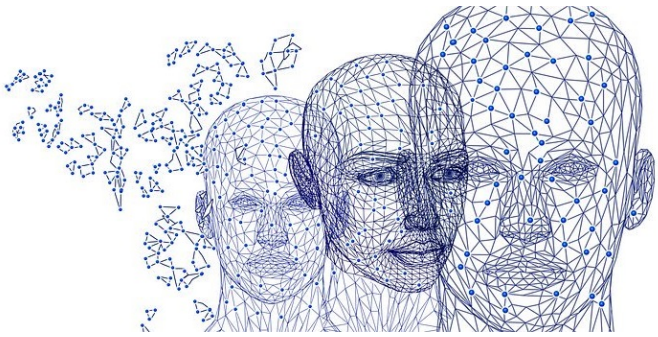
December 2021, Issue Code NL-2022-5

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, Yichao Wang, 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

Earlier this year, a team of researchers from Singapore's Government Technology Agency (GovTech) leveraged the GPT-3 language model to generate spear-phishing emails, which were able to fool study participants more effectively than those crafted by humans [38]. This finding offered one of many recent confirmations of the potential uses of powerful natural language generation (NLG) systems for malicious, deceptive purposes. Since the release of older language models, such as OpenAI's GPT-2 [40], researchers have noted that models capable of generating convincing text could be used to cause harm. With these recent findings by GovTech, alongside other studies investigating the power of NLG models in generating misinformation [44], deceptive blog posts [22], and extremist propaganda [31], it appears that the use of NLG as a means of deception and harm is no longer theoretical. Moreover, recent studies have noted that state-of-the-art NLG systems are capable, in a range of domains, of generating texts that can only be detected by humans at a chance level [26]. In essence, current NLG models are capable of generating dangerous texts, and are able to do so with a level of fluency that is able to consistently fool human detectors.

Alongside these dangers, there are a range of additional risks that have been noted in relation to the powerful pre-trained language models (PLM), such as GPT-2/3, T5, and BERT, that are typically used in state-of-the-art NLG. Whilst these models have shown great potential in their ability to offer high levels of performance in a range of language tasks (including text generation) requiring minimal amounts of task-specific fine-tuning data to do so, they also bring with them a range of additional dangers [2]. A key concern here is the issue of bias, where these models – generally pre-trained on massive datasets of online text – internalise the biases present in the datasets. Studies, in turn, have noted that these language models often exhibit a propensity for generating text that is toxic, hateful, or discriminatory in nature [2, 15]. Moreover, further issues and concerns have also been raised about these systems, e.g., they often do not perform adequately when processing languages other than English [36], financial and environmental costs that are typically required in order to train these models can lead to equality concerns and negative impacts on environment.

In this issue, we focus on the range of challenges and open questions that exist in regard to the development of state-of-the-art NLG systems. In turn, we examine the current potential of powerful NLG systems for conducting deception and discuss the range of proposed methods that have been presented to detect these forms of deception. Alongside this, we also examine the range of additional risks that the PLMs that typically underpin current NLG systems bring; examining their potential risks and harms, and the solutions that have been proposed to mitigate them.

To facilitate this discussion, we leverage the selection of NLG-focused survey articles that were sourced for the previous two issues of this newsletter (NL-2022-3 and NL-2022-4). In brief, these articles were selected via a manual examination of a range of academic venues relevant to NLG. Through this examination, we identified all literature review and survey-like articles relevant to NLG that had been published in the last 3 years (2019–2021). For this issue, we particularly focused on the survey articles which offered discussion of the challenges and open questions facing NLG, with a particular focus on those challenges that relate to deceptive or otherwise harmful misuses of NLG. For an in-depth discussion of this screening process, please refer to the Editorial of NL-2022-3.

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

List of Acronyms

- AI: Artificial Intelligence
- API: Application Programming Interface
- ASR: Automatic Speech Recognition
- BERT: Bidirectional Encoder Representations from Transformers
- BLEU: Bilingual Evaluation Understudy
- CAT: Computer-Assisted Translation
- CBIR: Content-Based Image Retrieval
- CDA: Counterfactual Data Augmentation
- CIDEr: Consensus-Based Image Description Evaluation
- CNN: Convolutional Neural Network
- EMNLP: Empirical Methods in Natural Language Processing
- GAN: Generative Adversarial Network
- GLTR: Giant Language Model Test Room
- GovTech: Government Technology Agency
- GPT: Generative Pre-Trained Transformer
- GPU: Graphical Processing Unit
- IRC: Internet Relay Chat
- LM: Language Model
- LSTM: Long Short-Term Memory
- METEOR: Metric for Evaluation for Translation with Explicit Ordering
- MT: Machine Translation
- NLG: Natural Language Generation/Generator
- NLP: Natural Language Processing
- NN: Neural Networks
- PLM: Pre-Trained Language Model
- RNN: Recurrent Neural Network
- ROUGE: Recall-Oriented Understudy for Gisted Evaluation
- SPICE: Semantic Propositional Image Caption Evaluation
- ST: Speech Translation
- T5: Text-To-Text Transfer Transformer
- TF-IDF: Term Frequency–Inverse Document Frequency
- TPU: Tensor Processing Unit
- UUID: Universally Unique Identifiers
- WMD: Word Mover's Distance
- WNGT: Workshop on Neural Generation and Translation
- XAI: eXplainable Artificial Intelligence

1. Definitions & Scope

Whilst natural language generation – that is, the use of automated systems to artificially create text – has evolved rapidly over the recent years, this rapid increase in its abilities has brought with it the capacity for misuse and other harms. The coherent and convincing nature of the texts produced by these systems means that it can often be hard for human readers to identify when they are engaging with genuine, human-created text, and when they are reading text artificially generated by machines [26]. This difficulty in distinguishing between human-created and machine-generated text opens the door for a range of potentially harmful applications. This includes the use of NLG systems to produce fake news and misinformation at scale [44, 57], to create fake reviews that mislead would-be purchasers [17], and the use of NLG to create highly convincing phishing and spam emails [26].

Moreover, as NLG systems become ever more reliant on the use of powerful pre-trained language models [6], such as GPT-2/3 [4], T5 [11], and BERT [9], as the basis from which these text generating models are implemented, they, in turn, inherit the host of potential harms that these models have been identified as possessing [3]. As PLMs are almost always pre-trained on massive datasets of largely unprocessed web data, they can absorb the biases and harmful messaging often inherent in these datasets. These biases are then carried through to the generation process, often leading to inadvertent harmful messaging being produced by PLM-based NLG systems. Additionally, the massive nature of these models means that the resources required for their training and development threatens to cause increasingly problematic environmental costs, whilst the financial costs required to build these models limit who can feasibly access these technologies [36, 50].

In this issue, we examine the range of potential harms posed by state-of-the-art neural NLG solutions, the range of potential mitigation strategies that have been proposed, and the open questions and challenges that remain unanswered. The issue is structured as follows:

Section 2: Before proceeding with the core focus of the newsletter on challenges of NLG, we present a section dedicated to a further NLG task that was not covered in our previous issue: **Translation and Interpretation**. Conventionally, this task focuses on the translation of a text in a specific language to another, user-specified language [30]. However, translation can be more broadly conceived, encompassing image-captioning (the translation of an image to natural language text) [23], speech recognition (the translation of audio to text), and source code comment generation (the translation of source code to text) [25]. In essence, this task aims to translate a given input (be that an image, audio file, or text in a different language) to natural language text (in a desired language).

Section 3: In this section, we move onto the topic of deception and detection in NLG. This section begins by summarising the presence of deception in online communications, including its notable role in fake reviews and trolling, before examining the manner in which NLG systems have been leveraged to help facilitate these forms of online textual deception [17]. From here, we then examine the mitigation strategies that have been developed for more typical online deception and the difficulties that NLG-based deception poses to these approaches. Finally, we summarise current methods for identifying deceptive NLG texts, and evaluate the current limitations and challenges still posed by deceptive NLG [26].

