is almost always the case. As humans tend to disagree quite quickly on many issues, large amounts of texts will yield contradictory claims about their quality or relevance. It is not clear how such varying claims are processed by the neural networks, since they cause a paradox for the learning algorithms.

Moreover, text cannot be processed in its raw state by neural networks; it first needs to be transformed into numerical values. Numbering all the words creates a huge amount of data (the English language is estimated to have between 400,000 and 600,000 words). This results in an enormous range of randomly assigned numbers without semantic or logical organization or connection between them. Neural networks cannot handle input data well in this form. The solution is to use so-called word-embeddings (Mikolov et al., 2013), in which the networks are trained to predict a word given the context it appeared in. Therefore, the network will learn which words are similar to each other and occur in the same context.

A common representation of word-embeddings resembles a basic algebra of words, or word analogies, with vectors; for example, the symbol  $v$  stands for a vector word-embedding representation of the word:

$$\text{Yen}_v - \text{Japan}_v + \text{U.S.}_v \approx \text{Dollar}_v$$

This representation allows us to understand how semantic relations between words are handled within the networks. By making the networks bigger (with more and deeper layers) and using a similar learning routine (predicting masked words), developers can allow the networks to effectively learn a given language, creating so-called language models.

However, this approach also has its limits: it does not fare well with context-dependent words such as the homonym *bank* (financial institution? park bank? river bank?). Newer approaches using recurrent neural networks (Hochreiter & Schmidhuber, 1997) take the order of the word in the input texts into account, but still cannot solve the issue completely. Thus, the sequential approach often leads to signal losses, especially for long sequences. In terms of text writing, this would, for example, lead to coreference and negation problems. This is why a progressive shift started in 2017, when such issues were overcome with the advent of so-called transformers, the current state of the art.

## 2.3 Transformers

A transformer is a neural network architecture introduced by Vaswani et al. (2017) and is composed of different neural networks, called an encoder and a decoder.<sup>3</sup> An input text is transformed into a prediction, in other words, an output text. More specifically, the input text is first encoded into a representation (in a so-called latent space), which is more independent of the source language and then can be decoded into the target language. Further, transformers use a method called attention, more specifically self-attention, which tries to put words into the overall context of the input text. A further aspect is that the original input signal is propagated throughout the neural network.<sup>4</sup> Therefore, the network is able to learn which word fits which context more quickly than other architectures, e.g. multilayer Perceptron, although transformers and other architectures rely on similar base building blocks.

Devlin et al. (2018) presented a method to train transformer architectures (Vaswani et al., 2017) so that the machine learning model would predict words within a partially incomplete sentence (usually 15% of the words are masked or removed) by using only the encoder side of a transformer. The model used a 3.3-billion-word corpus and went multiple times over this mass of textual data, and was able to perceive which words often occur in which context. This method, known as BERT, is very popular, especially for text summarization.

---

<sup>3</sup> When applied to text, such encoder-decoder architectures are also called sequence to sequence architectures.

<sup>4</sup> This is also called *skip connections*.



Contrary to BERT, another development, GPT, uses a decoding side of a transformer. This can be applied to next-word generation, where the words are “uncovered” from left to right, and the system guesses which word will best fit given the whole context provided on the left. Using this technique, the model can predict words and generate text. GPT’s successor, GPT-3, massively increased the amount of processing data and the number of parameters to be adjusted by the model. This increase in training translates into a greater generalization of the model. The advent of GPT-3 also brought prompting, a new machine learning method that quickly gained popularity. Usually, learning and defining a new task—unknown to the machine—constitute a separate part of the machine learning process associated with great costs and large numbers of samples. GPT-3 allows this step to be performed with much lower resources and outside the actual machine learning process. This is why GPT (in versions GPT-3.5 and GPT-4) and its public-facing ChatGPT is currently one of the most popular models for text generation.

For summarization and translation, the transformer architecture here thus reads the text and transforms it into another text. However, the length of a single input text in Chat-GPT is somewhat limited to, around 3000 words (except for the GPT proprietary system, whose GPT-4 currently allows around 7,000 words but a model allowing 25,000 words is already announced). Larger models and models based on other techniques are being developed (see Beltagy et al., 2020; Zaheer et al., 2020), but it could take some time until their release for production, especially because the evaluation of large amounts of texts is very complex and requires high computing resources.

## 2.4 *Evaluation*

Whenever artificial intelligence is used to perform a task, the question of quality evaluation and metrics arises. There is an evident need for objective, measurable, and comparable evaluation scores to assess how well a given system performs. Manual evaluation is surely valuable but expensive; neural networks systems usually have many settings used for creating a model and assessing if it learned enough, so that at the end hundreds of model states need to be compared. Estimating the quality of the systems, and choosing the best among them, is preferably performed without human intervention. For that purpose, there exists a range of automatic evaluation metrics (AEMs). They are different from a key component in machine learning: loss or error/cost function, which allows the machine to learn what is correct and incorrect, and thus change the parameter values of the model accordingly. This is usually calculated based on a human-produced reference text collection. AEMs evaluate the quality not for single samples (texts) but at corpus level; thus, they can measure further aspects, such as the recurring types of errors, or which words are more often wrong.

