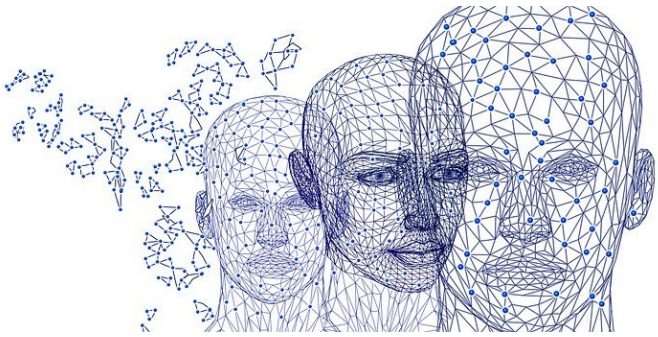
October 2021, Issue Code NL-2022-3

DDD (Digital Data Deception) Technology Watch Newsletter

Table of Contents



"*All warfare is based on deception. Hence, when we are able to attack, we must seem unable; when using our forces, we must appear inactive; when we are near, we must make the enemy believe we are far away; when far away, we must make him believe we are near.*"

— Sun Tzu, *The Art of War*

Editors: Keenan Jones, Enes Altuncu, Virginia Franqueira, Sanjay Bhattacherjee and Shujun Li
Affiliation: Institute of Cyber Security for Society (iCSS), University of Kent, UK
Contact Us: ddd-newsletter@kent.ac.uk

Editorial

A news feature published in *Nature Magazine* on March 2021 (https://www.nature.com/articles/d41586-021-00530-0) discussed how OpenAI's latest natural language model, GPT-3 [6], can be used to automatically generate coherent texts in a variety of contexts. Moreover, the article emphasises that advanced AI models, such as GPT-3, often do not understand what they generate, highlighting the risks of misuse that such technologies could present.

This potential for abuse was highlighted when a university student leveraged GPT-3 to generate fake blog posts using just a title and a brief introduction as the input (https://www.technologyreview.com/2020/08/14/1006780/ai-gpt-3-fake-blog-reached-top-of-hacker-news/). These fake posts were so successful that few suspected they had been generated by an AI engine, with one post reaching the #1 spot on the website Hacker News. Whilst this ended up being a relatively benign prank, the ease with which it was conducted has raised concerns as to what could happen if this was used for more dangerous applications, such as the generation of disinformation or spam. As humans become less able to differentiate between machine-generated and human-created texts, the potential for the weaponisation of powerful text generators as a vector for mass digital data deception (DDD) is, in turn, becoming a serious concern.

In this issue, we will cover text-based fake data at least partly generated by automated algorithms. While there has not been a universally accepted definition of deepfake images, videos and audio/speech data, the definition of text-based fake data (deepfakes and non-deep fakes) is even more ambiguous. In Section 1, we will discuss how we see the definition and how we decided to define the scope of this issue. In a nutshell, we will consider different types of AI-based fake text generation, including those based on deep learning and more traditional machine learning methods.

Text-based fake data is based on different methods for natural language generation (NLG), a key sub-area in natural language processing (NLP) and computational linguistics (CL). Considering the breadth and depth needed to provide a sufficiently useful coverage of NLG, we will use three issues to cover different aspects of this topic. In this issue we will cover general concepts, propose a taxonomy of NLG-related research, and cover three fundamental areas of the taxonomy: NLG methods, evaluation of NLG methods, and the underlying AI techniques that are used. In the next issue, we will focus on different applications and sub-topics of NLG relevant to DDD. In the third issue, we will look at detection of AI-generated texts, attacks on NLG methods and detectors, as well as the current challenges and open questions facing NLG and the detection of AI-generated texts.

To source the articles necessary to derive our taxonomy we opted for a venue-driven approach, selecting a number of relevant review and survey-like papers on AI-based NLG. This approach allowed for the selection of a focused set of relevant papers suitable for this newsletter, whilst still providing a strong overview of the field as a whole.

To this end, we looked at research papers published since 2019 at a number of venues known to have published NLG-related research, including those listed in the ACL (Association for Computational Linguistics) Anthology, all conferences and workshops of ACL's Special Interest Group on Natural Language Generation (SIGGEN), four additional NLG-related conferences not indexed by the ACL Anthology (IALP, CICLing, PACLING, TSD), four major journals related to NLG or for publishing survey papers (*IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *ACM Transactions on Asian and Low-Resource Language Information Processing*, *ACM Computing Surveys*, *IEEE Communications Tutorials & Surveys*, and *IEEE Access*), arXiv.org, and 14 NLG-related Chinese journals.

For some sources, we first used a search query to identify all surveys, systematisation of knowledge (SoK) papers, systematic reviews, taxonomies, ontologies and general reviews, and then screened all returned papers manually to identify NLG-related papers. For other sources, we screened all papers published in the time period (2019-2021) to identify NLG-related survey papers. In addition, we also manually inspected all SoK papers indexed by DBLP. All initially identified papers were further inspected and encoded for exclusion or inclusion in the three issues planned for the topic on text-based fake data. All finally selected papers were, or will be, used to derive the taxonomy presented in Section 1, and

the contents of this and the next two issues.

We hope you enjoy reading this issue. Feedback is always welcome, and should be directed to ddd-newsletter@kent.ac.uk.

List of Acronyms

- ACL: Association for Computational Linguistics
- AI: Artificial Intelligence
- BERT: Bidirectional Encoder Representations from Transformers
- BLEU: Bilingual Evaluation Understudy
- CIDEr: Consensus-based Image Description Evaluation
- CL: Computational Linguistics
- CNN: Convolutional Neural Network
- CTRL: Conditional Transformer Language Model
- DL: Deep Learning
- DNN: Deep Neural Network
- GAN: Generative Adversarial Network
- GNN: Graph Neural Network
- GPT: Generative Pre-trained Transformer
- GRU: Gated Recurrent Unit
- LSTM: Long Short-Term Memory
- MASS: Masked Sequence to Sequence Pre-training for Language Generation
- METEOR: Metric for Evaluation for Translation with Explicit Ordering
- ML: Machine Learning
- MLE: Maximum Likelihood Estimation
- MNMT: Multilingual Neural Machine Translation
- NLG: Natural Language Generation/Generator
- NLP: Natural Language Processing
- NN: Neural Networks
- PLM: Pre-trained Language Model
- PPLM: Plug-and-Play Language Model
- RL: Reinforcement Learning
- RNN: Recurrent Neural Network
- ROUGE: Recall-Oriented Understudy for Gisted Evaluation
- SoK: Systematisation of Knowledge
- SVM: Support Vector Machine
- T5: Text-To-Text Transfer Transformer
- TG: Text Generation/Generator
- UniLM: Unified pre-trained Language Model

1. Definitions and Scope

1.1. From NLG to Text-based Fake Data

Natural language generation (NLG) refers to the process of developing software-based models and systems aimed at creating fluent, human-readable text from a given set of inputs (e.g., text-based datasets, images, prompts) [36]. Given the value in being able to create new texts with limited human assistance, NLG can thus be applied to a wide-range of tasks related to natural language processing (NLP). This range of NLG tasks includes story and poem generation, interactive tasks such as the creation of chatbots and Q&A systems, and the translation of texts from one language to the other [7]. It is worth noting that we consider NLG a distinct subclass of text generation (TG), which includes other, non-natural language generation such as the automatic creation of programming source code (as seen in OpenAI's recent release of the Codex code generator [9]). In this newsletter, we focus on NLG research.

With the rise in deep learning based architectures that are better equipped for handling sequential data, such as recurrent neural networks (RNNs) and Transformers, improvements in NLG have come rapidly in the past few years [8]. These improvements have been particularly notable with the development of powerful language models, typically built around a Transformer architecture, that have been pre-trained on massive datasets to conduct a generic text prediction task (e.g., GPT-2/3, T5, XLNet) [7]. Leveraging these pre-trained models, state-of-the-art performances in NLG have been achieved simply by fine-tuning these models with (relatively) small amounts of data specific to a desired task [48].

As NLG models become more powerful, it is becoming ever harder for humans to distinguish between AI-generated texts and texts written by humans [27]. With these rapid improvements in NLG come concerns over the capacity for generation models to be leveraged to produce fake or otherwise deceptive texts. This could result in a variety of potentially dangerous uses, including the generation of fake news [52] or extremist/radicalising texts [27], and the creation of highly convincing spear-phishing emails [50]. Beyond these more commonly cited (mis)-use cases, however, it is worth noting that the potential capacity for NLG models to fool humans into believing that the texts they are reading are human-created means that all NLG tasks hold the potential for deception and misuse.

Given this broad capacity for deception, we opt to take a wide scope in our approach to fully cover the topic of NLG. This will allow us to present a complete consideration of the myriad ways in which various NLG systems, applied to various tasks, can be adapted for deception.

1.2. A Taxonomy

Based on the NLG-related survey papers identified during the screening process, we derived a taxonomy covering all of the important topics and concepts relevant to NLG. We consider two different types of nodes: classes and attributes. An attribute has two or more values, some of which refer to other nodes. A sub-class is normally defined by one or more attributes of a super-class, in other words, a super-class can be split into multiple sub-classes according to different values of one attribute or different value-combinations of multiple attributes.

It is worth noting that this taxonomy is currently work in progress, and further refinement will be conducted over the following issues of this newsletter. The complete and finalised full taxonomy will be included in NL-2022-5, alongside a discussion of the changes and refinements that have been made.

Currently, we have defined the following high-level concepts that compose the first level of our NLG taxonomy:

NLG Methods: This concept encapsulates the methodological approach taken in order to construct an NLG system. In turn, this concept includes the form of input utilised to generate text, the specific task that the NLG system is directed towards, the training approach taken, and the training data used to build NLG models. We discuss this concept's sub-tree in detail in Section 2.

Evaluation of NLG Methods: This concept refers to the process by which an NLG system is evaluated to measure the quality of its outputs and/or its performance towards a desired task. This sub-tree, in turn, includes the choice of evaluator, the interactivity and internality of the evaluation approach, the metrics used to measure generated text quality, and the methodology that follows from these classes. A complete discussion of the NLG evaluation sub-tree is presented in Section 3.

AI Techniques: AI techniques refers to the form of machine-learning technique(s) or model(s) that is/are leveraged for NLG. There exist a wide range of AI techniques, including neural network (NN) based approaches such as RNNs and their derivatives (e.g., LSTMs and GRUs), and pre-trained Transformer-based models like GPT-2/3, XLNet, and CTRL. Further discussion of the AI techniques sub-tree is presented in Section 4.

Applications: The Applications concept encompasses the array of uses to which NLG systems can be applied. As mentioned earlier, NLG can be applied widely, finding uses in the creation of chatbots and other dialogue systems, creative generation (including story and poem creation), as well as being highly valuable in developing language translation and image-captioning systems. A dedicated examination of NLG applications will be provided in NL-2022-4.

Attacks: In terms of NLG, attacks can take two forms: attacks on the system and attacks by the system. The former thus refers to malicious attempts to cause undesired or unintended behaviours of a given NLG system or to otherwise avoid any detection systems they may have implemented. Attacks by the system, instead, refers to the usage of NLG systems themselves for malicious purposes, such as generating hate-speech or fake news. An in-depth look at NLG attacks will be provided in NL-2022-5.

Detection: Detection encapsulates approaches aimed at identifying AI-generated texts produced by an NLG system – a crucial task given fears of NLG outputs being disguised as human-created texts. Detection also includes approaches to identify potential attacks against NLG systems or those launched by an NLG system. More details on detection as it relates to NLG will be provided in NL-2022-5.

Challenges & Open Questions: Finally, this concept covers the future of NLG, examining the issues facing current and future NLG approaches and the open questions that need answering to fix these. Beyond issues of NLG-based attacks and detection, this encapsulates other problems with current NLG approaches, such as fears of model bias produced by poorly curated datasets, issues of privacy, and issues with current approaches to generating non-English texts [4]. Further discussion of these challenges and open questions will be included in NL-2022-5.

In this issue, we examine the key concepts of our taxonomy related to the development and evaluation of NLG systems via a review of survey papers dedicated to NLG. In Section 2, we discuss the commonly used methods by which NLG systems are developed. In Section 3, we discuss the key aspects in designing evaluation methods for NLG systems, highlighting the current inadequacies and limitations of existing approaches. Finally, we end with Section 4, in which we provide a broad discussion of the various existing AI techniques currently used for NLG purposes. Through this, we hope to provide a good overview of the necessary approaches, techniques, and evaluative procedures currently used for NLG. The following two issues of this newsletter will then build on this foundation to further explore NLG applications, detection, attacks, challenges and open questions.