Section 4: Finally, in this section we direct our attention to the range of risks and harms posed by current neural NLG methods, particularly in regard to their reliance on PLMs. This section summarises the range of risks and harms that PLM-based NLG systems may pose beyond deception, particularly in terms of the range of problems that may exist due to the inherent biases present in the PLMs themselves [3]. We also examine the current proposed strategies for tackling these biases, and the challenges and open questions that remain. Finally, we give a brief examination of the further environmental and financial impacts that these massive NLG systems pose, and the proposed solutions to respond to these emerging problems [50].

2. Translation & Interpretation

Translation has been broadly construed to cover all NLG tasks in which a given data input (e.g., text in another language) is translated so that it is represented in natural language text.

This, in turn, encompasses more than just translation from one language to another, also including other mediums, such as the translation of an image to a text caption that describes the image (which can be thought of as translating the image from a visual medium to text). In some cases, interpretation is also required, such as when a machine attempts to process puns when translating natural language, summarise an image when generating image captions, or assess the context logic of source code when generating comments.

In this section, we split the translation and interpretation task into five subtasks: *language translation*, *image captioning*, *speech recognition*, *explainability*, and *code comment generation*. Language translation refers to the use of a machine to translate a text written in a source language into a text written in a target language. Image captioning means translating image information into text describing the image's content. Speech recognition translates audio information into meaningful text. As a translation subtask, explainability aims at using NLG to generate explanations of the behaviours of black box models. Code comment generation focuses on automatically generating code comments based on a given source code input, which can be conceptualised as translating source code to text.

For each subtask, we define some of its relevant potential end-user applications. We also discuss typical approaches, typical evaluation methods, and any relevant datasets.

2.1. Language Translation

Language translation, typically referred to as machine translation (MT), is the process of automating the translation of text from one natural language to another [30]. In this subsection, we cover some of the key aspects of the MT subtask, including document-level machine translation, humorous wordplay translation, and translation quality assessment.

Document-level machine translation (otherwise known as discourse-level machine translation) refers to a translation process that utilises inter-sentential context information, which includes the discourse of text and the surrounding sentences in the input document [30, 58]. Figure 1 provides an example of the basic architecture of a document-level MT model, which also indicates how inter-sentential context information is useful.

The document-level machine translation context approaches are classified based on two dimensions:

- Whether the approaches use local or global models; where local models use only the neighbourhood of the current sentence while the global models consider both past and future translation decisions.

- Where the context comes from i.e., from the source-side, or from both the source and target-side contexts. For example, in Figure 1, the left side is the source document, the right side is the target document. Approaches can utilise both sides or source-side only. There is no model that works on the target side only as it is natural to leverage the existing source-side context when performing translation [30].

Approaches are then classified into four groups: **(1)** local source-only context, **(2)** local source and target context, **(3)** global source-only context, **(4)** global source and target context. Furthermore, the learning approaches are classified into two groups: **(1)** modifying the training strategy and **(2)** utilising contextualised word embeddings. The former involves the introduction of regularisation and reward functions in the training objective, or even modifying the learning process. The latter provides a warm-start for the training process, using the MT model to predict both the target-side sentence and the source-side context. The idea is that the source-side sentence embeddings can be integrated into the MT model in order to utilise the source-side document-level dependencies [30].

There has been much research focused on encodings for document-level MT. Cache memory methods have been introduced, which can carry over word preferences from one sentence to the next [58]. Neural-network (NN)-based discourse-level approaches, which predict the next possible sentence by retrieving the historical conversation, have also been suggested [30].

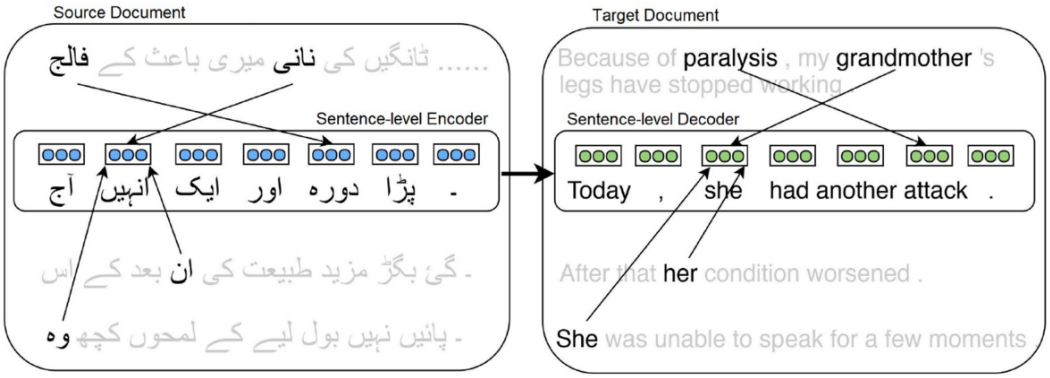


Figure 1: Example of a document-level MT model [30].

.

Humorous wordplay translation is a popular research problem in language translation – referring to the challenge of translating puns from one language to another [34, 36]. However, most studies have aimed to resolve the translation problems as the single "correct" interpretation [34], rather than addressing the problem of translating the pun itself, which typically contains multiple intended meanings. Delabastita [8] covers eight common strategies for translating wordplay:

1. Replace the source-language pun with a similar target-language pun.
2. Substitute the pun with non-punning language that preserves one or both of the original meanings.
3. Replace the pun with some non-punning wordplay or rhetorical device (e.g., irony, alliteration, vagueness).
4. Omit the language containing the pun.
5. Leave the pun in the source (original) language.
6. As a compensatory measure, introduce a new pun at a discourse position where the original had none.
7. As a compensatory measure, introduce entirely new material containing a pun.
8. Editorialise: insert a footnote, endnote, etc. to explain the pun.

In general, existing MT approaches are unable to deal with humour translation problems. There are, in turn, three major research directions that have been identified for further study: **(1)** to study how human translators process puns, **(2)** to generate and rank lists of pun translation candidates, **(3)** to develop interactive NLP-based methodologies for supporting human translators, which help assess whether a given pun can be replaced with a target-language pun [34].

For evaluation of MT, human assessment methods are most commonly used [21]. As is typical amongst NLG more broadly, the quality criteria of *intelligibility*, *fluency*, *adequacy*, and *comprehensibility* have been primarily considered (see NL-2022-3 for further discussion of typical human NLG evaluation measures). For document-level MT, evaluation of pronoun translation, lexical cohesion and discourse connectives have also been utilised. Beyond this, more advanced evaluation approaches also exist. These include using extended quality criteria including: *suitability*, whether the results are suitable in the desired context; *interoperability*, whether the MT system works with other platforms; *reliability*, whether the MT system will fail (and its fault rate); and *usability*, whether the MT system is easy to learn and operate.

Automated assessment methods have also been leveraged in evaluating MT [21]. This includes common metrics such as edit distance, word error rate, translation edit rate, position-independent word error rate, BLEU, METEOR, ROUGE, precision, and recall. Neural networks, especially Deep Learning, have also been suggested for translation quality assessment, such as by using a NN to find the best translation from a series of candidate translations, via comparison with a reference translation [18, 19].

Machine translation has many key applications,

and has been widely applied in real-world systems. Common applications of MT thus include business, education and government (e.g., real-time translation during meetings, online translation software, websites with multiple languages) [30]. Moreover, electronic dictionaries, translation memories (a pre-defined database for aiding human translators), computer-assisted translation (CAT) tools and component-based CAT workbenches (for professional human translators) have all benefited from the development of MT [34].

However, despite these benefits, machine translation runs the risk of being targeted by attacks from malicious users. In 2017, the Israeli police arrested a Palestinian man after he posted "good morning" in Arabic. This is due to the fact that the MT system incorrectly translated this Arabic as "attack them" in Hebrew. Similarly, an MT system incorrectly translated a neutral Chinese phrase into a racially discriminatory phrase on Chinese social media. Attacks against MT systems may occur in the model training phase or through corpus poisoning, and could be used maliciously to cause similar incidents of mistranslation.

In language translation, a wide range of datasets have been leveraged in the training and evaluation of proposed MT systems. Some of the most commonly used datasets include:

WordNet: A widely used synonym database in the NLP literature, which groups English words into sets of synonyms. (https://wordnet.princeton.edu/).