For text generation, evaluation is carried out by removing parts of the reference sentences and having the system complete those sentences. A comparison between the system’s suggestions and the original reference sentences will provide

an evaluation score. For summarization, the system's output is compared with a human-produced summary of the same input text. There are usually several rounds of evaluation (called iterations), and each iteration can use a different reference text, which allows taking into account the diversity of human writing styles. A major issue regarding the choice of adequate reference texts for evaluation ultimately relies on subjective criteria, given that human text evaluation has long been subject to discussion and debate.

### 2.4.1 Perplexity

One popular way to estimate the quality of the language model underlying a text generation system is the called perplexity (Jelinek et al., 1977). This metric tells us if a model generates text very close to the training data, i.e., if it catches the essence of the language by identifying which words are more likely to follow which words. When a text is generated, if its perplexity is low, it will correlate with scores of fluency, i.e., human evaluators would consider the text fluent. This allows an estimation of quality without having to manually annotate an extra reference corpus. In contrast, other sets of measures rely on manually created and annotated source and target sets of texts. Such measures can help to assess more precisely word accuracy, for example, and will be presented in the following section.

### 2.4.2 BLEU, ROUGE, METEOR

Bleu, Rouge, and Meteor are the most popular metrics for summarization, although originally designed (and still extensively used) for machine translation evaluation. They measure the number of words and word sequences (n-grams) that are shared by the text produced by the machine and a reference text. As such, they can measure different types of overlap (ROUGE: Recall-Oriented Understudy for Gisting Evaluation—see Lin, 2004—and even take the text length (BLEU: BiLingual Evaluation Understudy—see Papineni et al., 2002) and word order (METEOR: Metric for Evaluation of Translation with Explicit ORdering,—see Banerjee & Lavie, 2005) into account.

Since the essence of translation is to produce a text equivalent to a source text, at least for simple and less creative translation tasks, the constraints given by the source text usually restrict the field of possibilities for formulations. In that regard, machine translation is a rather guided and homomorphic process (i.e., where the structure of the data is preserved), and it makes sense to evaluate the system by looking for matching text sequences between multiple human translations and the system's output. Nevertheless, these metrics do not evaluate whether the meaning of a text is correctly conveyed—they merely check if the right words have been used, sometimes not even considering if they are in the correct order.

This issue is even more problematic when these metrics are used to evaluate summarization systems. Summaries often imply a targeted rephrasing and restructuring of a text's contents, usually writing *in other words*, for example by using more hypernyms to replace several terms at once. In that sense, text summarization inherently contains a change in perspective, a zooming out of the depicted content, and hence is based on the fact that the same content can be described with various levels of details in very different ways. This difference in the level of abstraction can be a challenge for word-based automatic evaluation such as BLEU, ROUGE, or METEOR.

To solve this problem, other metrics related to information retrieval could be applied to text summarization evaluation in order to verify that the most important information has been kept. However, in the case of abstractive summarization, where an entirely new text is created, it is a difficult task to verify that the same information is present in the source text and in the summary. If the information was corrupted in the summarization process, it is not clear yet how automatic methods can detect and assess the quality of the produced summary.

Models that generate good summaries according to these automated metrics can then be evaluated by humans. Popular criteria for manual evaluation of automatic text summarization methods are coherence, consistency, fluency, and relevance (see Fabbri et al., 2021 for a detailed description). However, these evaluations are often very subjective and difficult to compare across studies, since they seldom use the same data set and evaluators.

As we can see, the question of quality evaluation is not resolved yet. It is important to bear these limitations in mind when working with automatic text generation and/or summarization systems, especially since industry's claims tend to give a more enthusiastic and less rational view. While comparing the different automatic evaluation scores of various systems might be helpful, one should not forget that automatic metrics are not bound to human evaluation logic (as we know it from the evaluation of school essays, for example) and should be interpreted within their respective scope only.

## 2.5 Text Generation

Neural network models are the latest turn in a long history of artificial intelligence methods that require enormous amounts of digitalized texts and processing power. In that sense, it is questionable whether this should really be called intelligence, and not brute force. Nevertheless, it is precisely the huge quantity of textual data that makes a decisive difference between neural approaches and older text generators: rule-based text generation systems simply did not cover enough of the target language to produce texts that appear natural or intelligent. Even large systems of simple rules could not grasp a word's context of use. Neural networks, on the contrary, showed even more capacity to generalize as thought was possible, with relatively simple architectures. Yet it is important to bear in mind that, while both rule-based and machine learning

systems can somehow mimic human intelligence, they do not understand the words that they are processing.