2. NLG Methods

2.1. Introduction

NLG "aims to produce plausible and readable text in human language from input data" [36]. In turn, there exist a variety of popular NLG tasks (e.g., story generation, question answering) that can be adapted for a number of different applications (e.g., creative writing, conversation). This section contains a meta-review of a set of four survey articles [8, 36, 46, 67] and some supporting additional publications, and focuses on different aspects of NLG methods including the types of input, typical tasks, underlying techniques and training approaches used in the development of NLG models.

2.2. NLG Input

NLG tasks can be *uncontrolled* (also called unconditional) or *controlled* (also called conditional). In the former case, the text is generated without constraints, typically based on a random noise vector [36] or no input. For the latter case – i.e., controlled NLG tasks – a variety of inputs can be used to generate different types of text-based outputs according to the following generation paradigms:

Text-to-text: This paradigm includes topic-based and attribute-based NLG. It processes *unstructured* textual inputs to generate new output text(s). The types of input text include [36, 46, 67]: topics, keywords, sentiment labels, stylistic attributes (e.g., politeness, formality), demographic attributes of the "intended writer" (e.g., gender, age), information (e.g., event, entity), and text sequencing or ordering (e.g., from paragraphs, grounded documents, webpages).

Data-to-Text: This paradigm processes *structured* data input to generate new output text, retaining as much relevant information as possible from the structured data [36]. It includes non-linguistic data (knowledge-based or table-based data) and can take the form of knowledge graphs, expert system knowledge bases, database of records, spreadsheets, and simulations of physical systems [8, 36, 47]. Figure 1 shows an example of data input and its corresponding generated output.

Multimedia-to-Text: This paradigm uses *multimedia* data input (e.g., image, video, speech) to generate output text [36]. An example application is image captioning where the input is an image and the output is a corresponding text. This is further discussed in Section 2.3 under "vision-to-language".

2.3. NLG Tasks

There exist a wide range of common NLG tasks that can be adapted for a variety of different applications. This section provides an overview of typical NLG tasks, organised into nine classes of application [8, 36, 46, 67]. We will explore these applications in further depth in the next issue of this newsletter (NL-2022-04).

Conversation: This class of application is particularly relevant in the context of the Question Answering NLG task (e.g., generating responses to users' enquiries about a product), and of the Question Generation NLG task (e.g., given an answer, generating different questions with distinct focuses). It is also highly relevant to the creation of dialogue systems, including conversational agents and other chatbots [67]. The model typically takes as input some form of background information (e.g., product details and/or conversation history [8]) with the system aiming to generate a relevant utterance or response.

Summarisation: This class of application focuses on generating a concise version of a source text that includes its most relevant information. Beyond the source text, further input may be provided to increase the quality of the output text such as keywords, knowledge graphs, and "soft templates" [67] which contain some form of structure to guide the type of summary produced. From this, the NLG model thus aims to produce as output the new, shortened summary of the source text.

Machine Translation: This class of application encompasses NLG tasks aimed at accurately translating a given text from one language to another. An important aspect of machine translation is order preservation (i.e., alignment) of semantic units, like sentences, in the source and target text [36]. The input typically includes the source and target languages, and the text to be translated, while the output is the translated text.

Paraphrasing: This class of application aims to change the syntactic content of a source text while keeping its semantic value. This can be used in a variety of ways, including the plagiarising of a source text for academic purposes using tools such as "My

TEAM	WIN	LOSS	PTS	FG_PCT	RB	AST	...
Pacers	4	6	99	42	40	17	...
Celtics	5	4	105	44	47	22	...

PLAYER	H/V	AST	RB	PTS	FG	CITY	...
Jeff Teague	H	4	3	20	4	Indiana	...
Miles Turner	H	1	8	17	6	Indiana	...
Isaiah Thomas	V	5	0	23	4	Boston	...
Kelly Olynyk	V	4	6	16	6	Boston	...
Amir Johnson	V	3	9	14	4	Boston	...
...	...	...	...	...	...	...	...

PTS: points, FT_PCT: free throw percentage, RB: rebounds, AST: assists, H/V: home or visiting, FG: field goals, CITY: player team city.

The **Boston Celtics** defeated the host **Indiana Pacers 105-99** at Bankers Life Fieldhouse on Saturday. In a battle between two injury-riddled teams, the Celtics were able to prevail with a much needed road victory. The key was shooting and defense, as the **Celtics** outshot the **Pacers** from the field, from three-point range and from the free-throw line. Boston also held Indiana to **42 percent** from the field and **22 percent** from long distance. The Celtics also won the rebounding and assisting differentials, while tying the Pacers in turnovers. There were 10 ties and 10 lead changes, as this game went down to the final seconds. Boston (**5–4**) has had to deal with a gluttony of injuries, but they had the fortunate task of playing a team just as injured here. **Isaiah** Thomas led the team in scoring, totaling **23 points and five assists on 4–of–13** shooting. He got most of those points by going 14–of–15 from the free-throw line. **Kelly Olynyk** got a rare start and finished second on the team with his **16 points, six rebounds and four assists**.

Figure 1: An example of data-to-text generation from Puduppully et al. [47]; the left table shows data records provided as input, and the right part shows the corresponding generated text output.

Assignment Help" (https://myassignmenthelp.com/paraphrasing-tool.php), or simply to change or enhance a source text using tools such as "QuillBot" (https://quillbot.com/). Beyond the source text, topic information can be provided as input to help guide the paraphrasing model's output [36]. The desired output of a paraphrasing model is thus a rewritten version of the source text.

Creative Writing: This class of application concerns the generation of novel text(s) involving different NLG tasks, including story (or narrative) generation, scientific or report writing, artistic writing such as poems, and letter generation. The "Poem Generator" (https://www.poem-generator.org.uk/) is an example tool used for artistic writing. Inputs for creative writing applications are generally task dependent. Story generation might take as input information about persona and plot (i.e., sequence of events, topics, and desired ending) [46], or a knowledge graph capturing those [67]. Scientific/report writing typically takes as input a set of relevant source documents [46].

Vision-to-Language: This class of application aims to generate text based on a visual artefact such as an image or (visual content of a) video. Two NLG tasks relevant in this class are *Image Captioning*, where the goal is to generate a descriptive text of a given image [36], and *Visual Question Answering*, where the goal is to answer questions about a given image [69]. In both cases, the input is an image (or images) and the output is one or more sentences responding to the image. Image captioning can be constrained by different attributes to produce varying styles of caption, e.g., factual, romantic and humorous[19], as illustrated in Figure 2.

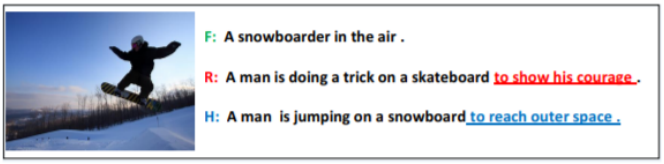


Figure 2: Image captioning task constrained by different styles [19].

Audio-to-Language: This class of application concerns the generation of text from audio artefacts such as the audio recording (or audio content) of a video. Relevant NLG tasks include *Speech Recognition* to process human voice recording into text format [36], and *Keyword Generation* to process audio and extract relevant keywords from it. The input is audio and the output is text or keywords.

Style Transfer: This class of application aims to re-write a text in such a way that it retains the content of a source text whilst changing a desired aspect of its style [46]. Common style attributes include sentiment (e.g., positive/negative), formality (e.g., formal/informal), and toxicity (e.g., offensive, non-offensive). The input is a source text and the source and target styles, and the output is a new text incorporating the target style [8].

Reasoning: This class aggregates tasks related to automatic reasoning that can be used to support other NLG applications, including Conversation and Creative Writing. NLG reasoning tasks include: *Commonsense Reasoning Generation* (or Generative Commonsense Reasoning) and *Argument Generation* [67]. The former aims to mimic humans' capacity to relate or generalise from a limited set of information into sentences that make syntactic and semantic sense [37], as illustrated in Figure 3. The input may be unstructured text-based input or struc-

tured data such as knowledge graphs. The latter task – Argument Generation – aims to produce valid and original arguments from a set of externally retrieved evidence such as a set of publications and/or Web resources [67].

Figure 3: Example of Commonsense Reasoning Generation from Lin et al. [37], where "Concept-Set" is the input and "Expected Output" is the goal of the task. The green box shows output generated by humans and the red box shows output generated by machines using different techniques elaborated in Section 4.

2.4. Training of NLG Models

The main goal of training an AI model is to reduce the gap between a desired output and a model selected output, where the gap is defined by a given objective function (otherwise known as a loss or cost function) [46]. In the context of NLG, objective functions are typically leveraged to generate more fluent, grammatical and diverse generations by attempting to find the optimum solution (i.e., next token in an NLG context) for the given objective function [46]. Whilst a number of loss functions are used in NLG, they typically centre around the comparison between a predicted token selected by the NLG model (given a sequence of tokens as input) and a reference token. By training the NLG model to minimise the loss between the generated output and the reference output, the model can 'learn' to output good quality texts.

As most NLG solutions currently rely on deep learning techniques, it is necessary to provide large amounts of text data to the NLG model in order to ensure it is adequately trained to perform a given NLG task to a desired level of capability. To train these deep learning models (commonly used techniques include RNNs and LSTMs) for NLG there exist a number of common training paradigms including: *supervised learning*, *reinforcement learning*, and *adversarial learning* [43].

Supervised Learning: The most common approaches to NLG utilise (pseudo) supervised learning via maximum likelihood estimation (MLE) [43]. In turn, large amounts of training text are provided to the model of choice as a series of training patterns (text fragments of a given size, sampled from the training data). The model is then tasked, for each training pattern, to predict the token that follows it [43]. This can, therefore, be thought of as a classification task, where each class is the equivalent of a unique token in the training dataset and the total of number of classes equals the total number of unique token (words, characters etc.) [43]. Via this training, the model is thus able to model the probability of each token occurring, given a sequence as input. MLE approaches, whilst popular, are limited by their tendency to overfit to the training data. This *exposure bias* means that MLE-based models are often limited in their ability to generalise beyond their training data [43], particularly as they must now generate tokens based on previously generated sequences, rather than sequences existing in the training set. This becomes particularly problematic as the length of the generated sequence increases [8].

Reinforcement Learning: To try and account for the limitations in MLE, reinforcement learning (RL) approaches have been suggested which aim to optimise non-differentiable metrics of text quality. A common approach to this is PG-BLEU. This learning method leverages the popular text evaluation metric BLEU (Bilingual Evaluation Understudy), which measures the n-gram overlap between a given (generated) text and a set of reference texts [7]. In turn, PG-BLEU aims to optimise for BLEU using typical RL policy gradient algorithms like REINFORCE [8]. Whilst, in theory, this could allow for the generation of more relevant texts than MLE-based approaches, the significant computational cost of computing BLEU so frequently means that this approach is seldom used in practice [43]. Additionally, criticism regarding the suitability of BLEU (and other, similar NLG evaluation metrics) to measure text quality raise further questions as to the applicability of these RL-based approaches [7]. As such, they

are less commonly seen in state-of-the-art work [43].

Adversarial Learning: Beyond the above, adversarial approaches to NLG have also been proposed. An early example in this space is the Professor Forcing algorithm, which uses adversarial domain adaption to reduce the distance between the training and generation of an RNN [8]. This, in turn, aims to limit exposure bias and boost generation quality. Beyond this, generative adversarial network (GAN) based approaches have also been suggested, using the discriminator's gradient to improve the generator's outputs [8]. A variety of GAN-based NLG approaches have been suggested, including seqGAN, maskGAN, and LeakGAN – these are described fully in Section 4. Whilst GAN-based approaches can be effective, they are inhibited by a number of problems. The most common of these are issues of vanishing gradient, in which the discriminator becomes much stronger than the generator leading to minimal updates being provided [43]. Issues of mode collapse are also common, in which the generator learns to sample from a small subset of tokens that receive higher evaluations from the discriminator [8]. This can lead to the GAN learning only a subset of the target distribution, limiting its ability to produce more generalised and diverse texts.

A problem with these typical core approaches to NLG is the necessity for large amounts of training data [36]. Without this, NLG models are prone to overfitting the training dataset and thus fail to generalise adequately to their desired task [36]. Moreover, even when datasets of sufficient size are available, the computational resources required to train these models are often prohibitive, restricting the feasibility of these solutions for many developers [36]. Additionally, these NLG models are largely task-specific in nature, only capable of generating text within the context of the training data provided. This means that developing NLG models to cover the large variety of NLG tasks and contexts requires a wide range of bespoke models, each trained on a significant amount of training data relevant to each task.

To help overcome this *data scarcity*, recent approaches instead leverage massive pre-trained language models [36, 58, 63]. Rather than training a given model to perform a specific NLG task, these language models, such as BERT and GPT-2/3 [14, 48], are instead pre-trained using a generic unsupervised text prediction task. Examples of these tasks include *Masked LM*, in which a series of sentences are presented to the model with a section of the sentence removed or 'masked' [14]. The model must then attempt to predict the missing part of the sentence. Other tasks include next sentence and next word prediction, in which a given input (e.g., a sentence) is provided to the model, and it is tasked with predicting the following word or sentence [48]. By training these models to successfully conduct these 'low-level' prediction tasks, the language model is able to achieve a good 'understanding' of how the language in the data provided is used.