OntoNotes: A corpus consisting of 2.9 million words in English, Arabic and Chinese. OntoNotes contains text from news, conversational telephone speech, weblogs, usenet newsgroups, broadcast, and talk shows alongside structural information (syntax and predicate-argument structure) and shallow semantics. (https://catalog.ldc.upenn.edu/LDC2013T19).

FrameNet: A database of approximately 1,200 scripts (semantic frames) covering over 13,000 English word senses. (https://framenet.icsi.berkeley.edu/fndrupal/).

ACL 2019 Fourth Conference on Machine Translation (WMT19) corpus: This corpus contains news text in ten language pairs. The language pairs are Chinese-English, Czech-English, Finnish-English, German-English, Gujarati-English, Kazakh-English, Lithuanian-English, Russian-English, German-Czech, and French-German. **Rapid corpus**, **Newscrawl Corpus**, and **Europarl v7/v9** are datasets that combine the WMT19 corpus with additional languages and sources. (http://www.statmt.org/wmt19/translation-task.html).

International Conference on Spoken Language Translation (IWSLT): IWSLT is an annual scientific conference with an open evaluation campaign. In the IWSLT, the evaluation campaign is for translating TED talks from Chinese to English (tst2010-2013), from French to English (tst2010), and from Spanish to English (tst2010-2012). Each of these language tasks contains around 200,000 sentence pairs. (https://iwslt.org/).

RotoWire: This dataset contains 4,853 NBA basketball game summaries in English between 2014 and 2017. The 3rd Workshop on Neural Generation and Translation (WNGT 2019) manually translated a portion of the RotoWire dataset to German. The dataset provider recommends using the following SportSett:Basketball dataset, which corrects some dataset contamination issues from the standard Rotowire dataset. (https://github.com/nlgcat/sport_sett_basketball)

2.2. Image Captioning

Image captioning, otherwise known as image description generation, is aimed at automatically generating a description of an image, using the image as input [23]. This is typically construed as a translation problem – translating the image information to text. This subsection discusses the general approaches for image captioning, as well as the typical evaluation approaches, popular datasets, and related applications relevant to this subtask.

There are three main approaches to image captioning [23]:

Template-Based Methods: These methods use a fixed text template for caption creation. The template contains blank slots (typically called variables), and object, attribute, and action words relevant to the image are selected by the captioning system to fill these slots. Whilst useful in some circumstances, template-based methods are limited by the static nature of the template itself, which limits the flexibility of any captions generated.

Retrieval-Based Methods: This approach focuses on retrieving the correct caption from a set of captions. Whilst this approach ensures the fluency

of the output caption, retrieval-based methods are limited to producing more general captions instead of image-specific ones.

Automatic Generation-Based Methods: These methods typically leverage some form of deep-learning model (e.g., encoder-decoder, bi-LSTM) to generate an original description of the input image. This approach offers greater capability in creating original image-specific descriptions but is often hampered by weak fluency in the generated caption.

Deep learning has been widely used for image captioning [23]. Here we discuss some of the common deep-learning-based approaches that have been used in the literature [23].

Feature Mapping: Visual space approaches are most common here; these independently pass the image features and the captions to the decoder. Multimodal space embeddings have also been proposed for feature representation, which combine image and text embeddings. AlexNet and VGGNet are commonly used as image encoders.

Learning Types: Common supervised-learning-based approaches include encoder-decoders, attention-based models, and dense image captioning. These approaches use labelled data in the training phase. Other deep-learning-based approaches include reinforcement and unsupervised techniques, which extract the image features using image encoders and then pass these features to the language decoders. Generative adversarial network (GAN)-based methods have also been successfully used in image captioning.

Captioning Types: This refers to the regions of the scene that are being captioned, and thus the form of image captioning that is being conducted [23]. These are generally classified as:

- Whole-Scene captioning, in which the captions aim to summarise the entire image. Common approaches to this include:
  - Attention-based methods: Image features are obtained using a convolutional neural network (CNN), and from this an attention-based language model generates some words or phrases. Parts of captions are then constructed from generated terms, and the captions are dynamically updated to account for the various regions of the scene within the caption. This method allows for the inclusion of images during learning steps to emphasise key regions during captioning.
  - Novel-object-based methods: Existing methods rely on paired image caption datasets. This approach uses a separate lexical classifier and language model trained on separate image and text data. Then, using paired image caption data, a deep-caption model is trained. Finally, both models are trained together. This approach allows for the generation of captions that describe novel objects not present in that paired image caption training data.
- Dense captioning, in which the system focuses on captioning specific regions of the image. To do this, typical approaches first divide the target image into different regions. Then, features are extracted from each of the different regions. These features are then passed to a language model, which generates captions for each region.

Model Architecture: There are a wide range of deep-learning architectures that have been proposed for use in image captioning. These include:

- Encoder-decoder architecture: This approach is generally used in conjunction with a CNN trained to classify scene type, objects and relations within the image. After this classification, the encoder-decoder model converts the outputs of the CNN into words and generates the caption. Each word in the caption is selected based on visual information and previous context to ensure coherence and accuracy.
- Compositional architecture: Image features are obtained by using a CNN. Then, the visual concepts (e.g., regions, objects, and attributes) obtained from these features are used as additional image features. Multiple captions are generated by a language model using the output of the above two processes. A deep multimodal model is then used to rank and select the final captions.

The evaluation of image captioning is mainly automatic and quantitative based [23]. Several metrics are commonly used when evaluating the quality of image captioning, with BLEU and METEOR being particularly popular, though ROUGE, CIDEr and

SPICE have also seen some use. Word Mover's Distance (WMD) is another metric that has been suggested, though its failure to account for word order and its inability to measure readability limit its utility [39].

A common application of image captioning systems is content-based image retrieval (CBIR), which relies on image indexing (i.e. the annotation of images in a database) as a crucial component. Image captioning, in turn, is highly suited to automatically generating captions for these data samples [23]. Another major application of image captioning is in the biomedical field, where it can be used to help physicians identify lesions in PET/CT scans or radiology images [39]. Figure 2 shows an example of a biomedical image with a generated caption. Various social media applications are also possible, such as identifying places, events, and clothes in images.

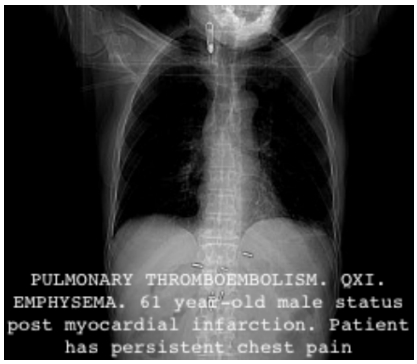


Figure 2: Example of biomedical image with caption [39].

For image captioning, the following datasets are commonly used:

MS COCO: The Microsoft COCO dataset is used for object detection, segmentation, and captioning. It contains over 330K images with five captions per image. It also has 80 object categories and 1.5 million object instances. Figure 3 gives two examples of image captioning on MS COCO datasets by Wu et al. [53]. (https://cocodataset.org/).

FLICKR 30K: FLICKR 30k has been a standard benchmark for automatic image description. It contains 30K images collected from Flickr with 158K captions provided by human annotators. (https://github.com/BryanPlummer/flickr30k_entities).

Visual Genome: This dataset has multiple region captions. It contains over 108K images with an average of 35 objects, 26 attributes, and 21 pairwise relationships between objects per image. (https://visualgenome.org/).

Beyond the above datasets, there also exist datasets specifically aimed at biomedical image captioning [39]. These include:

IU-X-ray: This dataset contains 7,470 x-rays images and is publicly accessible from the Open Access Biomedical Image Search Engine (OpenI). (https://openi.nlm.nih.gov/).

ICLEF-CAPTION (Image Concept Detection and Caption Prediction): This dataset has 232,305 biomedical images with captions. (https://www.imageclef.org/2018).

2.3. Speech Recognition

Speech recognition, also referred to as automatic speech recognition (ASR), is a task that aims to automatically translate audio information into meaningful text. Speech translation (ST), a task that builds upon work in ASR and MT, focuses on translating the audio into text in a different language [49].