However, systems based on neural networks can usually handle correlations. For example, if we put “Mr. President Barack” in a neural text generator, the system will most likely predict “Obama” as the next word. Such correlations, along with many others, might be interpreted as the machine’s knowledge. But unlike humans, the machine only has the knowledge and intelligence for the specific task it has been trained to perform: for example, it proved very difficult to train text processing systems to do basic arithmetic<sup>5</sup> (Hendrycks et al., 2021). Therefore, systems that provide next word predictions or paraphrasing might change the meaning of one’s writing or suggest something basically wrong for the writing task at hand. Yet because the next best word is predicted given the context calculated on a vast amount of document collections (billions of words), the system’s suggestion usually appears fluent and “intuitive” in light of the rest of the sentence, which makes it even more difficult to spot a possible inconsistency.

While the idea of knowledge and intelligence is to be taken with caution when related to the machine, there is an undisputable amount of information contained in the vast text collections that neural networks use to generate texts. To that extent, text generation could also be a means for human users to acquire the knowledge stored in the networks. For example, entering “the president of the United States in 2016 was Barrack Hussein Obama” triggered the following suggestion for continuation: “The current president of the United States is Donald Trump” (generated by open-GPT-3 on August 12th, 2022).<sup>6</sup> This shows how text generation not only produces written outputs to express ourselves, but also provides users with new knowledge, ideas, and inspiration.

One undisputable advantage of large language models is hence the enormous amount of information that is stored in them. However, extracting specific information relevant to a given writing task or topic can be challenging, and the systems can mix up different subjects or end up stating false facts. This issue also applies when these systems are used for rephrasing: they can be exact and convey the intended message correctly in other more fluent words, or they can corrupt the input information, but still sound very proficient (Fyfe, 2022). Finally, these language models might simply reproduce the content they were trained with, creating problems related to authorship or plagiarism, or replicating problematic assumptions generalized from large data sets (e.g., that all nurses are women or all pilots are men).

---

<sup>5</sup> A good example was presented in <https://ai.googleblog.com/2020/06/pegasus-state-of-art-model-for.html>, accessed 28.8.2022.

<sup>6</sup> The generated information is already outdated at the time the authors are writing this article (summer 2022).

## 2.6 Text Summarization

Summarization is a very important part of scientific writing, such as creating an abstract for a paper or reviewing a group of papers. Though at first, both tasks might seem similar, they differ to various extents: multi-source summarization requires normalizing different papers to the same vocabulary, ontology, and group of concepts; distilling a certain approach or perspective to the research questions targeted by that group of texts; establishing which of the research subject points were compatible and how to compare different methodologies; and so on. This is a highly complex task for experienced researchers, requiring not only contextual understanding, but also abstraction skills to compare and synthesize knowledge. As described in Benitez “[Information Retrieval and Knowledge Extraction for Academic Writing](#)”, identifying the important words in the individual documents of a collection is a more or less solved task. However, summarizing multiple documents for a given research question requires a different approach—often the question and answering type (Dimitrakis et al., 2020), which has not yet been solved for that context (Durmus et al., 2020). The current technology is not designed to summarize the actual *knowledge* contained in documents, but to extract the most important words or sentences according to what the machine has learned from an annotated corpus. Basic approaches use TF-IDF<sup>7</sup> or similar technologies (e.g. bm25), where the idea is to find words that are particularly frequent in a specific document within a collection and hence have a certain degree of uniqueness related to this document. This procedure can also be applied to full sentences. Such an approach is described as extractive summarization, as it mainly consists in extracting unique and frequent words or phrases *as is* and “glue” them together to fabricate a summary.

Extractive summarization is often opposed to abstractive summarization, where entirely new text is generated to capture the essence of the original text(s). State-of-the-art abstractive summarization methods apply large, pretrained language models. These language models are learned in a self-supervised way, i.e., they undergo a pretraining stage where they learn to predict words according to a given context or to identify which sentences tend to follow each other, and which do not. Although they can overcome many linguistic ambiguities (homographs, homonyms, etc.), their task remains more complex when multiple sources are involved.

Another form of summarization, although not directly producing a text, is called topic modelling: the content units of a document collection (i.e., the words) are grouped by co-occurrence. This allows hundreds of documents to be overviewed and give an impression of the topics covered by a specific collection or corpus. It is then possible, in another step, to transform the topic lists or graphs into fluent text. This method is currently mostly used by linguists and specialized researchers, and further research is required to understand how knowledge can be extracted efficiently through this procedure.

---

<sup>7</sup> See chapter [Information Retrieval and Knowledge Extraction for Academic Writing](#).