Having conducted this pre-training, the model can then be fine-tuned using small amounts of task-specific training data to conduct a given NLG task – leveraging its understanding of language achieved at the pre-training stage and specifying it using the task-specific data. Beyond requiring smaller amounts of data to conduct a given NLG task, this also allows for high degrees of transference in which the pre-trained understanding of language achieved by the model can be adapted to a variety of NLG (and other NLP) tasks. When conducting fine-tuning of language models for NLG, a few approaches are typically used:

Few-shot learning: Few-shot learning relies on providing only a small number of samples during fine-tuning [36]. A subset of this is one-shot learning, in which a single sample is provided [36]. This approach therefore leans heavily on the generalised understanding of language achieved during the language model's pre-training. Applications of this in NLG include question answering, in which a few examples of similar questions are used to fine-tune a given model, before it is prompted to answer a new (unseen) question.

Zero-shot learning: Moving beyond few-shot learning, zero-shot learning asks a given NLG model to respond correctly to an unseen prompt with no additional fine-tuning data provided [6, 36]. This approach thus relies even more heavily on the model's ability to generalise from its learning during pre-training. Examples of this in NLG include the writing of news articles, using only the headline of the article as a prompt, or the answering of a question with only the question itself as a prompt. This approach (alongside few-shot learning) has only become feasible in recent years, with the rapid increase in the amounts of training data used to build powerful language models [6]. An example of zero-shot, one-shot,

and few-shot learning can be found in Fig. 4

Domain Transfer: Whilst few-shot (and zero-shot) learning may be desirable, in practice there is generally a sufficient degree of difference between the pre-training domain and the task domain that the model is unable to generalise effectively with minimal fine-tuning data [36]. In turn, most NLG approaches utilise domain transfer, in which large amounts of data are used to adapt the model to the desired NLG task. This may involve simply providing large amounts of fine-tuning data, or potentially providing additional data at the pre-training stage [36]. This approach can allow for greater NLG performances than can be achieved using few-shot learning, whilst often still requiring less training data than other deep learning approaches.

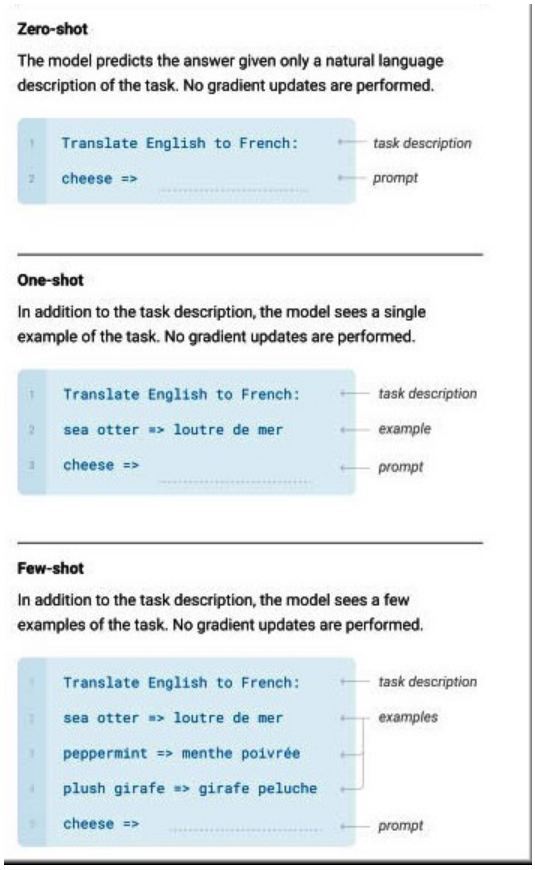


Figure 4: Examples of zero-shot, one-shot, and few-shot learning. Image source: https://medium.com/analytics-vidhya/openai-gpt-3-language-models-are-few-shot-learners-82531b3d3122.

Although the use of powerful language models pre-trained on massive datasets has allowed for state-of-the-art capabilities in NLG, this approach to training has brought with it a number of limitations.

Currently, most NLG models are notably English-centric, with many of the common pre-trained approaches specifically only utilising English datasets (e.g., GPT-2 [48] and XLNet [65]). Moreover, even when efforts are made to include additional languages, such as in GPT-3 [6], these additional languages are typically under-represented in the training data used [28]. This is particularly problematic in regard to the current reliance on web data for pre-trained language models. Whilst this approach is useful for extracting the vast amounts of data required for adequate pre-training, this approach yields clear biases towards a smaller number of over-represented languages [28].

In order to rectify this, approaches have been suggested to adapt current NLG methods to better support text generation in non-English languages [56]. Some approaches seek to leverage zero-shot learning, combined with large language agnostic datasets. Language models pre-trained on a variety of languages have thus been proposed, e.g., mBERT [56], mT5 [64], XLM [33], XLM-R [11]. These have, in turn, shown a reasonable degree of promise in adapting to text generation and other NLP tasks in different languages, either using their inherent language understanding achieved through pre-training or through minimal fine-tuning using texts written in the target language [56]. Whilst this approach has achieved a reasonable degree of success, these models are still limited by the degree of language variety within their pre-training datasets [28]. Additionally, issues of accidental translation in which the generated output contains the 'wrong' language are common [64]. Moreover, current research in this space has been limited by a focus on multilingual pre-training using genetically related languages. It is currently less clear as to the degree to which generalisation to non-related languages is possible [28].

Other approaches instead take a mono-lingual direction, in which the given language model is pre-trained entirely on text from the desired language (in the same way as English-only models are). This has proven to be the most effective approach, generally exceeding the capabilities of most multilingual models [4]. However, this method is limited in much the same way as English-only models are, being confined to generation in a single language. Moreover, as sufficient data for many languages is not currently available, reliance on this approach would thus ex-

Model	Dataset(s)	Size
BART	BooksCorpus, English Wikipedia, CommonCrawl (filtered), OpenWebText	160GB
BERT	BooksCorpus, English Wikipedia	16GB
CTRL	Wikipedia (En, De, Es, Fr), Project Gutenberg, OpenWebText, Amazon Reviews, and several other data sources	140GB
GPT	BooksCorpus	5GB
GPT-2	WebText	40GB
GPT-3	Common Crawl (filtered), WebText2, Books1, Books2, Wikipedia	570GB
MASS	WMT News Crawl (En, De, Fr, Ro)	41.5GB
T5	Colossal Clean Crawled Corpus (C4)	750GB
UniLM	BooksCorpus, English Wikipedia	16GB
XLNet	BooksCorpus, English Wikipedia, Giga5, ClueWeb 2012-B (filtered), Common Crawl (filtered)	126GB

Table 1: Pre-training datasets for language models popularly used in NLG.

clude the usage of NLG for many of the world's languages [4].

Moreover, issues of bias in the current (typically) web-based training data for NLG have been noted beyond the lack of multilingual support [4]. A major concern is the degree to which the often atypical and extreme nature of web content from certain platforms could have negative effects on the behaviour of NLG models trained using them [4]. This, in turn, could lead to NLG models that mistakenly internalise biases present in their (pre) training data, which could cause undesirable outputs hostile against certain protected groups, as well as causing potential predispositions towards violent or offensive language and hate-speech [4]. Whilst many of the filtering processes taken during the data collection phase aim to limit this, the vast amounts of training data currently used for natural language model-based NLG means that sufficiently curating datasets to limit these harms and biases is currently an open problem [4].

2.5. Training Datasets for NLG

As popular neural approaches to NLG require large amounts of high quality text data in order to be effective, there has been a large amount of effort dedicated to curating large datasets for NLG. This has become increasingly important with the rise of powerful pre-trained language models (mentioned above), which require sufficiently large datasets to generalise effectively. A table indicating the datasets used by several of the most popular pre-trained language models can be found in Table 1. In this section we detail some of the training datasets commonly used in NLG.

Common Crawl: The Common Crawl dataset is a massive collection of petabytes of web data currently hosted by Amazon (https://commoncrawl.org/the-data/). This dataset contains raw web page data, metadata extracts, and text extracts scraped over a period of 12 years. Due to its size, Common Crawl has been popularly used in the training and pre-training of many NLG models, including GPT-2/3 [6, 48], T5 [49], and XLNet [65]. However, as the dataset is so large and contains within it a great deal of low quality text [6], current NLG approaches conduct additional filtering to extract subsets of the data more suited to the NLG or general language modelling task at hand [6, 49].

WebText: Developed for use in the pre-training of GPT-2, WebText offers a subset of 8 million web pages from the Common Crawl dataset containing web pages that have been previously filtered or created by humans [48]. To do this, all outbound links from the web forum site Reddit with 3 or more karma were extracted. This ensured that the data included in WebText had received some degree of favourable response by humans and was likely of reasonable quality. Any Wikipedia articles included in this set of articles were then removed since they are commonly used in other NLG datasets.

OpenWebText: OpenWebText (https://skylion007.github.io/OpenWebTextCorpus/) is an open-source recreation of the WebText dataset. It contains 38GB of text data extracted from more than 8 million URLs shared on Reddit with at least

three upvotes.

Colossal Clean Crawled Corpus (C4): Another subset of the Common Crawl Corpus, C4 was created for use in pre-training the T5 language model [49]. To create C4, the authors filtered out all non-English texts, whilst also removing short texts, obscene texts, duplicates, and non-natural language text.

BooksCorpus: Created by [70], the BooksCorpus dataset contains a collection of 11,038 unpublished books scraped from the web. These books are all at least 20K words in length, and include a variety of genres including fantasy, romance, and science fiction.

ClueWeb12: Created by the Lemur Project, the ClueWeb12 dataset contains more than 7 million web pages scraped between 2010 and 2012 (https://lemurproject.org/clueweb12/).

Giga5: The English Gigaword Fifth Edition (Giga5) is a dataset of English newswire text data extracted from seven sources including the Los Angeles Times/Washington Post Newswire Service, the Washington Post/Bloomberg Newswire Service, and the Xinhua News Agency, English Service (https://catalog.ldc.upenn.edu/LDC2011T07). The data collection process was conducted over a number of years, beginning with the first edition in 2003 and ending with the fifth edition in 2010.

WMT News Crawl: WMT News Crawl (http://data.statmt.org/news-crawl/) dataset contains 1.5 billion lines of monolingual text from 59 languages, extracted from online newspapers. It was released for the Workshop on Statistical Machine Translation (WMT) series of shared tasks.

Wikipedia: Likely due to its size and availability (https://en.wikipedia.org/wiki/Wikipedia:Database_download), Wikipedia data has been commonly used in NLG [14]. In turn, many powerful models have leveraged subsets of the complete list of articles on Wikipedia, typically filtering it by the languages relevant to the desired language modelling/NLG task [14, 29]. This data is typically also filtered to only include the actual text content of the articles themselves, with non-prose based text such as tables, lists, and headers commonly being excluded [14].

2.6. Underlying Techniques of NLG Methods

Earlier approaches to NLG typically relied on techniques that leveraged rule-based or data-driven pipeline methods to generate text [7]. These approaches often utilised theoretical concepts of language, such as Rhetorical Structure Theory and Discourse Representation Theory, which allow for the modelling of discourse structures and syntactical relationships [7]. Template-based models were also commonly used, in which new texts are generated by 'filling in' the slots of a predetermined template with text-items selected by the NLG model [13]. Further approaches combined basic data-driven modelling with statistical techniques, e.g., by using Markov Chains to sequentially generate texts by predicting the next most probable word [44]. In general, however, these early techniques were typically limited in their NLG abilities, often struggling to model longer sequences and failing to adequately generalise from their initial training data and/or template.

With the rise of deep learning and several prominent methods for modelling sequences such as RNNs, their extension the LSTM, and Transformers, there has been a noted paradigm shift in NLG. In turn, current approaches typically leverage these deep learning approaches in an unsupervised manner, training them to learn sophisticated text representations from massive amounts of text data before fine-tuning them on a specific NLG task. Current trends in NLG have particularly focused on the use of Transformer models, such as GPT-2/3, pre-trained on very large datasets as a means of generating text in a wide variety of NLG tasks. We provide a more detailed overview of the common AI techniques used for NLG in Section 4.

3. Evaluation of NLG Methods

3.1. Introduction

Beyond the construction of a given NLG system, a further challenge in generating synthetic texts is presented by the manner in which these AI-generated texts can be successfully evaluated. Owing to the open-ended nature of many NLG tasks (reviewed in Section 2.3), the role of creativity in the text generation process, and the natural ambiguity of language, conducting successful evaluation is an ongoing challenge within the field of NLG [7].