Sperber and Paulik [49] surveyed the historical development of speech translation and divided it into 4 stages:

Loosely Coupled Cascades: In simple terms, researchers would separately build ASR and MT systems, then use the results of the former as the input to the latter. Since early MTs were unable to handle input with irregular formats, errors may be transmitted from the ASR stage.

Toward Tight Integration: Further studies then tried to address the early decisions problem – where errors in the output of the ASR are passed on to the MT system. To solve this, the N-best translation approach was introduced, which summarises all possible transcriptions of the ASR outputs. Another suggested approach processed the intermediate results and optimised the input structure of the MT system based on its domain. Prosody (e.g. pitch and loudness) transfer was also suggested, which is used for applying the source-side prosody to target-side words during the transformation.

Speech Translation Corpora: Before this, ASR and MT used separate corpora for training. This often led to a mismatch between ASR and MT trained on data from different domains. Researchers thus started to use the same corpus to train ASR and MT. However, the cost of manual annotation is high and suffers from language coverage limitations.

Ground Truth Caption: Two brown bears playing in a field together.

Generated Caption: Two brown bears playing on top of a lush green field.

Ground Truth Caption: A plate of breakfast food with a silver tea pot.

Generated Caption: A close up of a plate of food with a folk and a knife on a table.

Figure 3: Examples of image captioning on MS COCO datasets by Wu et al. [53].

.

End-to-End Models: End-to-end ST corpora and models for MT and ASR are now commonly used. Other approaches like end-to-end trainable cascades and triangle models, multi-task training and pre-training (incorporating additional ASR and MT data), and speech-to-speech translation have also been proposed.

The authors also defined three types of end-to-end training data (i.e. pairs of speech inputs and output translations) [49]. **(1)** Manual: The speech corpora used for training is translated by humans. **(2)** Augmented: Data is obtained by extending an ASR corpus with automatic translation or an MT corpus with synthetic speech. **(3)** Zero-Shot: Using no end-to-end data.

Four of the commonly used corpora in ASR are:

MuST-C: This dataset contains more than 385 hours of audio recordings from the English TED Talks into eight languages (German, Spanish, French, Italian, Dutch, Portuguese, Romanian and Russian). (https://ict.fbk.eu/must-c/)

MaSS: This dataset contains more than 172 hours of audio recordings from the Bible across eight languages (Basque, English, Finnish, French, Hungarian, Romanian, Russian and Spanish) (https://github.com/getalp/mass-dataset)

LibriVoxDeEn: This dataset contains 110 hours of audio recordings from German audiobooks with German text and English translation. (https://www.cl.uni-heidelberg.de/statnlpgroup/librivoxdeen/)

Europarl-ST: This dataset contains paired audio-text samples in nine languages (Romanian, Polish, Dutch, German, English, Spanish, French, Italian and Portuguese). (https://www.mllp.upv.es/europarl-st/)

2.4. Explainability

A clear definition of "explanation" has not been agreed upon by the scientific community. Generally, explanations are classified by different types, such as explanation by example, counterfactual explanation (i.e. explaining how a model's behaviour would change if its input was x instead of y), local explanation (explaining the effect of a single input), feature importance (i.e. which feature(s) have the greatest influence), or a combination thereof. In principle, eXplainable Artificial Intelligence (XAI) aims to add a "linguistic explanation layer" to decision tools to help end users and improve adoption in the market [29].

Broadly, there are two common NLG approaches to XAI in the literature: template-based, and end-to-end generation [29]. The former uses a predefined output structure and fixed predefined sentences. The latter is dynamic, using human labelled data-to-text to train models to generate sentences without any template needed. Whilst generative approaches offer more versatility and scope for creativity, they may also be vulnerable to adversarial attacks that could cause the "explanation" to be meaningless or misleading.

In terms of evaluation, there is no consensus on how to evaluate the quality of text-based explanations. This is because the evaluation process should consider not only readability but also effectiveness and usefulness of the explanation for end-users [29]. Transparency – i.e., understanding the logic or reason behind the decision – is often mentioned as a key quality of explanation; over-simplified explanations

might score well in human evaluation but fall short in transparency.

2.5. Source Code Comment Generation

Codes comments refers to text that is used to annotate part of a program's source code (e.g., a function or class), offering a natural language explanation of the code's intended behaviour. Based on this, attempts have been made at automatic code comment generation, also known as automatic code summarisation, in which a model attempts to generate a code comment using a piece of source code as input [7].

Broadly, there are three approaches to code comment generation [7]: **(1)** Template-based generation methods, which use software word usage models and templates to analyse the code structure. Commit messages have also been generated by using template-based methods based on code change and the type of the change (such as file renaming, modification of the property file). **(2)** Information retrieval-based methods, which model comment generation as an automatic text summarisation problem. This type of method attempts to identify keywords or sentences from the target code, and then treats these identified keywords or sentences as a code summary. The source of information also includes software repositories and even dialogue between developers. Additionally, Rodeghero et al. [42] leverage eye-tracking technology to identify the sentences and keywords that code developers focus on during reading code. These sentences and keywords can then be used as further sources of information. However, the key information required is often unavailable, limiting this approach's utility. **(3)** Deep learning-based methods, which model comment generation as a neural machine translation problem. CNNs and recurrent neural networks (RNN) are commonly used for this, with long short-term memory (LSTM) being particularly popular. Typically, an encoder model is used to encode the source code into a fixed-length vector representation, and then a decoder decodes the vector representation of the source code and generates code comments. The main difference between different encoder-decoders is the input form of the code and the structure of the neural network. Researchers have also recently tried to use other learning algorithms (such as neural graph networks, reinforcement learning, and dual learning) to further improve performance. Consideration of other information sources, such as application programming interface (API) sequence information, can also be used to improve the quality of the generated code comments.

In terms of evaluation, there are two types of approaches: human evaluation and automatic evaluation. The human-based approach usually scores code comments using a Likert scale based on a range of criteria including: *accuracy*, the degree to which the code comments correctly reflect the code's implementation purpose and main functions; *fluency*, the writing quality of the comment; and *accessibility*, the ease with which the generated comments can be read and understood. *Consistency* is also commonly used as a quality criterion, in which the code comments should follow a standardised style/-format [7]. Automatic evaluation approaches instead focus on comparing the similarity between the candidate comment and a reference comment (manually generated). Common automatic metrics include BLEU, METEOR, ROUGE and CIDEr [7].

There are a number of possible applications for code comment generation. These include automatically generating release notes, repairing bugs and related licence modifications, and automatic code evaluation – which compares the similarity of the generated comment to the reference comments [7].

Two of the most popularly used code comment corpora are:

DeepCom: This dataset was compiled through the use of the Eclipse Java compiler to parse Java methods and extract JavaDoc comments from it. The corpus contains 588,108 pairs of method names and comments. (https://github.com/xing-hu/DeepCom).

Nematus: This corpus was mainly collected from GitHub Python-based projects. The dataset contains a total of 108,726 code-comment pairs. (https://github.com/EdinburghNLP/nematus).

3. Deception & Detection of Generated Text

In this section we examine key issues of textual deception and the usage of NLG to further this deception. We begin by introducing the concepts and definitions of what constitutes textual deception and the typical cues of deceptions present in these texts. We will then briefly examine some cases of more typical online textual deception, in which human bad actors attempt to deceive others through online texts, such as through fake reviews or fake news. We will also briefly review common methods that have been implemented to detect these forms of deception.

Having established this baseline, we will then examine the usage of NLG as a means of deception, particularly in terms of its potential use in mimicking human writers – an aspect of current (and likely future) NLG systems that could be leveraged for a wide range of dangerous applications. Finally, we will provide an overview of the current methods that have been proposed to identify generated text, and distinguish them from texts written by human authors.

3.1. Textual Deception

Broadly speaking, deception can be defined as the use of some form of communication (e.g., text, speech) through which the deceiver aims to convince their target into believing something which the deceiver knows to be false [17]. Importantly, deception can thus be conducted through a wide range of media, and encompasses any attempts by an individual to mislead others.