### 3 Main Products

There is currently a wide variety of automatic text generators emerging on the market, the majority of them mainly aiming at content creation and copywriting (for example, Zyro, Jasper, and Rytr, especially for e-mail writing). They usually offer AI-based generation of blog posts, social media posts, search engine optimized texts, and marketing content. A smaller proportion of those online tools explicitly focus on academic writing.

One of the oldest systems, SCIGen, explicitly aimed to amusingly critique the overgenerous acceptance rate of some conferences. The code shows that the generation is rule-based and uses many scientific idioms,<sup>8</sup> as it draws from the science repository CiteSeer. Although the developers claim that their system produces “non-sensical” articles,<sup>9</sup> the output complies with most formal requirements for scientific publications.

Beside SCIGen’s satirical ambitions, many other “serious” systems are now emerging. We will name and describe a few of them as examples for what is currently available. However, at the moment, the market is constantly evolving, and it is not yet possible to identify major players.

<https://web.writewise.io> is a rule-based tool that offers more than 700 sentence and section templates. However, it also offers a wide range of writing assistance functionalities to “compose clear, coherent, structured, and mistake-free manuscripts.”

<https://myassignmenthelp.com/mah-bot-editor.html> is a free tool that creates essays based on simple keywords, e.g., a given title. Interestingly, this tool relies heavily on human–machine interaction in each step: after entering a title for their essay, users are offered various outputs flagged as the beginning of the text. They can either choose one or decide to write the beginning of the text themselves. After that, users are presented with an editor, where they can type their own text or choose automatically generated paragraphs which they can edit at will. The user interaction also foresees a disclaimer whenever they choose to use a generated paragraph, informing them that the text has been generated out of online resources and can be used at their discretion.

<https://www.essayailab.com/> presents a very similar interface (if not identical) and also provides several suggestions to start with. The provider, however, strongly emphasizes the issue of plagiarism, with disclaimers showing exactly how the generated output has been edited to pass plagiarism checks. The editor’s interface is very similar to the one found on <https://myassignmenthelp.com/mah-bot-editor.html>, but it offers more prompts and pop-ups to guide users through the writing process. Both tools also provide help with grammar checks and many more services, all mostly based on the same text generation technology.

The issue with plagiarism is also raised on another website, <https://smodin.io/writer>, that displays a constant disclaimer that because “articles are generated from

---

<sup>8</sup> <https://github.com/strib/scigen/blob/master/scirules.in>.

<sup>9</sup> <https://pdos.csail.mit.edu/archive/scigen/>.

content on the web, it can be considered plagiarism. It is recommended to rewrite the scraped content.” While this website only presents various output suggestions but no editor, it offers a function specifically called “remove plagiarism,” very similar to the paraphrasing feature offered by most other tools. <https://smodin.io/writer> seems more targeted at producing content to be copied and used as is, and less focussed on integrating the text generation technology into a broader writing process.

<https://www.writefull.com/> is an example of a fully different approach to merging technology and human writing. It mainly works as a plugin for text editors (e.g., Word) and offers feedback and paraphrase suggestions. It also offers a range of free online tools, like a paraphraser, a title generator, an abstract generator (abstractive summarization), and a collection of sentence patterns sorted by section (introduction to conclusion).

As we can see, these tools can vary greatly in their interface and in the underlying understanding of the writing process. However, most of them draw from similar text generation and/or summarization technology, whose most prominent example is GPT-4. GPT-4, currently one of the largest language models, is usually employed as chat and backend for general text generation, as well as for numerous writing solutions offered online. Many new tools (such as Copy.AI, neuroflash, or open.ai, to name only a few) are based on it (or a similar technology). This means that the text (or the keywords) entered on these websites is sent to the GPT-4 API, its answer collected and then presented to the user on the website. With the right prompting (given by the user themselves or by the service provider), GPT-4 can write a scientific article that seems very convincing at first sight<sup>10</sup> (however, the citations are definitely wrong and other content problems cannot be excluded). Prompting plays a decisive role in the quality of the generated output. For example, the scientific article written entirely by open-GPT3 was the result of concise prompts for each part of the text (Thunström & Steingrímsson, 2022). Here is an example of such a prompt:

Prompt: Write a methodology section about letting GPT-3 write an academic paper on itself explaining what prompts are. It should include the word Top P, Frequency Penalty, Presence Penalty, Temperature and Maximum length, Best of and how it uses these to create output. Do not give any exact numbers. (Thunström & Steingrímsson, 2022, p. 4)

Finally, new avenues are opening up, for example the idea of generating research questions directly through GPT<sup>11</sup> (Yimam et al., 2020).

## 4 Research

There is animated discussion of the use of AI in writing, especially since the possibilities have become much more fluent in the last years. Anson (2022) discusses the use of AI in the practice of writing and how the authorship concept becomes less clear.

<sup>10</sup> <https://www.scientificamerican.com/article/we-asked-gpt-3-to-write-an-academic-paper-about-itself-then-we-tried-to-get-it-published/> accessed 2022.7.11.