Currently, there exist many different approaches to evaluating the outputs of a given text generator [7, 23]. In turn, we systematise these approaches using a series of different categories: the evaluator, the level of interactivity used, the internality of evaluation, the measures used for evaluation, and the overall methodological approach taken. Finally, we end by reviewing some of the existing standardised evaluation methods and tools that are currently in use, the best practices suggested in previous research, and the challenges still posed by a lack of standardisation in this area.

The content of this section is derived from a meta-review of 12 survey articles regarding the evaluation of NLG systems [1–3, 5, 7, 17, 20, 23, 26, 51, 59, 60].

3.2. Evaluators

Central to the evaluation of NLG systems is the role of the evaluator. That is, the agent responsible for evaluating the given NLG system. In turn, we identify two overarching types of evaluator that are used in NLG evaluation: **human-based** and automated **machine-based**.

The human evaluator refers to the use of human agents as the judges of a given NLG system [7]. Human evaluators are typically called upon to perform one of three evaluation tasks: (1) they may be asked to rate a sample of generated outputs from a given NLG model in a stand-alone fashion [23], (2) they may be tasked with comparing or ranking outputs from a series of NLG models [7], or (3) they may be tasked with conducting some form of modified Turing test, using their abilities to distinguish between generated texts and human-created texts [20].

Additionally, human evaluators can then be subdived into *expert* and *non-expert* evaluators [59]. Typically, NLG evaluations are conducted using either a small number of expert evaluators or a larger number of non-expert evaluators, though the precise numbers commonly used vary significantly from study to study – with typically 1-4 expert evaluators used and anywhere from 10-60 non-expert evaluators being common [59]. Whilst most approaches utilise either expert or non-expert evaluators, there is the potential for the usage of both to be of value as research indicates that different insights can be gleaned from human evaluators with varying levels of expertise [59].

Humans evaluators are commonly used and generally considered the 'gold-standard' for NLG evaluation. This is due to their superior capabilities of language comprehension and strong abilities towards context-based evaluation (relative to automated evaluators), as well as the added depths of insight that they can provide [7, 26, 51].

However, it is also worth noting that there are limitations in the current usage of human evaluators. Firstly, human evaluation can be time-consuming and inefficient to conduct [2, 26]. Moreover, the use of human evaluators requires some form of recruitment process which may be prohibitively expensive for some NLG projects [7]. Additionally, there can be issues of consistency when using human evaluators. As human evaluators must typically rely on relatively, if not entirely, subjective judgements, criticisms of the consistency and replicability of projects evaluated using human evaluators are also common [7, 51].

Parallel to the usage of human evaluators is automated evaluation. In this approach, some form of automated methods are utilised to provide evaluation of the NLG system [17]. Automated evaluation, in turn, can then be categorised into two further sub-domains: *untrained automatic evaluation* and *machine-learned evaluation* [7].

Untrained automatic evaluators, the form of automated evaluator most commonly utilised in NLG studies, rely on the use of one, or a series of, objective metrics to evaluate a given NLG system [17]. These metrics require no pre-training and can be used to provide efficient evaluations of large amounts of generated data.

However, there are issues regarding the abilities of these automatic approaches to successfully repli-

cate human judgement – with low correlations often being found between these metrics and human evaluations [17]. Moreover, automatic approaches are often limited in their ability to provide more holistic measures of automated text quality, which can be particularly limiting in scenarios in which a given NLG system is applied to more creative tasks such as story generation [7].

A more recent approach to automated evaluation concerns the use of machine-learned evaluators [7]. Rather than leveraging predetermined measures of evaluating NLG systems, this approach attempts to train machine learning models to act as pseudo-human evalautors. In turn, this aims to account for the limitations of automatic evalautors in capturing holistic qualities of generated texts and in correlating with human judgement. Currently, however, there has been less research into the creation of these evaluators and they remain relatively underused in NLG studies [7].

3.3. Interactivity

Interactivity refers to the the manner in which the evaluator engages with the generated artefacts to evaluate. In turn, this leads to two forms of interactivity: **static**, and **interactive** [17].

In static evaluation, the evaluator is simply presented with samples of the generated output of a given NLG model [17]. The evaluator can then proceed to offer judgements on the sample provided. This approach is by far the most common in NLG evaluation and is task agnostic – meaning that it can be conducted with any form of NLG system directed towards any of the numerous NLG tasks possible – and can be used for both human and automated evaluation.

Whilst static evaluation has the advantage of being broadly applicable (often being the only applicable approach to NLG evaluation) it can be limited when evaluating dynamic NLG systems. This is particularly true in the case of chatbots and other dialogue systems, where the offline evaluation of their text outputs in isolation can often lack relevance in terms of evaluating the true capabilities and behaviours of the dynamic NLG model [17].

In turn, interactive evaluation is offered as a solution to this. Interactive evaluation is conducted via direct interactions with the NLG system, as if the evaluator was a user [17]. The evaluator is thus able to directly interact with the NLG system and provide their evaluations in response to the NLG's abilities within its desired role [17].

Although interactive evaluation can provide greater value in understanding the capabilities of a dynamic NLG system it is also more complex to implement, typically requiring more time and evaluator effort than static evaluation. Moreover, the added complexity of interactive evaluation means that the use of automated evaluators is considerably more difficult to implement. This, in turn, typically necessitates the use of human evaluators [17].

3.4. Internality

Internality refers to the area that the evaluator should emphasise when making judgements of an NLG system. Specifically, internality can take one of two distinct attributes: **intrinsic** and **extrinsic** [3, 7, 59].

With an intrinsic approach to NLG evaluation, the goal is to evaluate the proposed NLG system via a direct assessment of its generated outputs [7]. With extrinsic evaluation, the aim is instead to assess the NLG system based on how successfully it achieves an intended goal or downstream task [7]. For instance, an extrinsic evaluation of an NLG system designed to generate advertisements for cars may be conducted by examining how well these adverts increase car sales, whereas an intrinsic evaluation would focus on assessing the content of the advertisements themselves.

In general, intrinsic methods are most commonly used in NLG studies [7] – with one review identifying only 3 (3%) papers that utilise extrinsic evaluation [59]. Due to its focus on evaluating the NLG system in a downstream process, extrinsic evaluation often requires a more long-term and costly evaluation process [7]. Additionally, current NLG research is typically focused on smaller sub-tasks lacking the clear real-world application required to perform extrinsic evaluation [59]. However, it is also typically considered a more relevant and tangible form of evaluation, and as NLG systems become more integrated into real-world applications an increase in this form of evaluation is likely.

3.5. Evaluation Measures

Emerging from the choices of the evaluator and its level of interactivity and internality are the measures that are to be used to score the 'quality' of

Original Criterion	Mapped to normalised criteria	Count
fluency	fluency; goodness of outputs in their own right; goodness of outputs in their own right (form); goodness of outputs in their own right (both form and content; grammaticality; humanlikeness); readability; [*multiple (3)*: goodness of outputs in their own right (both form and content), grammaticality, naturalness (form)]; [*multiple (2)*: goodness of outputs in their own right (form), grammaticality]; [*multiple (3)*: fluency, grammaticality]; [*multiple (2)*: grammaticality, readability]; [*multiple (2)*: fluency, readability]; [*multiple (3)*: goodness of outputs in their own right (both form and content), grammaticality, naturalness (form)]; [*multiple (3)*: coherence, humanlikeness, quality of outputs]; [*multiple (2)*: goodness of outputs in their own right (both form and content), grammaticality]	15
readability	fluency; goodness of outputs in their own right; goodness of outputs in their own right (both form and content); quality of outputs; usefulness for task/information need; readability; [*multiple (2)*: coherence, fluency]; [*multiple (2)*: fluency, readability]; [*multiple (2)*: readability, understandability]; [*multiple (3)*: clarity, correctness of outputs in their own right (form), goodness of outputs in their own right]	10

Figure 5: Example of the original quality criterion names used in the literature and the variety of normalised criteria these names are actually referring to [26]. This highlights the lack of common definitions in the current usage of quality criteria.

a given NLG system or model. These measures, in turn, are generally divided by the evaluator used into measures for human evaluator-based evaluation and measures for machine-based evaluation [7]. We present the most common measures used in NLG evaluation for each of these.

Human-based evaluation measures are typically based on examining the direct scores provided by a series of human evaluators on a set of generated texts [26]. In turn, this can be divided into two parts: the *quality criterion* and the *evaluation mode* [3, 26]. These two parts combine to allow a human evaluator to provide their assessment of the relevant aspects of an NLG system's quality.

Quality criterion describes the aspect of the NLG system's outputs that the human evaluator is attempting to measure [3, 23, 26]. Human-based evaluation methods will often measure multiple quality criteria, where each criterion relates to a desirable component that should appear in a quality AI-generated output.

There are a large range of quality criteria in current usage that can be sub-divided taxonomically in a variety of ways. In [26], the authors identify the high level quality concepts of *measures of correctness* and *measures of goodness* that define the current quality criterion used in NLG studies. Further sub-classes then ask the evaluator to consider the quality of the text either in its own right, relative to a reference external to the NLG system, or relative to the inputs provided to the NLG system [26].

A key limitation in current studies is that the quality criteria measured are essentially innumerable and often very different from one study to the next [3, 26]. A lack of a common vocabulary for these quality criteria is also a key limiting factor in replicability and comparison, with papers frequently using the same quality criteria name but with varying definitions (as shown in Fig. 5) [3, 5, 23, 26, 51, 59].

Some of the most commonly used quality criterion are:

Fluency: This refers to the degree to which a generated text, or set of generated texts, mimics the intended language it is written in [23]. A broad criterion, this can include considerations of correct grammar and syntax, spelling, and style. Whilst most commonly used in machine translation, fluency can be easily applied to any NLG task in which fluent writing is desired. This has lead to it becoming one of the most commonly used dimensions in existing studies [59]. Additional aspects of fluency, including tone and formality, are also of importance to style transfer tasks [5]. Due to its broad nature, however, a lack of clear definition for fluency within the NLG literature is problematic [3].

Usefulness: Usefulness is a criterion focused around the degree to which the generated text is valuable for a given task or information need [26].

Factuality: Factuality examines the degree to which the generated text is logically coherent, and the degree to which its statements are true [7]. The first aspect of factuality is broadly useful for NLG evaluation, whilst the second aspect is of particular value to tasks such as news generation, where accu-

rate reporting in the generated text is desired.

Naturalness/Typicality: Naturalness (also called typicality) asks the evaluator to assess how 'typical' a given generated text is, or how often they'd expect to see a text like this [23, 59]. This is usually considered in terms of how likely a natural speaker would produce the given text, and can be measured in terms of both the content and form of the output [26].

Grammaticality: This criterion asks the evaluator to measure the extent to which the generated text is free of grammatical errors [26].

Evaluation mode, in turn, describes the approach by which the human evaluator provides a measure of how successful the NLG system has been in capturing a given quality criterion (or criteria) [3].

Evaluation modes also have their share of limitations, often hindered by subjective criteria that makes interpretation and comparison of scores recorded by multiple evaluators difficult [23]. This is particularly problematic when comparing studies [2]. Additionally, there is also a clear lack of consensus as to the most effective evaluation modes to use in human evaluation of NLG systems, with a wide variety of scoring techniques commonly being used in the literature [5, 59].

Common evaluation modes include:

Preference: This evaluation mode involves the evaluator selecting their preferred text, or texts, from a set of texts [59]. These texts are typically a collection of generated outputs from a series of models or a combination of AI-generated and human-created texts [59].

Numerical Scale: One of the most commonly used evaluation modes, numerical scales ask the evaluator to rate the quality of a set of generated text(s) on a sliding scale (e.g., from 1 – 5 as shown in Figure 6) [2]. This allows for a more fine-grained measurement of the quality of the generated outputs compared to binary scoring [7].

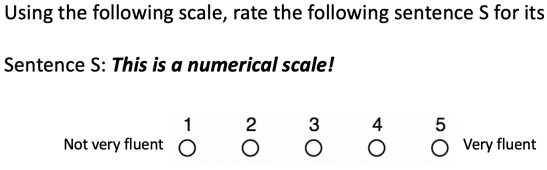


Figure 6: Example of a numerical scale.

Graphical Scale: Similar to numerical scales, graphical scales utilise words or phrases (as opposed to numbers) for each rating value (See Figure 7 for an example) [2].

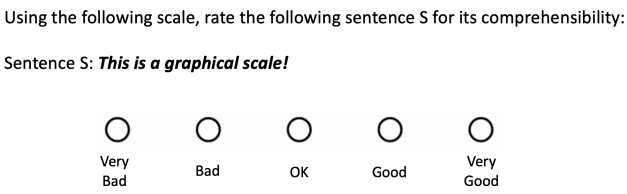


Figure 7: Example of a graphical scale.