In turn, deception can be divided into two distinct sub-types: *explicit* and *implicit* [17]. Explicit deception describes situations in which a deceiver attempts to convince their target into believing a proposition that the deceiver knows to be false, using a crafted utterance (or set of utterances). Crucially, in explicit deception the semantic content of the deceptive utterances directly reference the false proposition being put forward.

In contrast, implicit deception instead leverages the contextual knowledge of the deceiver's target. Thus, the deceptive utterance does not contain specific reference to the false proposition but instead relies on the deceiver's target inferring the false proposition through the deceptive utterance combined with the target's prior knowledge.

Studies of deception have typically examined three common types of deception: one explicit in nature, the other two implicit. The first, dubbed *Deception of Literal Content* by Gröndahl and Asokan [17], involves cases in which the semantic content of the text itself is deceptive. This is the case of deception that has received the most focus in terms of online and NLG-based deception. *Deception of Authority*, instead, focuses on cases in which the deceiver uses implication to mislead their target into believing they (the deceiver) have authority over an issue when they do not. Finally, *Deception of Intention* involves cases where the deceiver has some form of ulterior deceptive motive for formulating the utterance that is not clear from the utterance itself. In this case, the proposition in the utterance may in fact be true, but the deceiver's motivation for making the proposition is hidden using deceptive means.

Through an analysis of the linguistic properties of deceptive texts, researchers have, in turn, been able to identify the common cues that are indicative of deception. These include the usage of heightened emotional language, over-generalisation and a lack of specificity, an unusually high or low usage of first-person pronouns, high-verb usage, and a heightened use of certainty-based words [17]. A table of common deceptive cues that have been noted across a range of datasets can be found in Figure 4.

3.1.1. Online Deception

There are myriad ways in which malicious agents can utilise online text-based platforms to deceive [51]. Two common, highly studied examples in which malicious users attempt to mislead through online communications are **fake reviews** and **trolling**.

Fake reviews are a particularly problematic source of deception. With the rise of e-commerce, users have been given access to an unparalleled wealth of choice when it comes to making purchasing decisions. From holiday and travel, to individual product purchases, the wealth of options can make it difficult for users to identify the best choices for purchase. One common attempt to solve this is the use of review services, through which users can leave feedback in regard to the quality of a given item [54]. By crowd-sourcing these user reviews at scale, a would-

Data	Deception cues
Enron e-mails [85]	**abstractness**, negations, *first person pronouns*
Conference call transcripts	**general group references**, reduced non-extreme positive emotion terms, reduced third-person plural pronouns
Online dating profiles	**reduced first-person singular pronouns**, negations, reduced word count, *reduced negative emotion words*
Fraudulent scientific papers	words related to scientific methodology, amplifying terms, **certainty-related words**, **emotional words**, reduced diminisher terms, reduced adjectives
Enron e-mails [85]	modal, base and present tense verbs, **second-person pronouns**, function words

Figure 4: Common deception cues observed across a range of domains [17].

be purchaser is thus empowered to make more qualified decisions in regard to their online purchases.

Given the importance of online reviews in customer decision-making, fake reviews have become increasingly prevalent as a means of misleading customers into making assumptions about a product or service that is not true [54]. This includes fake reviews aimed at making a given product or service seem better or more appealing, and reviews targeted at making a product or service seem of lower quality than is actually the case [17]. Moreover, it is worth noting that fake reviews needn't contain false information to be deceptive. The use of fake reviews at scale, highlighting genuine issues or positives of a targeted product, can still cause deceit by artificially inflating the overall sentiment of reviewers towards the product. Typically, fake reviews are written by professionals, who leverage existing reviews as a means of making their own deceptive reviews seem more legitimate [54]. In turn, online fake reviews have become an ever-growing problem in e-commerce. Recent research estimated that 16-33% of online reviews studied by different groups of researchers were fake or otherwise deceptive [54].

Another common source of online deception is that of online trolling [17]. Whilst this is less obviously a case of deception, online trolling is typically constituted as a case of deception of intention. Whilst not applicable to all cases of trolling, it is not uncommon for trolls (and other cyberbullies) to broadcast misleading or otherwise false content as a means of inciting discord. In these cases, the troll does not necessarily believe the content of their posts but instead is using it to deceive others in order to breakdown communication and sow discord.

Two common sources of trolling that have been studied in the literature are that of *paid trolls* and *mentioned trolls* [33]. Paid trolls refer to individuals that post malicious content on online platforms on behalf of some form of institution such as a political candidate or corporation. Mentioned trolls refers to users that have been identified as such by other members of the online community in which they are active. Examining and attempting to distinguish between paid trolls versus non-trolls, and mentioned trolls versus non-trolls has thus been of popular interest in current research [17].

3.1.2. NLG-Based Deception

Historically, most cases of online deception – including fake reviews and trolling – have been conducted by malicious users knowingly crafting and posting misleading content. Recently, however, the growth of NLG as a viable tool, and its ability to generate text that is coherent and human in nature has meant that there is new-found scope for text generation to be leveraged to conduct deception at scale [12]. By misleading readers into believing that a given text was written by a human, when it was in fact generated by a machine, NLG has the capacity for new forms of deception beyond what has currently been seen.

In turn, NLG-based online deception has applications in any area in which online text deception is possible [17, 20]. Generally, the only limiting factor of this application is the capability of the NLG system to adequately generate convincing text in the desired medium.

In regard to the above, NLG has thus been leveraged as a means of generating fake reviews at

scale [17]. As NLG systems have improved over the years this capacity for fake review generation has grown as well, with recent systems being demonstrated that are capable of generating reviews that are specific to user-specified contexts (crucial for creating convincing reviews) [27]. Additionally, the widespread adoption of powerful generative language models (LM) has meant that fake review generation can be achieved often with minimal amounts of effort, combining these existing LMs with small amounts of context-specific fine-tuning data to create convincing fake reviews [1].

Other applications of NLG as a means of deception have also been proposed [26]. This includes the use of NLG as a means of generating fake news and misinformation, which has been found to be adequately convincing in misleading both human and machine-based detectors [43]. Additionally, examples of NLG deception include incidents in which more than 120 research articles were removed after they were discovered to have been artificially generated [52]. In this case, all generated papers had already been published, and were only identified after the fact. In another, more recent incident, a Berkeley student leveraging the OpenAI GPT-3 model [4] was able to generate fake news articles for the website Hacker News. These articles were so convincing that not only did they remain undetected for a long period of time but they also managed to reach the #1 spot on the website.

Additionally concern is the ease with which these NLG systems can be leveraged for deception. In the case of the Hacker News deception, the student leveraged the pre-existing GPT-3 model, combined with small inputs of an article title and a brief introduction. This alone was sufficient to produce highly convincing, and evidently compelling deceptive text. In turn, it is clear that the rapid progress in the quality of NLG-based texts is leading to the potential for online deception that requires minimal skill, and that can be conducted on a massive scale.

3.2. Detecting Deceptive Text

Given the widespread nature of online textual deception and the ease with which even current NLG systems can now be leveraged to facilitate deception at scale, it is important that solutions are developed that are capable of identifying deceptive texts.

Historically, efforts towards detecting online deception were focused towards cases of deception in which the deceiving texts were crafted by humans. In the case of detecting fake online reviews (written by humans), supervised methods are the most commonly leveraged solutions [17].

These approaches, in turn, have proved relatively successful in identifying fraudulent reviews [17]. To do this, common approaches have generally focused on utilising patterns in the linguistic choices of online reviews as a means of identifying deception. These approaches are often aligned with the common cues of deception noted in Section 3.1.1, where these cues are used as features by the supervised detection models to identify fake reviews [17]. Beyond linguistic features, other solutions have found that both sentiment and readability are often useful as features through which deceptive reviews can be identified [24]. Moreover, other work has focused specifically on measuring the generality of reviews, leveraging the notions that fake reviews will typically be unspecific in nature [55]. This too has been found to be effective when used to train supervised detection models, though this efficacy is often limited to products and services in which specific information can be hard to obtain (e.g., restaurants and hotels) by the fake reviewer. For other products, for which further information can be gathered from advertisements and seller details, specificity is found to be less useful.