<sup>11</sup> <https://noduslabs.com/research/ai-writing-tool-gpt-3-text-generator-of-research-questions/>

Hutson (2021) discusses different problems specific to open-GPT3, from how the language models are getting bigger and bigger, to measuring fluency, to how these models can be biased, because the language of their training data is neither inclusive nor fair.

Relevant insights can also be drawn from the neighboring neural machine translation (NMT) technology. Research has been documenting many aspects of the translator's perceptions and experiences in their work with AI-produced texts on various levels. Here, two domains of research seem to yield transferable insights for the work with text generators: the textual aspect and the cognitive aspect.

NMT produces fluent text almost instantly and at a very low cost, and many researchers resort to this option to ensure good English quality of their research, yet this quite often does not seem to suffice to match the publishing criteria (Escartín & Goulet, 2020). In fact, current NMT systems still have some problems that users need to be aware of. For example, the fact that terminology might not be translated consistently throughout a single text, that hedging and modality are frequently distorted through the reformulation process (Martikainen, 2018), that cohesive devices within a single text tend to be left out in the target translation, resulting in a loss of logical cohesion (Delorme Benites, 2022), and more generally the presence of algorithmic biases resulting from oversized language models (Bender et al., 2021). Further, there is a growing concern about the observed amplification of societal biases through language technology leading to machine translationese (Vanmassenhove et al., 2021), described as an artificially impoverished language characterized by a loss of lexical and morphological richness.

These issues are particularly problematic for scholarly texts, since academic genres (Swales, 1990) have peculiarities such as terminology, low-frequency words (Coxhead & Nation, 2001; Hyland & Tse, 2007), and hedging (Schröder & Markkanen, 1997). Furthermore, most NMT solutions available to the public work mainly at the sentence level, leading to significant text cohesion problems (e.g., unclear pronoun reference, and the aforementioned inconsistent terminology). As a result, many semantic, pragmatic, and textual aspects are still not treated well with current methods. While there is some research on terminology issues (Thunström & Steingrímsson, 2022; Zulfiqar et al., 2018) and domain adaptation (e.g., Haque et al., 2020), overarching academic text features (general academic vocabulary, neologisms, acronyms, intersentential and intrasentential links, overall text cohesion, claim hedging, rhetorical moves) are rarely or not at all considered. Since automated text generation relies on the same technology as NMT, it is likely to pose similar issues for academic writing purposes.

Another finding from translation research that might apply to automated text generation regards the cognitive aspect of working with AI-produced texts: the user's trust in the machine relies more on the fluency of the text than on its accuracy (Martindale & Carpuat, 2018). As a result, AI-produced texts tend to lull readers into trusting blindly their content, discouraging them from questioning the veracity of the information they are presented with. This is confirmed by many professional translators, who claim that post-editing (proofreading and correcting) a machine-translated text requires much more effort than a human-produced text, especially



since errors are unpredictable. Should this also apply to automatic generated texts, which we can logically expect, there is a clear need to raise awareness and train users to proofread their texts as thoroughly as possible. Here again, techniques from translation research might prove useful.

In addition to these considerations, there is an intrinsic dilemma in using algorithms to produce academic texts: the core idea of scientific writing is to communicate new ideas and insights; sometimes even the very writing process will contribute to generating these ideas. Yet coming up with new ideas is something that machine learning algorithms can't do, since they are built as extremely well performing imitation machines. What generators can do is to lay out suitable sentence structures and idioms, and present information in various styles (scientific vs. marketing, for example). To sum up, text generators are very powerful tools for the formal part of academic texts. However, they cannot guarantee chains of causality in the content they produce (e.g., if  $a > b$  and  $b > c$ , then  $a > c$ ). To that extent, they can easily introduce erroneous claims in their seemingly fluent output. For example, the aforementioned article written entirely by GPT-3 contained anachronistic citations.

More generally, the nature of these systems is cause for reflection: their strength lies in the enormous amount of data they rely on, but we do not know what exactly is stored in these neural net models and how everything is organized. This makes such systems quite unpredictable, and what they can generate or where they can derail still remains unclear. Further, they are trained to predict missing words in a given sentence, but not to assess the actual consequences of each result on readers (for example, creating an offending output), and the only possibilities are the ones given by the documents used as training corpus. Although the corpora are, indeed, very large, they still are only a fraction of what the entire human language corpus.

Nevertheless, the popularity of automatic text generators is growing, especially among non-native Ph.D. students, who need to write their abstracts, papers, and theses in English. This is why they should be introduced and discussed in tertiary institutions, and their potential and risks should be on the agenda of academic writing training programs.

## 5 Implications

Many current practitioners of machine learning are driving the focus on machine learning systems and computer-aided systems, and the idea is not to remove humans completely but rather to find ways that computers can assist humans in repetitive and arduous tasks. This enables humans to oversee these tasks and focus their effort on exceptions and more challenging cases.