Likert Scale: Likert Scales are an aggregate scale comprised of multiple graphical scales called Likert Items [2] (see Figure 8 for an example). This allows for a survey-style approach that record overall evaluator impressions based on responses to multiple dimensions of a set of generated texts. Due to the lack of consistent intervals between scale items, however, many researchers state Likert Scale results must only be considered in aggregate, though there is much disagreement regarding this [2]. This confusion makes evaluator–evaluator and study–study comparisons using Likert Scale evaluation particularly difficult (though all of the evaluation modes presented here suffer from this to some extent). Despite this, Likert Scales remain popular in NLG evaluation [2].

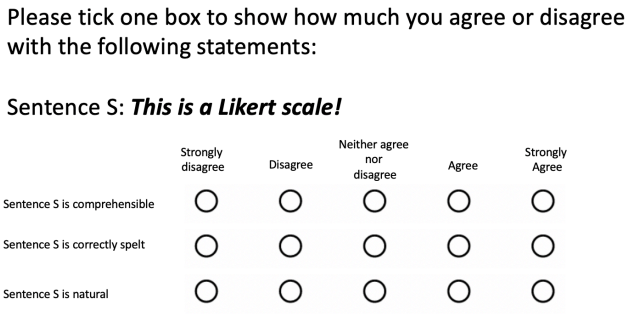


Figure 8: Example of a Likert scale.

Ranking: To overcome the limitations of the above approaches, ranking has been used as a means of human-based evaluation. Rather than scoring the generated texts, the evaluator instead ranks the generated texts according to their quality [23]. This has the advantage of allowing for better inter-evaluator comparison, but only provides a relative appreciation of the NLG model or its outputs, rather than a true measure of its output quality [7]. Ranking can also be limited by its complexity, with large numbers of comparisons becoming prohibitively complex.

Extending the use of direct measures of quality, some studies leverage existing inter-annotator agreement metrics to better account for the level of unity in the scores provided by a set of human evaluators [7]. These metrics measure the amount of agreement between evaluators and provide indications of the extent to which evaluators concur with the suitability or quality of the NLG model they are evaluating [1]. It is worth noting that the use of inter-annotator metrics in NLG evaluation is generally less common than the sole use of individual metrics (appearing in only 12.5% of papers examined in [59]), and that even when they are used, the agreement scores are typically lower than what would be considered 'acceptable' [1, 5, 7, 23, 59].

Furthermore, there are questions regarding the suitability of inter-annotator agreement as a sole metric for evaluating consistency in NLG evaluations [1]. Due to the ambiguous and varied nature of language, there are questions regarding whether a metric focusing on strict agreement – as is the case with inter-annotator agreement metrics – is suitable given the potential for different (but valid) interpretations of the same texts [1].

The most commonly used inter-annotator metrics are:

Percent Agreement: Percent agreement is the most straightforward means of measuring agreement between two independent evaluators. It simply reports the percent of cases in which the two evaluators agreed with each other [7]. Whilst popularly used in NLG evaluation, percent agreement fails to account for the possibility that agreement between evaluators may occur by chance [1]. This is particularly problematic when utilising individual metrics with fewer scoring options such as binary scoring. Percent Agreement is given as

$$P_a = \frac{\sum_{i=0}^{|X|} a_i}{|X|}$$

where X is a set of generated texts for which evaluators assign a score to each text x_i, and a_i is the characteristic function denoting agreement in the scores for x_i. Hence, $a_i = 1$ if the evaluators assign the same score and $a_i = 0$ if not.

Cohen's κ: Improving on percent agreement, Cohen's κ is able to account for the possibility of agreement occurring by chance in the annotations of two evaluators [10]. To achieve this, Cohen's κ defines the probability of two evaluators, e_1 and e_2, agreeing by chance as

$$P_c = \sum_{s \in S} P(s|e_1) * P(s|e_2)$$

where S is the set of all possible scores for texts in X. The conditional probabilities $P(s|e_i)$ are estimated via the frequency with which the given evaluator assigned each of the possible scores in S. Combining percent agreement P_a with the probability of agreement by chance P_c, Cohen's κ is defined as

$$\kappa = \frac{P_a - P_c}{1 - P_c}.$$

Fleiss' κ: Fleiss' κ improves on Cohen's κ by measuring the agreement of more than a single pair of evaluators by considering all pairwise inter-annotator agreements [18]. To this, a_i, the agreement of scores for two evaluators for a given generated text, is redefined as

$$a_i = \frac{\sum_{s \in S} \#\text{ pairs scoring } x_i \text{ as } s}{\#\text{ evaluator pairs}}.$$

The probability of agreement by chance, P_c is also redefined by estimating the probability of a given score by the frequency of that score across all evaluators. This is defined as

$$P_c = \sum_{s \in S} r_s^2$$

where r_s is the proportion of evaluators that assigned a given score s. Fleiss' κ is thus defined using the same definition of Cohen's κ, combining the definition of P_a used for percent agreement alongside the redefined P_c.

Krippendorff's α: Reevaluating the approaches above to consider the likelihood of disagreement, Krippendorff's α, as with Fleiss' κ, can be used to evaluate multiple annotators whilst accounting for agreements that occurred by chance [32]. Moving beyond Fleiss' κ, however, Krippendorff's α is also capable of handling missing values, where Fleiss' κ and Cohen's κ cannot [32]. To define Krippendorff's α, we first find the probability of disagreement using

$$P_d = \sum_{m=0}^{|S|} \sum_{n=0}^{|S|} w_{m,n} \sum_{i=0}^{|X|} \frac{\#\text{ pairs scoring } x_i\text{: } (s_m, s_n)}{\#\text{ of evaluator pairs}}$$

where (s_m, s_n) indicates one possible score pair, and $w_{m,n}$ the weight used to adjust the degree of penalisation for a given disagreement. The probability

of agreement by chance is also redefined, using $r_{m,n}$ to represent the proportion of all evaluation pairs that assign the scores s_m and s_n. Given this, the probability of agreement by chance is defined as

$$P_c = \sum_{m=0}^{|S|} \sum_{n=0}^{|S|} w_{m,n} r_{m,n}.$$

Given the probability of disagreement P_d, and the redefined probability of agreement by chance P_c, Krippendorff' s α is calculated using

$$\alpha = 1 - \frac{P_d}{P_c}.$$

Whilst a range of automated evaluation measures and metrics exist, including more sophisticated approaches built around trained machine learning models, most automated evaluation of NLG leverage untrained automated evaluation metrics [7]. These automated metrics offer an objective, easy-to-implement method for measuring the quality of generated texts [17], centred around the comparison of a set of generated texts to a gold-standard set of (generally human-created) reference texts [7]. The assumption is that the closer the generated texts are to the reference texts, the better. Whilst a large number of automated evaluation measures exist, the most commonly used in NLG studies are n-gram overlap metrics [7, 17, 60].

N-gram overlap metrics are designed to measure the degree of similarity in the n-grams present in the generated texts when compared to a set of reference texts [20]. These approaches typically leverage word-based n-grams, though other n-gram approaches (e.g., character n-grams) can be used. The assumption is that the larger the overlap in n-grams between the generated text and the reference texts, the higher the quality of the generated text.

Some of the most commonly used n-gram overlap metrics are:

BLEU: The Bilingual Evaluation Understudy (BLEU), is one of the oldest and most commonly used n-gram overlap metrics [45]. Originally intended for evaluating machine translation tasks, BLEU has seen further use in other generation tasks including style transfer, story generation, and question generation [7]. BLEU works by comparing the overlap in the n-grams of a candidate (generated) text and the n-grams of a set of reference texts using the weighted geometric mean of modified n-gram precision scores [45]. N-gram precision scores are calculated by measuring the fraction of n-grams appearing in the generated text that appear in any of the reference texts. An example of BLEU can be found in Fig. 9. In this figure, note that candidate 2 is ranked lower than 3, despite more closely matching the meaning of the reference. This highlights a key limitation of n-gram based metrics, which can overemphasise surface level lexical similarities.

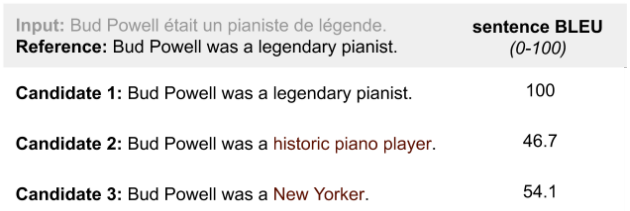


Figure 9: Example of BLEU for three generated candidates. Image source: [54]

ROUGE: The Recall-Oriented Understudy for Gisted Evaluation (ROUGE) works in much the same way as BLEU, but focuses on measuring recall rather than precision [38]. In other words, ROUGE measures the fraction of n-grams in the references texts that appear in the generated candidate text. ROUGE itself is a broad class that describes a set of variants. Most commonly used are the ROUGE-N variants, where N is the size of n-gram to be evaluated (e.g., ROUGE-1 evaluates unigram overlaps) [7]. Another common variant is ROUGE-L, which evaluates the longest sequence of shared tokens in both the generated and the reference texts [38]. ROUGE is generally considered to yield more interpretable scores than BLEU [7].

METEOR: The Metric for Evaluation of Translation with Explicit ORdering (METEOR) attempts to improve on some of BLEU's weaknesses by utilising the weighted F-score using unigrams, where recall is weighted more heavily than precision as this has been found to yield higher correlations with human judgement [34]. Moreover, METEOR also incorporates a penalty function that penalises incorrect unigram order [34].

CIDEr: The Consensus-based Image Description Evaluation (CIDEr) utilises a consensus-based protocol to measure the similarity of a generated sentence to that of a set of human-created reference sentences using TF-IDF weighted n-gram frequencies [62]. Originally intended for evaluating generated image captions, CIDEr has also been used in

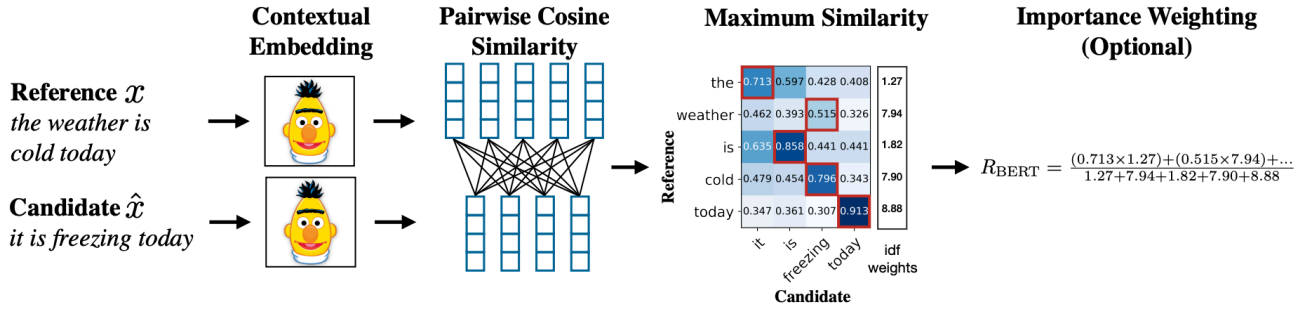


Figure 10: Diagram of BERTScore. Image source: [68]

the evaluation of other NLG tasks including online review generation [20].

Whilst being some of the most commonly used metrics in NLG evaluation, n-gram overlap metrics are often criticised for their lack of correlation with human judgement [7, 59, 60]. Additionally, these metrics fail to account for more holistic qualities in generated texts including fluency and grammatical correctness [60]. Moreover, the assumption that a good text is equivalent to a text that closely mirrors the reference set is potentially faulty, as unexpected texts could still be of sufficient quality [60]. This leads to additional difficulties in justifying the relevancy of the reference set and, in turn, the meaningfulness of the metric scores achieved [59]. Due to the high degree of criticism levelled at untrained automated NLG metrics, more recent proposals have been made to leverage machine-trained evaluators to measure an NLG system's quality [7].

Whilst these metrics leverage more sophisticated machine learning models, they are typically still used to examine similarities between the generated samples and a series of reference texts [7]. The use of the machine-trained evaluator is thus a means of conducting more sophisticated comparisons that leverage more latent, and more relevant, aspects of text (e.g., semantics and syntax) as opposed to the more surface-level comparisons of the untrained metrics above.

A common approach is to leverage machine-trained models for measuring the semantic similarity between generated and reference texts via the use of learned word and sentence embeddings [7]. Various attempts have been made towards this, including more traditional embedding approaches such as skip-thought [31], fastsent [25], quick-thought [41]; and newer approaches that leverage pre-trained language models and contextual embeddings like BERTScore (Fig. 10) [68]. In essence, these approaches examine similarities in syntax and semantics between generated texts and gold-standard reference texts via examining the embedding distances between the two using some form of distance, e.g., cosine similarity [7]. The assumption being that higher quality generated texts will more closely match the reference texts in terms of both semantics and syntax.