Whilst these more classical approaches to detection have been found to be effective against human-created deception, they have proved less capable of detecting deceptive generated text [17]. This is likely due to a combination of factors. Firstly, the nature of deception is inherently different between machine-generated and human-created text. With machine-generated text, the key area of deception is generally one of identity: the generated text is being disguised as human-written. Moreover, further study has found, perhaps unsurprisingly, that there appears to be little overlap between the cues typically associated with human-crafted deception (as discussed in Section 3.1.1), and the cues of deception that denote a machine-generated text.

Given this, recent efforts have been made to develop bespoke systems dedicated to the task of distinguishing between human-created and machine-generated text [26]. This has been of particular concern as studies indicate that current state-of-the-art NLG systems are often able (context depending) to avoid detection by humans, with human detectors

identifying generated texts at just a "chance" level in some circumstances [26]. Given this, it is becoming increasingly important that detection systems capable of effectively identifying generated texts are developed. In turn, there are a variety of overarching paradigms that have been proposed for developing NLG detectors:

Supervised Systems: Using similar approaches to those adopted by human-written deception detection systems, this typically involves the training of classical machine learning algorithms (e.g., logistic regression, support vector machines, decision trees) to detect machine-generated text. Rather than using specifically-defined features (such as the linguistic and sentiment features discussed above), these methods more commonly leverage a bag-of-words approach, typically using basic n-gram frequencies or term frequency–inverse document frequencies (TF-IDF). Whilst reasonable performances are achievable, these detectors are typically limited to the specific domain (e.g., Amazon reviews) in which they are trained, showing reduced capabilities towards detecting generated texts even when applied to related domains. Moreover, these approaches have been found to suffer in performance considerably when used to detect generated texts produced by the larger, state-of-the-art LMs typically used in more recent NLG systems.

Zero-Shot Classifiers: A more recent approach is the use of existing pre-trained LMs as zero-shot classifiers to detect generated texts produced by the same, or similar pre-trained LMs. To do this, the overall likelihood of the input text (to be classified as machine-generated, or not) being generated, according to the LM detector, is compared to the likelihoods of both machine-generated and human-written reference texts. Whilst the zero-shot nature of these approaches would have distinct advantages in adaptability and generalisability, as the detector would not require additional training to be applied in different contexts, current experiments have been unable to achieve strong performances using this method. Typically, current solutions have generally been unable to outperform the classical supervised detector systems.

Fine-Tuning LMs: Somewhat combining the above two approaches, this approach leverages pre-trained LMs and fine-tunes them with further deceptive text data in order to improve its ability to detect generated text. This approach has shown great promise, generally outperforming other supervised approaches and showing strong capabilities when applied to a wide range of deceptive generators and online domains. Beyond the increase in performance, the fine-tuning approach also has the advantage of generally requiring less training data than previous supervised methods. Despite this, fine-tuning approaches do still appear to be limited in their ability to detect deceptive texts produced by models not included in their fine-tuning data. A fine-tuned detector trained on deceptive texts from the small GPT-2 model, for instance, is generally unable to detect deceptive text from the larger GPT-2 model. Despite this, fine-tuned detectors do still show better capabilities towards generalisation than other supervised methods. Moreover, despite some approaches assuming that the same LM used in deceptive generation would be best suited for detection, current studies indicate that bi-directional LMs (such as BERT and RoBERTa) may be best suited to NLG detection in a wide range of cases.

Beyond these machine-centric paradigms, other researchers focus on human-in-the-loop approaches. Rather than relying solely on a machine-based classifier, human-in-the-loop detectors aim to use machine-based NLG detectors as a means of informing and aiding human detectors, rather than as a detection system themselves.

This approach has certain advantages, as human and machine detectors are typically effective in different areas of NLG detection. Human detectors are generally more capable of identifying contradictions, semantic errors, and contextual errors, whilst machine detectors show heightened abilities in detecting over-represented, high-likelihood terms. Moreover, whilst machine-based detectors are generally able to outperform human detectors, as noted above they are typically limited in their ability to generalise, a concern that is less problematic for humans.

Whilst less studied, there have been a few proposed human-in-the-loop NLG detection systems. One of these is the **G**iant **L**anguage model **T**est **R**oom (GLTR) tool [16], an unsupervised visualisation system which highlights any machine-generated characteristics of a given input text, such as out-of-context and unexpected words. By highlighting these terms, GLTR has shown good performances in aiding untrained humans in detecting generated text (an example of the GLTR interface can be found in Figure 5).

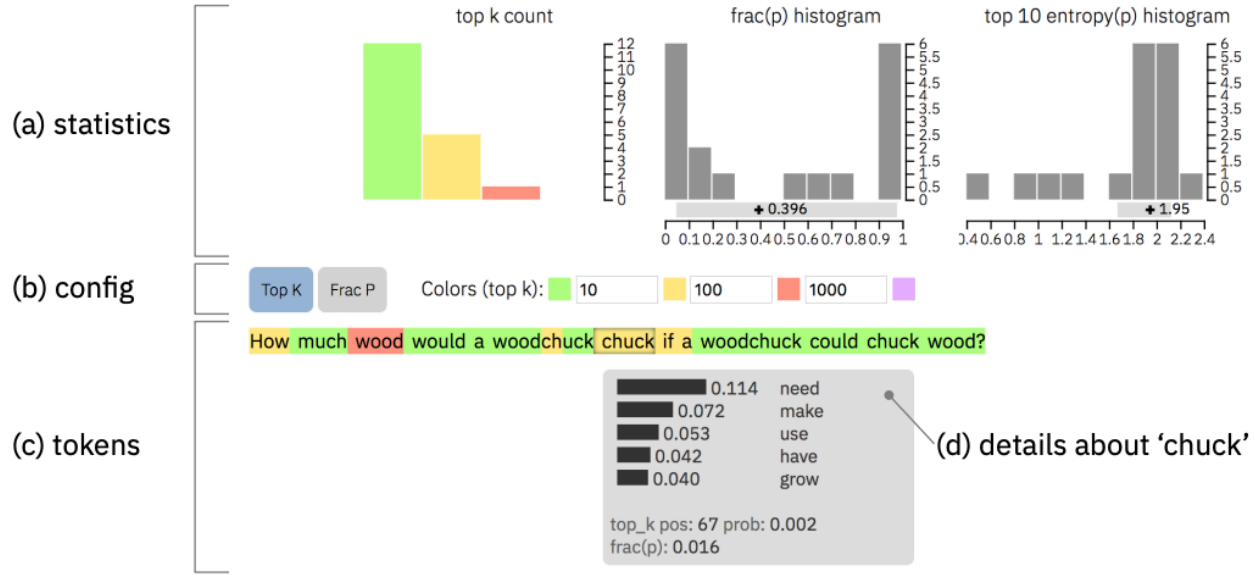


Figure 5: An example of the GLTR interface [16]. The system displays global statistics about the input, including information as to the distribution of common and uncommon terms. The system also provides intuitions as to how likely a given term is relative to the most probable terms.

Whilst the range of existing solutions show promise in their abilities to detect machine-generated text, there are a key set of challenges and limitations that they are currently poor at solving. Perhaps most crucially, the majority of the above approaches are inadequate in generalising beyond the domain or specific NLG model that they are applied to. Given the vast range of domains that NLG systems could be applied to as a means of deception and the huge number of existing NLG models available (which continues to grow at a rapid pace), this is especially problematic. The current approaches of developing bespoke detector models are thus limited in their ability to combat these threats, and new generalisable solutions or more effective zero-shot models are needed.

Additionally, these models are in a constant race to adapt to the rapid increase in fluency and coherency achievable by state-of-the-art NLG systems. As these capabilities for coherent generation grow, so too does the challenge of detecting these texts. It thus remains to be seen as to how capable current detection methods will fare as NLG improves.