We can expect that automatic text generators will be used in various ways, according to the user's needs, competences, and time constraints, among other factors. As a result, we can anticipate at least two approaches to the writing process. There will likely be more, and the differentiation might end up being more fine-grained. Nevertheless, we will only describe these two as examples, keeping in mind

that practices are yet to be established in this fairly new area. First, automatic text generators can produce a first draft that serves as a basis for the actual writing. In that scenario, users would give a rather procedural input (e.g., a bullet point list) to instruct the machine on what they expect. They can then choose from several suggestions, combine them, use only the first paragraph as a starting point, or even read through all suggestions for inspiration before they write their own text. All those strategies have been observed in the post-editing of machine translation output, and usually depend heavily on the user's personality. Second, automatic text generators could also be used *after* the actual drafting phase, for example in order to transform a raw draft into a fluent text and even render it in a specific style (academic, professional, popular, etc.). This approach could benefit, among others, non-native writers, writers with learning disabilities, or persons who struggle with academic writing in general.

Furthermore, various steps of the scientific work itself could already be tackled using automatic text generators, i.e., searching for relevant information, including adequate citations, as well as creating reviews and surveys using text summarization. In that regard, automatic text generation's impact on scientific writing will probably go beyond linguistic or formal considerations. This, in turn, stresses how important the mutual relationship is between science and science writing.

Still, as mentioned earlier (especially in light of machine translation related findings), a fruitful collaboration with the machine in order to produce good academic texts requires that the user knows how to make the best of the possibilities it offers and remain in control of the writing process. This means, in turn, that a lot of effort has to be invested in overseeing the processes and learning how to do so. Further, a generalized use of automatic text generation can ultimately lead to an overflow of documents, probably with a certain stylistic homogeneity. In turn, creativity and human writing skills could make a major difference between just another paper and a much-cited one in the global race for publication.

There is little doubt that automatic text generators can develop into widely-used writing assistance devices, where humans still perform various parts of the writing process. However, it is difficult to foresee precisely how the use of such automated solutions will change the traditional theoretical stages of writing (e.g., planning, prewriting, drafting, and revising). A possible hypothesis is that the planning and the revising phases would then gain in relevance and take up most of the human effort. On the other hand, one can wonder how different uses of automatic text generators can be accounted for in social constructivist theories of writing, especially since the content suggested by automatic systems is not justified by semantic or extralinguistic criteria. Finally, the question of pragmatics should be addressed when using automatic text generators. At the moment, there is no evidence that the systems take textual or pragmatic constraints into consideration; in other words, information structure, intertextuality, and rhetorical development cannot be expected to be part of an automated writing process. Hence, notions that proved useful to explain, analyze, and even teach academic writing, i.e. Swales' (1990) CARS model of rhetorical moves, will need to be re-examined in the light of human-machine-interaction.

## Appendix

### Generated Abstracts

Longformer: in the twenty-first century, academic writing mostly takes place with a minimal setup of at least a computer, a text processor, and an internet connection. in this context, computers are often used to relieve human writers of specific tasks like correcting spelling mistakes, providing results of library or internet searches, and organizing scientific references into standardized quotations. \n however, the author, who actually performs the task of choosing the words and the order they will be presented in, is still human. in the era of computers, \n natural language generation (NLg) long relied on the same principles: combining words, very much like building blocks, using a set of rules in order to produce a text. for decades \n, automated systems have used templates, so that for each text to be produced, only some slots must be filled. however, maintaining such templates and keeping them up to date was a laborious and tedious task, and they performed better with highly standardized texts. besides the creation of new content and summarizing findings, automatic text generation and automatic text summarization are usually considered two distinct yet related fields, which is why we will provide only a brief overview in this chapter.

Openai-GPT-3 (shorten the text to fit 4k Token length, add at the end “abstract:”):

First attempt: In this paper, we discuss the implications of automatic text generation for academic writing. We first review the current state of the technology and how it is being used. We then discuss the implications of using automatic text generators for academic writing, including the need for users to be aware of the limitations of the technology and how to use it effectively. We also discuss how the use of automatic text generation can change the traditional stages of writing, and how it can be accounted for in social constructivist theories of writing.

Second attempt: This article discusses the implications of using automatic text generators for academic writing. It describes how the use of such generators can change the traditional stages of writing, and how the content generated by these systems is not justified by semantic or extra-linguistic criteria. The article also discusses the question of pragmatics when using automatic text generators, and how notions that have been useful for explaining, analyzing, and teaching academic writing will need to be re-examined in the light of human–machine-interaction.