Beyond this, further proposals have been made to leverage trained models to perform regression-style evaluation of NLG translation systems, such as the GRU (gated recurrent unit) based RUSE [7], which are used to predict a scalar value indicative of the quality of a generated (translated) text relative to a reference text. Semi-supervised methods, like ADEM [42] and HUSE [24], have also been proposed as a means of leveraging human judgements in the decision making of the machine learning model [7].

In general, the research conducted towards developing machine-learned metrics are promising, with high correlations often being found between their scores and human judgement. Despite this, however, their adoption is still relatively low in NLG evaluation [7].

3.6. Evaluation Methods

As with the majority of the items above, the methodological process of evaluating NLG systems is typically categorised by the type of evaluator used: i.e., human-based methods and machine-based methods [7].

As discussed above, human judgement is typically considered the most effective form of NLG evaluation [7]. Due to the innate ability of humans towards language and their clearer appreciation of

the role of context and semantics, human evaluator methods can be highly effective.

With human approaches to NLG evaluation, the most common methods utilise intrinsic, static forms of evaluation [7, 17]. Through this approach a set of human evaluators are generally presented with a series of generated texts and will be tasked with providing judgements on the quality of these texts based on their ability to maximise a set of quality criteria, via a prescribed scoring mode [3]. The scores recorded by each human evaluator are then generally analysed in some manner, typically via basic statistical measures, to gain an overall appreciation of the capabilities of the NLG system in question [23].

Interactive intrinsic human evaluation methods are also possible, and are commonly advocated for when evaluating dialogue systems [17]. These approaches again utilise the basic dimension-scoring measures, but allow the human evaluator to examine the artificial texts as they are generated in real-time [17]. This approach also allows the human evaluator to directly prompt the NLG system, allowing for the evaluator to gain a better sense of the quality and relevance of the texts being generated.

Additionally, some approaches have utilised human evaluation methods with a focus on interactive extrinsic evaluation [7]. These methods of human-based extrinsic evaluation thus measure how effective the NLG system is at allowing the user to succeed at a given task. One early example of this utilised an instruction generation system, where human evaluators were required to follow the generated instructions [7]. Evaluations were then made based on the success of the evaluators in achieving the desired tasks by following the instructions.

Moreover, extrinsic interactive human evaluation is more commonly used in evaluating dialogue systems and chatbots [7]. These methods typically utilise some kind of feedback form or dimension-based scoring measure and evaluate the ability of the dialogue system to meet user needs over longer periods of time. This is distinct from intrinsic interactive evaluation, as the focus is on the satisfaction of user requirements rather than the direct quality of the dynamically generated text [17].

There are, however, distinct limitations and inconsistencies in the current usage of human evaluation methods of NLG, the core issue being that there is little agreement or standardisation in the manner in which human evaluation should be conducted [3, 23, 26, 51].

In the literature at large, there is a wide range of evaluator numbers used, with some studies leveraging as few as two, whilst others use 500+ (typically through crowd-sourcing) [5, 20, 23]. Additionally, some studies rely on expert evaluators, whilst others recruit non-experts – typically with little justification for this decision [59]. Moreover, many studies do not report the number of evaluators used [5, 51, 59].

There is also little consensus on how many generated samples should be evaluated, with as few as two up to more than 5,400 [5, 23, 59]. Moreover, some studies provide the same set of samples to all evaluators, whilst others provide different subsets to each evaluator [59]. It is currently unclear as to how these variations between studies may effect the quality of the evaluations performed.

There are also questions regarding the manner in which evaluators are selected [5, 23]. Some research has published concerns as to the role that selection bias may play in evaluation, particularly with the use of crowd-sourced evaluators where demographic information is hard to produce. This, coupled with the typically low number of evaluators used, could cause further unconsidered effects on the evaluators given [23].

Finally, there are significant inconsistencies in the reporting of human evaluation methods [3, 5, 26, 51, 59]. In turn, it is not uncommon for NLG papers to not include details of the number of evaluators used, the questions posed to evaluators, or even the manner of scoring or quality criterion that the evaluators used [26, 51, 59]. An example of this is provided by Howcroft et al. [26] (as shown in Fig. 11), in which they note that over half the papers studied did not define the quality criterion used in their evaluations. These inconsistencies in reporting within nearly every aspect of the human evaluation method further emphasise the problems above, compounding the difficulties of replicability and comparison between NLG studies [26, 51].

Whilst machine-based evaluations can, in theory, be conducted using both static and interactive approaches and intrinsic or extrinsic approaches, in general automated evaluation has focused on the use of machine-based evaluation as applied to intrinsic, static evaluation [7, 17, 20].

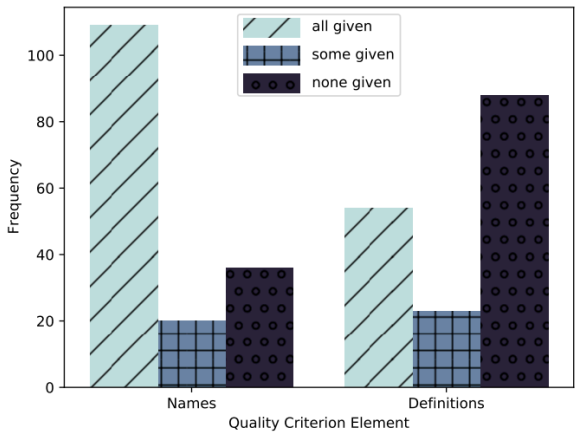


Figure 11: The number of papers explicitly naming and defining the quality criterion used in their human evaluation, as identified by Howcroft et al. [26].

Generally speaking, machine-based evaluation is centred around the leveraging of one or more of the untrained automated evaluation metrics discussed in Section 3.5 [7]. Often, multiple metrics will be reported as a means of attempting to balance the weaknesses of each individual metric [17].

The popularity of this approach is likely due to the efficiency of implementation and the ease with which it can be scaled to evaluate large numbers of generated texts/systems. However, many criticisms have been levied towards the use of these automated metrics, arguing that they offer poor indications of the genuine quality of NLG systems with typically low correlations with human judgement [17]. Additionally, issues of reporting are equally prevalent, with studies often neglecting to include relevant information such as the the number of generated samples evaluated and the sampling method used to select these generated samples.

Moreover, whilst the usage of machine-trained evaluation approaches are becoming more studied, they are still relatively uncommon [7]. This is likely to be an area of distinct progress in future, due to the unreliable nature of existing, untrained approaches to automated NLG evaluation.

To help compensate for these weaknesses, it is common for studies to utilise human evaluators to provide greater insights and confidence to the performance of a given NLG system, whilst also reporting automated metrics to better aid with replicability and comparison with the state-of-the-art [7].

However, limitations regarding the lack of reporting in general, replicability in regard to human evaluators, and lack of correlation with human judgement in regard to automated metrics, means these combined approaches are still vulnerable to some of the key weaknesses inhibiting current NLG evaluation methods [7, 17, 23].

3.7. Standards

The evaluation of NLG systems is currently hampered by the distinct lack of standardised approaches and generalisable methodologies [3, 26, 59, 60]. Instead, NLG papers – even those conducting similar tasks – often take highly contrasting approaches to their evaluation [7]. Even with more popular approaches, such as the use of human evaluators using a quality criterion-evaluation mode method, the exact specifications of these approaches can vary significantly [3, 59]. This includes variations in definitions of quality criteria, scoring methods used, and the construction of the evaluation methodology as a whole [3, 23]. Moreover, a lack of adequate recording of the evaluation approaches taken is additionally problematic, often making it hard to fully understand the evaluative process of a given NLG study and making it highly difficult to replicate and/or compare studies [59].

Given this, one could be forgiven for assuming no such standardised NLG evaluative approaches exist, but this is not the case. In [7], the authors detail several examples of existing platforms developed to standardise NLG evaluation. These include GENIE [30] (an example of the GENIE architecture can be found in Fig. 12) and GEM [21]; two evaluation platforms that have been proposed as a means of bench-marking NLG systems using both automated and human evaluation across multiple datasets and NLG tasks, including text summarisation, text simplification, and dialogue generation [7]. Moreover, other task-specific evaluation platforms also exist, including ChatEval which provides a web-interface for standardised evaluation and comparison of NLG chatbots and dialogue systems to the state-of-the-art results, leveraging both automated metrics and human judgement [53]. Despite the existence of these platforms, however, adoption has been very low across NLG research.

To help solve the problems posed by a lack of consistency and generalisability, many of the review papers included in this study propose best practices that should be followed when conducting NLG evaluation.

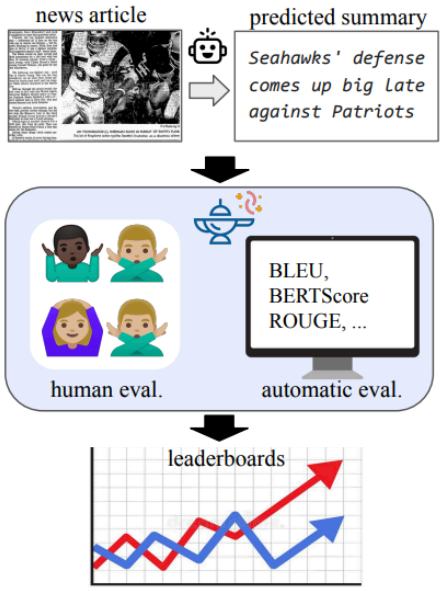


Figure 12: The GENIE architecture [30] applied to a text summarisation task. GENIE combines both human and automated evaluation techniques.

In [51], the authors discuss the need for clearer and more consistent reporting of NLG human evaluation methodologies. To this end, they list a set of important design parameters that should be detailed when reporting. These aspects focus on: including discussion of question design and presentation, including wording and scoring approaches; detailed discussion of the quality criteria used, including the clear naming of each criterion and a description of how these criteria are defined in the work; and, an in-depth description of any human evaluators used, including details of their expertise, demographics, recruitment, and compensation. The authors also note that justifications of these decisions should be included where relevant and be rooted in previous literature where possible to improve consistency and aid in cross-study comparison.

Belz et al. [3], van der Lee et al. [59], and Howcroft et al. [26] provide similar guidance as to the best practices in reporting human NLG evaluation. The authors, in turn, highlight similar needs for details regarding quality criteria, the evaluators used, and the questions posed. They also highlight the need for additional details not mentioned by Schoch et al. [51], such as a clear discussion of the evaluation design, including the number of evaluators used, the number of samples provided to each evaluator, the information given to evaluators (e.g., training, instructions, interface), and the method for sampling the generated outputs provided to evaluators.

In summary, there exist a range of papers aimed at developing generalisable approaches and best practices to NLG evaluation. However, adoption of these appear low, with little indication of a trend towards more standardised approaches to NLG evaluation. This, in turn, remains an ongoing problem that needs addressing to ensure adequate NLG evaluation, and to allow for clear comparison between different approaches to various NLG tasks.

4. Relevant AI Techniques for NLG

4.1. Introduction

Recent advancements in AI, particularly in the field of deep learning, have led to the development of more powerful language models that can better 'understand' natural languages. This, in turn, has led to increased capabilities towards the generation of more realistic and convincing natural language text. This section sheds light on some of the commonly used AI techniques for NLG, including neural networks, Transformers, and combined techniques which make use of multiple methods.

4.2. Neural Networks (NN)

NNs have been extensively used in NLG, and provide a strong foundation toward realistic and meaningful NLG. In spite of the paradigm shift toward Transformer-based language modelling in recent years [36], NNs are still used in many NLG applications. The most commonly used NNs are as follows:

Recurrent Neural Network (RNN): RNNs are designed to model sequential information – of which text is a common example – and as such have been commonly used in NLG. Early encoder-decoder frameworks for NLG were generally based on RNNs [67]. In addition, RNNs have been used in *controllable NLG*, which aims to generate text with controllable attributes, such as sentiment, formality and politeness [46]. Apart from being utilised as a generation method, RNNs have also been used in the automatic evaluation of NLG models [7]. Whilst the RNN has proven effective at modelling language, it suffers from an inability to adequately 'remember' relevant information over long sequences [46]. To solve this, two variants of RNN have been proposed:

- **Long Short-Term Memory (LSTM)**: LSTM is a form of RNN that is equipped with an additional memory cell which allows it to better remember information over time (and thus handle longer text more effectively) [46]. To achieve this, LSTM utilises a series of gates in order to dictate when pieces of information are remembered and when they are forgotten [46]. Due to its ability to model longer texts, LSTM has been frequently used in NLG. Models leveraging LSTMs for NLG evaluation have also been proposed [7]. In addition, ELMo, a popular state-of-the-art language model, leverages BiLSTM which is a type of LSTM. Beyond typical NLG tasks, LSTMs have also been used for knowledge-enhanced NLG by incorporating BiLSTM-based keyword extraction [67].
- **Gated Recurrent Unit (GRU)**: Like LSTMs, GRUs are another refinement of the RNN. GRUs are similar to LSTMs, leveraging gating to help mitigate the problems posed by longer sequences [46]. However, GRUs are simpler in nature than LSTM, with fewer gates and no additional memory cell [46]. This typically allows GRUs to be trained faster and to achieve better performances than LSTMs on smaller amounts of training data. However, this also means that GRUs are typically less effective at handling longer sequences. Similar to LSTMs, GRUs have seen frequent use as a generation method, and have also been proposed as a means of evaluating NLG models [7].