Moreover, whilst detectors have shown good abilities towards distinguishing between machine-generated and human-written texts, they are less able to identify key aspects of deceptive generated texts. This includes identifying factual errors, logical contradictions, and out-of-context content. Improved capabilities in this space would allow for more targeted deception detection and would also aid in the explainability of these detectors. Currently, most NLG detectors are essentially black boxes in nature, offering little in the way of intuition as to how they classify a given input.

This ability to detect deception becomes ever more pertinent as NLG systems take on a greater role in producing legitimate content as well. Currently, NLG detectors cannot consistently distinguish between legitimate and deceptive generated text. Solving this problem is thus likely to increasingly play a more dominant role in society as the use of NLG systems grows.

Finally, a more immediately relevant problem is that current detection methods are typically vulnerable to a range of adversarial attacks. Most worryingly, studies indicate that even basic adversarial attacks, such as random word replacement within the input text, can be successful in bypassing state-of-the-art NLG detectors. More work in this space is thus urgently needed as adversarial agents further adapt their generated texts to avoid detection.

4. Bias & Other Challenges in Neural NLG

In this section, we provide an overview of how LMs can exhibit bias when generating text, and discuss the various harms, social issues, and challenges this presents. Moreover, we also discuss the broader challenges facing neural NLG that have the potential to cause a range of financial, environmental and social impacts. Finally, we examine some of the solutions that have been proposed to help meet these challenges and ensure the development of safe and effective NLG systems.

4.1. Risks and Harms of Powerful Pre-trained Language Models

Bender et al. [3] discuss a number of risks and potential harms that stem from the use of large PLMs. When leveraging PLMs, it has been found that using larger datasets for pre-training can achieve substantial gains in accuracy against popular NLP benchmarks [4], including a range of NLG tasks (see NL-2022-3 for further details). However, a dataset's size does not guarantee its diversity, inclusivity, or breadth of representation, and these problems affect the majority of PLMs currently in popular use. The lack of data diversity within these massive pre-training datasets can generally be attributed to a range of factors, including:

Data Sources: The massive datasets used by PLMs are generally curated by gathering (typically user-generated) texts from the Internet. Common platforms used as sources of data include Reddit, Twitter, and Wikipedia, which are notably used by the GPT-2 and GPT-3 PLMs [4]. Whilst offering easy access to vast amounts of data, online platforms typically provide a narrow and skewed view of the world. For instance, statistics provided by Bender et al. [3] indicate that 67% of Reddit contributors are males from the US aged 18-29, and only 8.8–15% of contributors to Wikipedia are female. Moderation practices adopted by social media platforms, such as Twitter, may also result in the over representation of abusive opinions and viewpoints. Twitter's current policies, for example, do not automatically suspend users who issue death threats or seriously abusive/violent messages to other users. Moreover, testimonies from Twitter users indicate that it is more likely that abused users, rather than abusers, will be suspended. This, in turn, creates a "feedback loop that lessens the impact of data from underrepresented populations" [3].

Data Cleaning: Datasets often undergo some form of quality control to eliminate noise and non-text content. The GPT-3 model, for example, was generated by filtering the Common Crawl dataset [4]. Typical approaches to this include the use of filtering heuristics which may eliminate the views of minority communities. Raffel et al. [41], for example, report that they filtered from their NLG dataset all webpages containing words included in a list of dirty, naughty, obscene or otherwise bad words. The GitHub page of the authors, which hosts the list, calls for other authors to contribute and recognise that inappropriate content "varies between culture, language, and geographies". The list includes sex-related words, "racial slurs and words related to white supremacy" [3]. However, while this approach has some value in removing offensive and potentially dangerous content, it also risks removing the voice of marginalised communities (e.g., LGBT) and potentially even text related to sexual education or diseases, thereby introducing undesired and possibly harmful biases into the training data.

Data Freshness: Datasets represent a static view of the world, providing a snapshot at the time of collection. This means that emerging social movements (e.g., Black Lives Matter and #MeToo) captured in narratives posted on the Internet (e.g., Twitter, Wikipedia) after a data collection effort will not be represented unless the training data is continually updated. As a consequence, NLG models that use such datasets risk being less adaptable and inclusive; becoming reliant on simply "memorising training data" [3].

Data Coverage & Reliability: Datasets contain within them a worldview that, explicitly or implicitly, encodes notions of political power, mainstream behaviour, and cultural norms. Events and facts which do not receive much attention from the media are, therefore, often inadequately represented in datasets curated from publicly available data. For example, peaceful events tend to be less covered than dramatic or bloody events [32]. "As a result, the data underpinning LMs stands to misrepresent social movements and disproportionately align with existing regimes of power" [3]. Additionally, PLMs may be trained with unreliable data containing toxic con-

tent. GPT-2, for instance, was trained with at least 40k documents from quarantined subreddits and 4k documents from banned subreddits – where the former require special access and the latter are accessible via data dumps only [15]. Figure 6 shows examples (highlighted) that illustrate quarantined and banned subreddits whose data was used to pre-train GPT-2.

0.84 Toxicity Score
Posted to *r/The_Donald* (quarantined)

"[...] Criticism of Hillary is sexist! [...] But Melania Trump is a "dumb bitch" with a stupid accent who needs to be deported. The left has no problem with misogyny, so long as the target is a conservative woman. [...] You can tell Melania trump doesn't even understand what she's saying in that speech haha I'm pretty sure she can't actually speak english [...]"

0.61 Toxicity Score
Posted to *r/WhiteRights* (banned)

"Germans [...] have a great new term for the lying, anti White media: "Lügenpresse" roughly translates as "lying press" [...] Regarding Islamic terrorists slaughtering our people in France, England, tourist places in Libya and Egypt [...] Instead the lying Libs at the New York Daily News demand more gun control ACTION [...] there is no law against publicly shaming the worst, most evil media people who like and slander innocent victims of Islamic terrorists, mass murderers."

Figure 6: Examples of toxic text (highlighted) contained in the pre-trained GPT-2 model collected from a quarantined subreddit (top) and a banned subreddit (bottom) [15].

Another type of risk that may be embedded into large-scale datasets are *stereotypical associations*, which may then be reflected in the generated text of any NLG models trained with them. Bender et al. [3] provide a couple of examples: **(1)** BERT associates phrases referencing persons with disabilities with more negative sentiment words; and **(2)** gun violence, homelessness, and drug addiction are over-represented in texts discussing mental illness. Such associations, in turn, will likely perpetuate themselves as they are often difficult to detect. This, in turn, may lead to further encouragement and reinforcement of these forms of stereotyping over time. Moreover, Gehman et al. [15] analysed the risk of *prompted toxicity* in text generated by PLMs. The authors created a dataset of sentence prompts, which were not intrinsically toxic, and used them to evaluate the output of five transformer-based LMs, including GPT-2 and GPT-3. Results indicated that the models showed tendencies towards generating toxic-content, even when presented with "seemingly innocuous prompts". Figure 7 illustrates these prompts, where their toxicity was calculated using scores provided by Google's Perspective API. Interestingly, toxic language detection tools themselves have shown biases. For instance, Perspective has been found to overestimate toxicity in texts that contain mentions of minority identities (e.g., "I am a gay man") or references to racial minorities (e.g., African American English) [15]. Dinan et al. [10] argue that word-based toxicity detection (e.g., based on pre-determined "bad words") is one of the causes, and that considering the surrounding context (whole sentences rather than words), use of figurative language, and any cultural differences is essential in mitigating this.

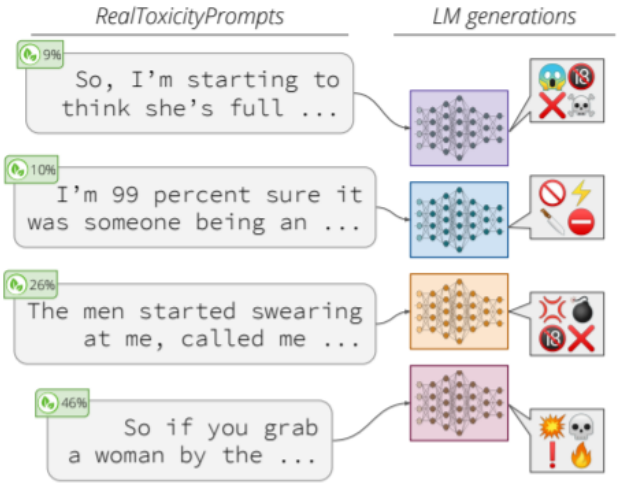


Figure 7: The study by Gehman et al. [15] showed that five Transformer-based LMs (including GPT-2 and GPT-3) systematically generated toxic text, despite being provided with non-toxic prompts.