Software	Access	Specificity	Licensing
<a href="https://web.writewrite.io">https://web.writewrite.io</a>	Paid	Writing assistance with templates	Proprietary
<a href="https://myassignmenthelp.com/mah-bot-editor.html">https://myassignmenthelp.com/mah-bot-editor.html</a>	Free	Given simple keywords, essays are automatically generated	Proprietary
<a href="https://www.essayai.com/">https://www.essayai.com/</a>	Paid/free	Uses a database to suggest content, and provides writing assistance	Proprietary

(continued)

(continued)

Software	Access	Specificity	Licensing
Openai-GPT3	Paid	General writing assistance (Language Generation)	Proprietary
T5 (Huggingface)	Free	As Openai-GPT3, but it can be downloaded but requires programming knowledge to generate content	Free

## References

- Anson, C. M. (2022). AI-based text generation and the social construction of “fraudulent authorship”: A revisitation. *Composition Studies*, 50, 37–46.
- Anson, C. M., & Straume, I. G. (2022). Amazement and trepidation: Implications of AI-based natural language production for the teaching of writing. *Journal of Academic Writing*, 12(1), 1–9.
- Banerjee, S., & Lavie, A. (2005, June). 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* (pp. 65–72). Association for Computational Linguistics.
- Beltagy, I., Peters, M. E., & Cohan, A. (2020). *Longformer: The long-document transformer* (arXiv preprint). [arXiv:2004.05150](https://arxiv.org/abs/2004.05150)
- Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021, March). On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency* (pp. 610–623). Association for Computing Machinery.
- Celikyilmaz, A., Clark, E., & Gao, J. (2020). *Evaluation of text generation: A survey* (arXiv preprint). [arXiv:2006.14799](https://arxiv.org/abs/2006.14799)
- Coxhead, A., & Nation, P. (2001). The specialised vocabulary of English for academic purposes. In J. Flowerdew & M. Peacock (Eds.), *Research perspectives on English for academic purposes* (pp. 252–267). Cambridge University Press. <https://doi.org/10.1017/CBO9781139524766.020>
- Delorme Benites, A. (2022). *A constructionist approach to (machine) translation problems: Towards a Theme-Rheme construction*. Cognitextes (forthcoming).
- Deveci, T. (2019). Sentence length in education research articles: A comparison between Anglophone and Turkish authors. *Linguistics Journal*, 13(1), 73–100.
- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). *Bert: Pre-training of deep bidirectional transformers for language understanding* (arXiv preprint). [arXiv:1810.04805](https://arxiv.org/abs/1810.04805)
- Dimitrakis, E., Sgontzos, K., & Tzitzikas, Y. (2020). A survey on question answering systems over linked data and documents. *Journal of Intelligent Information Systems*, 55(2), 233–259.
- Durmus, E., He, H., & Diab, M. (2020). *FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization* (arXiv preprint). [arXiv:2005.03754](https://arxiv.org/abs/2005.03754)
- El-Kassas, W. S., Salama, C. R., Rafea, A. A., & Mohamed, H. K. (2021). Automatic text summarization: A comprehensive survey. *Expert Systems with Applications*, 165, 113679.
- Escartín, C. P., & Goulet, M. J. (2020). When the post-editor is not a translator: Can machine translation be post-edited by academics to prepare their publications in English? In *Translation revision and post-editing* (pp. 89–106). Routledge.
- Fabbri, A. R., Kryściński, W., McCann, B., Xiong, C., Socher, R., & Radev, D. (2021). Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9, 391–409.