Convolutional Neural Networks (CNN): Whilst more commonly used in computer vision tasks, CNNs have recently seen a wide area of utilisation in NLG, from text generation to topic modelling for knowledge-enhanced NLG [67]. In NLG, CNN-based encoder-decoder frameworks have been increasingly preferred.

Graph Neural Networks (GNN): GNNs are neural models that capture the dependence of graphs via message passing between the nodes of graphs. They have the potential to combine graph representation learning and text generation. This can enable the integration of knowledge graphs, dependency graphs, and other graph structures into NLG [67].

4.3. Transformers

Transformers are deep learning models adopting an attention mechanism which can provide context for any position in the input [61]. Therefore, unlike RNNs, Transformers do not need to process data in order. This paves the way for greater parallelisation, which reduces the amount of time required for training. Therefore, the Transformer architecture enables

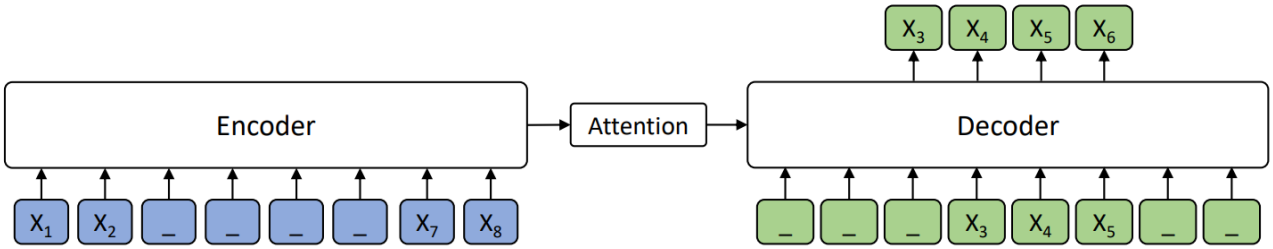


Figure 13: Encoder-decoder architecture of MASS [57].

models to be trained over larger corpora. As a consequence of this, several state-of-the-art pre-trained language models (PLMs) used for NLG are based on Transformers [14, 48]. In terms of their architectures, Transformers can be categorised into three categories:

Encoder-Only Transformers: These types of Transformer only leverage a single Transformer encoder block to build a language model. The most well-known examples are as follows:

- **Bidirectional Encoder Representations from Transformers (BERT)** [14]: BERT is a PLM developed by Google, which is used in a wide range of NLP tasks. Moreover, it is contextual and bidirectional, meaning that BERT can contextualise each word in an input utilising both its left and right context. BERT has been widely adapted as an NLG method [36] and has also seen use in the development of metrics for NLG evaluation, such as BERTScore, RoBERTa-STS and BLEURT [7].
- **Unified pre-trained Language Model (UniLM)** [15]: UniLM, developed by Microsoft, combines multiple language model pre-training objectives: unidirectional (both left-to-right and right-to-left), bidirectional and sequence-to-sequence prediction.

Decoder-Only Transformers: This type of Transformer contains only a single Transformer decoder block used for language modelling.

- **Generative Pre-trained Transformer (GPT)** [6]: GPT is a unidirectional autoregressive PLM, developed by OpenAI. It has two successors, GPT-2 and GPT-3, which have been trained on larger datasets and have larger numbers of training parameters. While all GPT versions share a similar architecture, GPT-3 is the largest PLM with 175 billion parameters.
- **Conditional Transformer Language Model (CTRL)** [29]: CTRL is a PLM developed by Salesforce that allows users to control generated content by providing *control codes*. Control codes can be URLs, questions, or languages, and enable users to explicitly specify domains, subdomains, entities and dates. CTRL has been trained on 50 control codes.
- **XLNet** [65]: XLNet is a generalised autoregressive pre-trained method that utilises permutation language modelling to combine the advantages of autoregressive and bidirectional language modelling objectives. It employs Transformer-XL, an improved Transformer architecture, as the backbone model.

Encoder-Decoder Transformers: This corresponds to the standard encoder-decoder architecture in which there are two stacks of Transformer blocks. The encoder is thus fed with an input sequence, and the decoder tries to generate the output sequence based on an encoder-decoder self-attention mechanism [36].

- **BART** [35]: BART is a denoising autoencoder for pre-training sequence-to-sequence models, developed by Facebook. It combines autoregressive and bidirectional language modelling objectives by using a bidirectional encoder (e.g., BERT) and an autoregressive decoder (e.g., GPT).
- **Masked Sequence to Sequence Pre-training for Language Generation (MASS)** [57]: MASS is a masked bidirectional sequence-to-sequence pre-training

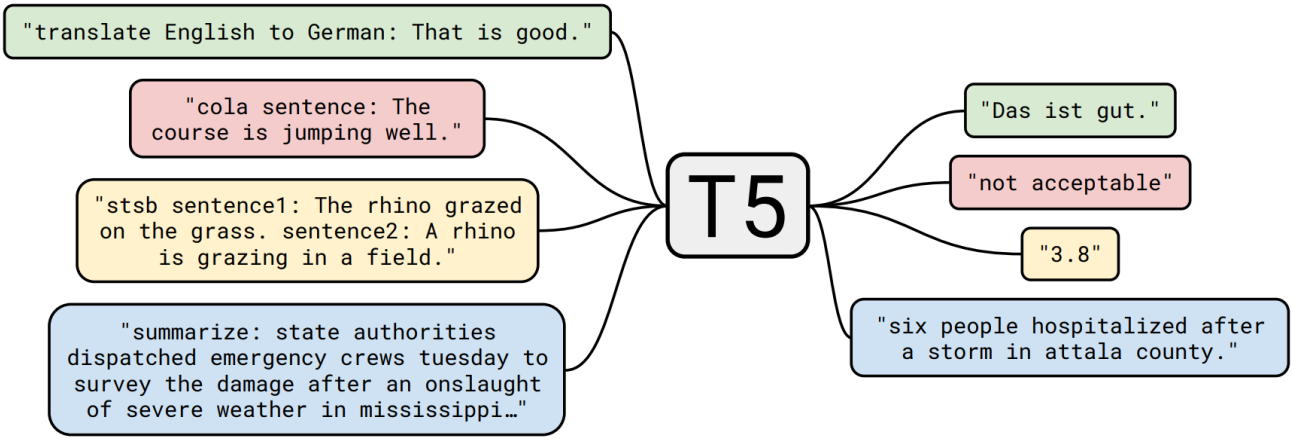


Figure 14: Some example input-output pairs generated with T5 [49].

method for NLG, proposed by Microsoft. It jointly trains the encoder and decoder by feeding the encoder with a sentence containing a randomly masked fragment, and using the decoder to try to predict the masked fragment, as shown in Figure 13.

- **Text-To-Text Transfer Transformer (T5)** [49]: T5 is a text-to-text framework proposed by Google to reframe all NLP tasks into a unified text-to-text format in which the input and the output are always text strings, rather than a class label or a span of the input. The idea behind this proposal is to be able to use the same model, loss function, and hyperparameters on any NLP task. Some example input-output pairs can be seen in Figure 14. While not its intended use, T5 can be adapted for controllable NLG [46].

4.4. Combined AI Techniques

Considering the complexity of the NLG task, it is a reasonable approach to combine multiple AI techniques to be able to make use of the advantages of each technique. Some examples of popular combined AI techniques are mentioned below.

Plug-and-Play Language Model (PPLM) [12]: PPLM, developed by Uber, is a language model aimed at controlled NLG which allows users to flexibly plug in small attribute models representing the desired control objective(s) into a large, unconditional language model, e.g., GPT. The main difference between PPLM and CTRL is that PPLM does not require any additional training or fine-tuning.

Generative Adversarial Network (GAN): GANs contain two neural networks, a generator and a discriminator, which compete with each other to provide more accuracy. CNN and RNN models, as well as their variants, are frequently used as generators and/or discriminators in GANs. Since GANs were originally designed for generating differentiable values, using it for discrete language generation is not easy. However, GANs still have several applications in the context of NLG, such as poetry and lyrics generation [55]. There exist a number of GAN variants specialised for NLG:

- **seqGAN** is a sequence generation framework aimed at solving the problems of GANs regarding discrete token sequence generation, e.g., texts. It considers the GAN generator as a Reinforcement Learning (RL) agent, and the RL reward signal is received from the GAN discriminator judged on a complete sequence [66].
- **I2P-GAN** is a GAN-based model for automatic poem generation relevant to an input image. The model involves a deep coupled visual-poetic embedding model to learn poetic representations from images and a multi-adversarial training procedure optimised with policy gradient. In its architecture, there exists a CNN-RNN generator that acts as an agent, and two discriminators provide rewards to the policy gradient [40].
- **RankGAN** focuses on one of the limitations of GAN discriminators, which is that they are generally binary classifiers. In this manner, it

aims to improve GAN for generating high-quality language descriptions by enabling the discriminator to analyse and rank a collection of human-written and machine-generated sentences. RankGAN uses LSTMs for the generator and a CNN for the discriminator [39].

- **MaskGAN** is an actor-critic conditional GAN which can fill in missing text according to the surrounding context. It uses LSTMs for both the generator and the discriminator [16].
- **LeakGAN** addresses the limitation of GANs regarding long text generation, i.e., more than 20 words. Its main idea is that the discriminator leaks its extracted high-level features to the generator in order to provide richer information. LeakGAN architecture contains a CNN as the discriminator, and two LSTMs as the generator. While one of the LSTMs is in charge of obtaining leaked features from the discriminator as the *Manager*, the other performs the generation as the *Worker*, according to the guiding goal formed by the Manager [22].

References

[1] Jacopo Amidei, Paul Piwek, and Alistair Willis. 2019. Agreement is Overrated: A Plea for Correlation to Assess Human Evaluation Reliability. In *Proceedings of the 12th International Conference on Natural Language Generation*. ACL, 344–354. https://doi.org/10.18653/v1/W19-8642

[2] Jacopo Amidei, Paul Piwek, and Alistair Willis. 2019. The Use of Rating and Likert Scales in Natural Language Generation Human Evaluation Tasks: A Review and Some Recommendations. In *Proceedings of the 12th International Conference on Natural Language Generation*. ACL, 397–402. https://doi.org/10.18653/v1/W19-8648

[3] Anya Belz, Simon Mille, and David M. Howcroft. 2020. Disentangling the Properties of Human Evaluation Methods: A Classification System to Support Comparability, Meta-Evaluation and Reproducibility Testing. In *Proceedings of the 13th International Conference on Natural Language Generation*. ACL, 183–194. https://aclanthology.org/2020.inlg-1.24

[4] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. ACM, 610–623. https://doi.org/10.1145/3442188.3445922

[5] Eleftheria Briakou, Sweta Agrawal, Ke Zhang, Joel Tetreault, and Marine Carpuat. 2021. A Review of Human Evaluation for Style Transfer. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics*. ACL, 58–67. https://doi.org/10.18653/v1/2021.gem-1.6

[6] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, ..., and Dario Amodei. 2020. Language Models are Few-Shot Learners. arXiv:2005.14165 [cs.CL] https://arxiv.org/abs/2005.14165

[7] Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. 2020. Evaluation of Text Generation: A Survey. arXiv:2006.14799 [cs.CL] https://arxiv.org/abs/2006.14799

[8] Khyathi Raghavi Chandu and Alan W Black. 2020. Positioning Yourself in the Maze of Neural Text Generation: A Task-Agnostic Survey. arXiv:2010.07279 [cs.CL] https://arxiv.org/abs/2010.07279

[9] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde, Jared Kaplan, Harri Edwards, Yura Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, ..., and Wojciech Zaremba. 2021. Evaluating Large Language Models Trained on Code. arXiv:2107.03374 [cs.LG] https://arxiv.org/abs/2107.03374

[10] Jacob Cohen. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement* 20, 1 (1960), 37–46. https://doi.org/10.1177/001316446002000104

[11] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. ACL, 8440–8451. https://doi.org/10.18653/v1/2020.acl-main.747

[12] Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and Play Language Models: A Simple Approach to Controlled Text Generation. arXiv:1912.02164 [cs.CL] https://arxiv.org/abs/1912.02164

[13] Kees van Deemter, Mariët Theune, and Emiel Krahmer. 2005. Real Versus Template-Based Natural Language Generation: A False Opposition? *Computational linguistics* 31, 1 (2005), 15–24. https://doi.org/10.1162/0891201053630291

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs.CL] https://arxiv.org/abs/1810.04805

[15] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified Language Model Pre-Training for Natural Language Understanding and Generation. arXiv:1905.03197 [cs.CL] https://arxiv.org/abs/1905.03197