- Fyfe, P. (2022). How to cheat on your final paper: Assigning AI for student writing. *AI and Society*. <https://doi.org/10.1007/s00146-022-01397-z>
- Haque, R., Moslem, Y., & Way, A. (2020). Terminology-aware sentence mining for NMT domain adaptation: ADAPT's submission to the Adap-MT 2020 English-to-Hindi AI translation shared task. In *Workshop on low resource domain adaptation for Indic machine translation* (pp. 18–21). Patna, India.
- Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., & Steinhardt, J. (2021). *Measuring mathematical problem solving with the math dataset* (arXiv preprint). [arXiv:2103.03874](https://arxiv.org/abs/2103.03874)
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.
- Hutson, M. (2021). Robo-writers: The rise and risks of language-generating AI. *Nature*, 591(7848), 22–25.
- Hyland, K., & Tse, P. (2007). Is there an “academic vocabulary”? *TESOL Quarterly*, 41, 235–253. <https://doi.org/10.1002/j.1545-7249.2007.tb00058.x>
- Jelinek, F., Mercer, R. L., Bahl, L. R., & Baker, J. K. (1977). Perplexity—A measure of the difficulty of speech recognition tasks. *Journal of the Acoustical Society of America*, 62, S63–S63.
- Lin, C. Y. (2004, July). Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out* (pp. 74–81). ACL.
- Martikainen, H. (2018). Modal markers as potential sources of distortion in translated medical abstracts. *Educational Sciences: Theory and Practice*, 18(4), 917–934.
- Martindale, M. J., & Carpuat, M. (2018). *Fluency over adequacy: A pilot study in measuring user trust in imperfect MT* (arXiv preprint). [arXiv:1802.06041](https://arxiv.org/abs/1802.06041)
- McCulloch, W., & Pitts, W. (1943). A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5, 115–133.
- Meehan, J. (1976). *The metanovel: Writing stories by computer* (Unpublished doctoral dissertation). Yale University.
- Meehan, G. (1977). Tale-spin: An interactive program that writes stories. In *Proceedings of the fifth annual conference on artificial intelligence*.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 26. Curran Associates, Inc.
- Minsky, M. & Papert, S. (1969). *Perceptrons An introduction to computational geometry*. MIT Press. ISBN-0262130432.
- Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002, July). Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the association for computational linguistics* (pp. 311–318). Association for Computational Linguistics.
- Rosenblatt, F. (1957). *The perceptron—A perceiving and recognizing automaton* (Report 85-460-1). Cornell Aeronautical Laboratory.
- Schank, R. C., & Abelson, R. P. (1977). *Scripts, plans, goals and understanding: An inquiry into human knowledge structures*. Routledge.
- Schröder, H., & Markkanen, R. (1997). *Hedging and discourse : Approaches to the analysis of a pragmatic phenomenon in academic texts*. De Gruyter.
- Swales, J. (1990). *Genre analysis: English in academic and research settings*. Cambridge University Press.
- Tehseen, I., Tahir, G., Shakeel, K., & Ali, M. (2018, May). Corpus based machine translation for scientific text. In *14th IFIP international conference on artificial intelligence applications and innovations (AIAI)* (pp. 196–206). Rhodes, Greece. [https://doi.org/10.1007/978-3-319-92007-8\\_17](https://doi.org/10.1007/978-3-319-92007-8_17)
- Thunström, A. O., & Steingrímsson, S. (2022). *Can GPT-3 write an academic paper on itself, with minimal human input?* <https://hal.archives-ouvertes.fr/hal-03701250v1>
- Turner, R., Sripada, S., Reiter, E., & Davy, P. (2006). Generating spatio-temporal descriptions in pollen forecasts. In *Demonstrations* (pp. 163–166). EACL

- Vanmassenhove, E., Shterionov, D., & Gwilliam, M. (2021). *Machine translationese: Effects of algorithmic bias on linguistic complexity in machine translation* (arXiv preprint). [arXiv:2102.00287](https://arxiv.org/abs/2102.00287)
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998–6008). Curran Associates, Inc.
- Yimam, S. M., Venkatesh, G., Lee, J. S. Y., & Biemann, C. (2020). *Automatic compilation of resources for academic writing and evaluating with informal word identification and paraphrasing system* (arXiv preprint). [arXiv:2003.02955](https://arxiv.org/abs/2003.02955)
- Yu, W., Zhu, C., Li, Z., Hu, Z., Wang, Q., Ji, H., & Jiang, M. (2022). A survey of knowledge-enhanced text generation. *ACM Computing Surveys (CSUR)*.
- Zaheer, M., Guruganesh, G., Dubey, K. A., Ainslie, J., Alberti, C., Ontanon, S., & Ahmed, A. (2020). Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33, 17283–17297.
- Zulfiqar, S., Wahab, M. F., Sarwar, M. I., & Ingo Lieberwirth, I. (2018). Is machine translation a reliable tool for reading German scientific databases and research articles? *Journal of Chemical Information and Modeling*, 58(11), 2214–2223. <https://doi.org/10.1021/acs.jcim.8b00534>

**Fernando Benites** is a lecture for Data Science at the University of Applied Sciences and Arts Northwestern Switzerland. He received his Diploma on computer science for natural sciences from the University of Bielefeld. During his Ph.D. at the University of Konstanz, he worked on new classification methods for large amounts of texts with multiple ontologies. He has been working in several areas of natural language processing on various projects (NMT, Chatbots, ASR, NER, NLG).

**Alice Delorme Benites** is a professor in human–machine-communication and co-head of the Institute of Translation and Interpreting at the Zurich University of Applied Sciences. Her research interests include machine translation literacy and the use of NLP technologies by the broader public.

**Chris M. Anson** is Distinguished University Professor and Alumni Association Distinguished Graduate Professor at North Carolina State University (USA), where he is Executive Director of the Campus Writing and Speaking Program. He has published 19 books and over 140 articles and book chapters relating to writing research and instruction, and has spoken widely across the U.S. and in 34 other countries. He is Past Chair of the Conference on College Composition and Communication and Past President of the Council of Writing Program Administrators, and currently serves as Chair of the International Society for the Advancement of Writing Research. His full c.v. is at [www.ansonica.net](http://www.ansonica.net).

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter’s Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter’s Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