[16] William Fedus, Ian Goodfellow, and Andrew M Dai. 2018. MaskGAN: Better Text Generation via Filling in the_. arXiv:1801.07736 [stat.ML] https://arxiv.org/abs/1801.07736

[17] Sarah E. Finch and Jinho D. Choi. 2020. Towards Unified Dialogue System Evaluation: A Comprehensive Analysis of Current Evaluation Protocols. In *Proceedings of the 21st Annual Meeting of the Special Interest Group on Discourse and Dialogue*. ACL, 236–245. https://aclanthology.org/2020.sigdial-1.29

[18] Joseph L. Fleiss. 1971. Measuring Nominal Scale Agreement Among Many Raters. *Psychological Bulletin* 76, 5 (1971), 378–382. https://doi.org/10.1037/h0031619

[19] Chuang Gan, Zhe Gan, Xiaodong He, Jianfeng Gao, and Li Deng. 2017. StyleNet: Generating Attractive Visual Captions with Styles. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 955–964. https://doi.org/10.1109/CVPR.2017.108

[20] Cristina Garbacea, Samuel Carton, Shiyan Yan, and Qiaozhu Mei. 2019. Judge the Judges: A Large-Scale Evaluation Study of Neural Language Models for Online Review Generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*. ACL, 3968–3981. https://doi.org/10.18653/v1/D19-1409

[21] Sebastian Gehrmann, Tosin Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Aremu Anuoluwapo, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna Clinciu, Dipanjan Das, Kaustubh D Dhole, et al. 2021. The GEM Benchmark: Natural Language Generation, its Evaluation and Metrics. arXiv:2102.01672 [cs.CL] https://arxiv.org/abs/2102.01672

[22] Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang. 2018. Long Text Generation via Adversarial Training with Leaked Information. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32. AAAI, 5141–5148. https://ojs.aaai.org/index.php/AAAI/article/view/11957

[23] Mika Hämäläinen and Khalid Alnajjar. 2021. Human Evaluation of Creative NLG Systems: An Interdisciplinary Survey on Recent Papers. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics*. ACL, 84–95. https://doi.org/10.18653/v1/2021.gem-1.9

[24] Tatsunori B. Hashimoto, Hugh Zhang, and Percy Liang. 2019. Unifying Human and Statistical Evaluation For Natural Language Generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. ACL, 1689–1701. https://doi.org/10.18653/v1/N19-1169

[25] Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning Distributed Representations of Sentences from Unlabelled Data. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. ACL, 1367–1377. https://aclanthology.org/N16-1162.pdf

[26] David M. Howcroft, Anya Belz, Miruna-Adriana Clinciu, Dimitra Gkatzia, Sadid A. Hasan, Saad Mahamood, Simon Mille, Emiel van Miltenburg, Sashank Santhanam, and Verena Rieser. 2020. Twenty Years of Confusion in Human Evaluation: NLG Needs Evaluation Sheets and Standardised Definitions. In *Proceedings of the 13th International Conference on Natural Language Generation*. ACL, 169–182. https://aclanthology.org/2020.inlg-1.23

[27] Matthew Hutson. 2021. Robo-Writers: The Rise and Risks of Language-Generating AI. *Nature* 591, 7848 (2021), 22–25. https://www.nature.com/articles/d41586-021-00530-0

[28] Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The State and Fate of Linguistic Diversity and Inclusion in the NLP World. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. ACL, 6282–6293. https://doi.org/10.18653/v1/2020.acl-main.560

[29] Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL: A Conditional Transformer Language Model for Controllable Generation. arXiv:1909.05858 [cs.CL] https://arxiv.org/abs/1909.05858

[30] Daniel Khashabi, Gabriel Stanovsky, Jonathan Bragg, Nicholas Lourie, Jungo Kasai, Yejin Choi, Noah A Smith, and Daniel S Weld. 2021. GENIE: A Leaderboard For Human-in-the-Loop Evaluation of Text Generation. arXiv:2101.06561 [cs.CL] https://arxiv.org/abs/2101.06561

[31] Ryan Kiros, Yukun Zhu, Russ R. Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-Thought Vectors. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*. MIT Press, 3294–3302. https://doi.org/10.5555/2969442.2969607

[32] Klaus Krippendorff. 1970. Estimating the Reliability, Systematic Error and Random Error of Interval Data. *Educational and Psychological Measurement* 30, 1 (1970), 61–70. https://doi.org/10.1177/001316447003000105

[33] Guillaume Lample and Alexis Conneau. 2019. Cross-lingual Language Model Pretraining. arXiv:1901.07291 [cs.CL] https://arxiv.org/abs/1901.07291

[34] Alon Lavie and Abhaya Agarwal. 2007. METEOR: An Automatic Metric For MT Evaluation With High Levels of Correlation With Human Judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*. ACL, 228–231. https://aclanthology.org/W07-0734.pdf

[35] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. ACL, 7871–7880. https://doi.org/10.18653/v1/2020.acl-main.703

[36] Junyi Li, Tianyi Tang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. Pretrained Language Models for Text Generation: A Survey. arXiv:2105.10311 [cs.CL] https://arxiv.org/abs/2105.10311

[37] Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2020. CommonGen: A Constrained Text Generation Challenge for Generative Commonsense Reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. ACL, 1823–1840. https://doi.org/10.18653/v1/2020.findings-emnlp.165

[38] Chin-Yew Lin. 2004. ROUGE: A Package For Automatic Evaluation of Summaries. In *Text Summarization Branches Out*. ACL, 74–81. https://aclanthology.org/W04-1013.pdf

[39] Kevin Lin, Dianqi Li, Xiaodong He, Zhengyou Zhang, and Ming-Ting Sun. 2017. Adversarial Ranking for Language Generation. arXiv:1705.11001 [cs.CL] https://arxiv.org/abs/1705.11001

[40] Bei Liu, Jianlong Fu, Makoto P. Kato, and Masatoshi Yoshikawa. 2018. Beyond Narrative Description: Generating Poetry from Images by Multi-Adversarial Training. In *Proceedings of the 26th ACM International Conference on Multimedia (MM '18)*. ACM, 783—791. https://doi.org/10.1145/3240508.3240587

[41] Lajanugen Logeswaran and Honglak Lee. 2018. An Efficient Framework For Learning Sentence Representations. arXiv:1803.02893 [cs.CL] https://arxiv.org/abs/1803.02893

[42] Ryan Lowe, Michael Noseworthy, Iulian Vlad Serban, Nicolas Angelard-Gontier, Yoshua Bengio, and Joelle Pineau. 2017. Towards an Automatic Turing Test: Learning to Evaluate Dialogue Responses. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. ACL, 1116–1126. https://aclanthology.org/P17-1103.pdf

[43] Sidi Lu, Yaoming Zhu, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Neural Text Generation: Past, Present and Beyond. arXiv:1803.07133 [cs.CL] https://arxiv.org/abs/1803.07133

[44] Alexandre Papadopoulos, Pierre Roy, and François Pachet. 2014. Avoiding Plagiarism in Markov Sequence Generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 28. AAAI, 2731–2737. https://ojs.aaai.org/index.php/AAAI/article/view/9126

[45] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method For Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. ACL, 311–318. https://doi.org/10.3115/1073083.1073135

[46] Shrimai Prabhumoye, Alan W. Black, and Ruslan Salakhutdinov. 2020. Exploring Controllable Text Generation Techniques. In *Proceedings of the 28th International Conference on Computational Linguistics*. ICCL, 1–14. https://doi.org/10.18653/v1/2020.coling-main.1

[47] Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-Text Generation with Content Selection and Planning. In *The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)*. AAAI, 6908–6915. https://ojs.aaai.org/index.php/AAAI/article/view/4668

[48] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. *Language Models Are Unsupervised Multitask Learners*. Technical Report. OpenAI. https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf

[49] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research* 21, 140 (2020), 1–67. http://jmlr.org/papers/v21/20-074.html

[50] Bruce Schneier. 2021. Using AI to Scale Spear Phishing. Schneier on Security. https://www.schneier.com/blog/archives/2021/08/using-ai-to-scale-spear-phishing.html

[51] Stephanie Schoch, Diyi Yang, and Yangfeng Ji. 2020. "This is a Problem, Don't You Agree?" Framing and Bias in Human Evaluation for Natural Language Generation. In *Proceedings of the 1st Workshop on Evaluating NLG Evaluation*. ACL, 10–16. https://aclanthology.org/2020.evalnlgeval-1.2

[52] Tal Schuster, Roei Schuster, Darsh J. Shah, and Regina Barzilay. 2020. The Limitations of Stylometry for Detecting Machine-Generated Fake News. *Computational Linguistics* 46, 2 (2020), 499–510. https://doi.org/10.1162/COLI_a_00380

[53] Joao Sedoc, Daphne Ippolito, Arun Kirubarajan, Jai Thirani, Lyle Ungar, and Chris Callison-Burch. 2019. ChatEval: A Tool For Chatbot Evaluation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*. ACL, 60–65. https://doi.org/10.18653/v1/N19-4011

[54] Thibault Sellam and Ankur P. Parikh. 2020. Evaluating Natural Language Generation with BLEURT. Google AI Blog. https://ai.googleblog.com/2020/05/evaluating-natural-language-generation.html

[55] Sakib Shahriar. 2021. GAN Computers Generate Arts? A Survey on Visual Arts, Music, and Literary Text Generation using Generative Adversarial Network. arXiv:2108.03857 [cs.AI] https://arxiv.org/abs/2108.03857

[56] Wissam Siblini, Charlotte Pasqual, Axel Lavielle, Mohamed Challal, and Cyril Cauchois. 2019. Multilingual Question Answering From Formatted Text Applied to Conversational Agents. arXiv:1910.04659 [cs.CL] https://arxiv.org/abs/1910.04659

[57] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: Masked Sequence to Sequence Pre-training for Language Generation. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*. PMLR, 5926–5936. https://proceedings.mlr.press/v97/song19d.html

[58] Yi Sun, Hangping Qiu, Yu Zheng, Chaoran Zhang, and Chao Hao. 2021. Knowledge Enhancement for Pre-trained Language Models: A Survey. *Journal of Chinese Information Processing* 35, 7 (2021), 10–29. http://jcip.cipsc.org.cn/CN/Y2021/V35/I7/10

[59] Chris van der Lee, Albert Gatt, Emiel van Miltenburg, Sander Wubben, and Emiel Krahmer. 2019. Best Practices for the Human Evaluation of Automatically Generated Text. In *Proceedings of the 12th International Conference on Natural Language Generation*. ACL, 355–368. https://doi.org/10.18653/v1/W19-8643

[60] Emiel van Miltenburg, Chris van der Lee, Thiago Castro-Ferreira, and Emiel Krahmer. 2020. Evaluation Rules! On the Use of Grammars and Rule-Based Systems for NLG Evaluation. In *Proceedings of the 1st Workshop on Evaluating NLG Evaluation*. ACL, 17–27. https://aclanthology.org/2020.evalnlgeval-1.3

[61] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS'17)*. Curran Associates Inc., 5998–6008. https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[62] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. CIDEr: Consensus-Based Image Description Evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 4566–4575. https://doi.org/10.1109/CVPR.2015.7299087

[63] Nai-Yu Wang, Yu-Xin Ye, Lu Liu, Li-Zhou Feng, Tie Bao, and Tao Peng. 2021. Language Models Based on Deep Learning: A Review. *Journal of Software* 32, 4 (2021), 1082–1115. http://www.jos.org.cn/jos/article/abstract/6169

[64] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. ACL, 483–498. https://doi.org/10.18653/v1/2021.naacl-main.41

[65] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R. Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Proceedings of 33rd Conference on Neural Information Processing Systems*. Curran Associates, Inc., 5754–5764. https://proceedings.neurips.cc/paper/2019/file/dc6a7e655d7e5840e66733e9ee67cc69-Paper.pdf

[66] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31. AAAI, 2852–2858. https://ojs.aaai.org/index.php/AAAI/article/view/10804

[67] Wenhao Yu, Chenguang Zhu, Zaitang Li, Zhiting Hu, Qingyun Wang, Heng Ji, and Meng Jiang. 2020. A Survey of Knowledge-Enhanced Text Generation. arXiv:2010.04389 [cs.CL] https://arxiv.org/pdf/2010.04389.pdf

[68] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. BERTScore: Evaluating Text Generation with BERT. arXiv:1904.09675 [cs.CL] https://arxiv.org/abs/1904.09675

[69] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J. Corso, and Jianfeng Gao. 2020. Unified Vision-Language Pre-Training for Image Captioning and VQA. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 7 (2020), 13041–13049. https://doi.org/10.1609/aaai.v34i07.7005

[70] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books. In *Proceedings of the IEEE International Conference on Computer Vision*. IEEE, 19–27. https://doi.org/10.1109/ICCV.2015.11