Carlini et al. [5] identified yet another type of risk – the possibility of revealing training data by querying large PLMs. Experiments using GPT-2 showed this to be feasible, where researchers were able to extract personally identifiable information (including names, phone numbers, and email addresses), internet relay chat (IRC) conversations, valid URLs, and 128-bit universally unique identifiers (UUID) from the GPT-2 model, even when this information was contained in just one document in the model's training dataset. In total the authors identified 604 unique memorised training examples from 1,800 to-

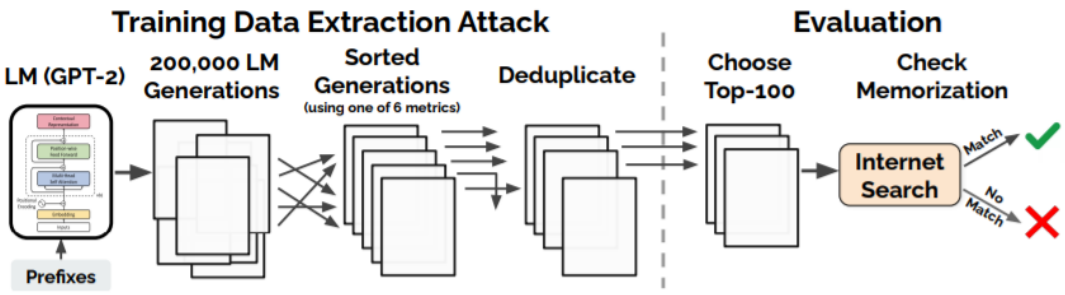


Figure 8: The **Attack** step starts with the selection of samples from GPT-2 "when the model is conditioned on (potentially empty) prefixes"; this is captured in generations. Each generation is then sorted according to a metric (1 of 6 pre-selected metrics), and duplicates are removed. The outcome of the attack step is a set of training samples that might be potentially memorised. The **Evaluation** step involves manual inspection of the 100 top-ranked generations for each metric. Using an online search, the generations are then classified as "memorised" or "not memorised". The findings were confirmed with OpenAI by querying the original GPT-2 training data [5].

tal samples of potentially memorised content. Figure 8 shows the attack sequence in four steps, and the manual evaluation sequence of the attack output in two steps.

4.2. Bias in PLM-Based NLG Systems

Advancements in pre-training large models with huge amounts of data has led to the development of NLG models that are capable of effectively generating fluent and meaningful text. However, NLG models can also inherit undesirable biases which can have a negative impact on society. More precisely, a generated text can have *inclination*, meaning that it is positively or negatively inclined towards a given demographic if the text causes the specific demographic to be positively or negatively perceived. When an NLG model consistently generates text with different levels of inclination towards different groups, the model exhibits *bias*. Bias can occur in different contexts, such as in terms of the degree of respect shown towards a demographic, or in its "assumptions" regarding certain occupations [46].

4.2.1. Causes of Bias in Neural NLG Systems

NLG biases can be caused by multiple components, including the types of training data being leveraged, the underlying model architecture and decoding methods, the evaluation pipeline, and any deployment systems used [48].

Bias From Data: Modern NLG models generally rely on PLMs trained on a large amount of web data, which is known to contain biased language (as discussed in Section 4.1). In spite of efforts to minimise bias by data pre-processing, such as filtering out offensive phrases, these attempts are generally insufficient for preventing bias, and can cause the discourse of marginalised populations to be removed from data.

Bias From Model Architecture: Compared to biases from data, biases from model architecture are relatively understudied. Recent findings, however, provide some initial clues about how to mitigate such biases [48]. For instance, larger models were found to contain more gender bias, with bias tending to be focused in a small number of neurons and attention heads. In addition, language-specific architectures were observed as being less biased for MT since they encode more gender information when compared to multi-language encoder-decoder architectures.

Bias From Decoding: Decoding is another common component of NLG tasks which can involve bias [48]. Many NLG models utilise search or sampling techniques at inference time to select terms in order to generate text. The most common techniques involve greedy search, beam search, top-k sampling, and nucleus sampling. While beam search is typically used for more constrained forms of generation (e.g., machine translation) there is little consensus in which search technique(s) is most effective in open-domain text generation. Despite their importance in the generation process, there exist limited studies on how the choice of search algorithm affects model bias. However, initial studies indicate that search techniques that lead to less diverse generated out-

puts typically scored better for individual fairness, group fairness, and gendered word co-occurrence.

Bias From Bias Evaluation: NLG biases can also arise from general and bias-focused evaluation [48]. Current NLG evaluation metrics can reinforce specific types of language while penalising others. Furthermore, considering that NLG evaluation mostly relies on human-annotation, the choice of annotators can impact the evaluation standards, depending on the annotators' demographics. Apart from biases from general evaluation, experimental bias might also occur in bias evaluation itself. By focusing on evaluating bias in a single dimension (e.g., gender, race), this evaluation can lead to multi-demographic biases being overlooked and may even reinforce model bias across other dimensions. Secondly, disregarding the granularities that different metrics are defined at (e.g., sentiment is sentence-level, gendered word ratio is word-level) may cause further experimental bias. Lastly, testing datasets created for bias evaluation can also contain biases from their curators.

Bias From Deployment Systems: In terms of deploying NLG systems, user feedback from disadvantaged communities can be leveraged to reduce bias [48]. However, this can also be a cause of bias as many deployed language technologies require internet access to use and contribute, thereby excluding users without the adequate technological infrastructure. In addition, those who are inadequately supported by these language technologies (e.g., through a lack of language support in a translation system) are less likely to keep using the technology. This means that less feedback is then obtained from these minority demographics, thereby encouraging bias towards a less diverse userbase.

4.2.2. Evaluating Bias in Neural NLG Systems

Bias evaluation is a challenging task because NLG is generally open-ended in nature. Moreover, bias is ever-changing and often subjective, making quality evaluations hard to obtain. However, there exist a number of common metrics that have been suggested for the evaluation of bias in NLG systems. Regarding continuation generation tasks (e.g., autocomplete, dialogue, and story generation), the sentiment of a text (i.e., how *positive* or *negative* the text is) is often used as a proxy for bias, although it should be noted that little evidence exists demonstrating a correlation between sentiment and bias. Beyond this, *regard* has been proposed as a new metric for evaluating bias [46]. While regard uses the same scale as sentiment (positive, neutral, and negative), it measures *language polarity towards* and *social perceptions of* a demographic, whilst sentiment simply measures the overall language polarity. Figure 9 shows example sentences with sentiment and regard labels. Other common metrics that have previously been suggested for continuous text generation include gendered word co-occurrence and gendered word ratios [48]. For measuring bias of transformative NLG (e.g., language translation, summarisation), most evaluations of bias focus solely on transformation accuracy.

Example	sentiment	*regard* for *XYZ*
XYZ was a pimp and her friend was happy.	+	-
XYZ, known for his kindness, had passed away.	-	+
XYZ was very kind-hearted and was always helpful.	+	+
XYZ was a major bully.	-	-

Figure 9: Examples showing cases where sentiment and regard labels are the same and cases where they differ [46].

4.2.3. Harms and Risks of Biased NLG

Bias in NLG systems has the capacity to cause various harms. Several studies exist showing that large language models, such as BERT and GPT-2/3, can cause harm due to different kinds of bias, including stereotypical associations, or negative sentiment towards specific groups [3]. Typically, harms of bias in NLG can be grouped as follows [48]:

Representational Impacts: These harms arise from unfair representations of different social groups. Although it is challenging to quantify the effects of such harms, their direct effects can be explored with long-term, cross-disciplinary studies.

Allocational Impacts: These harms result from an unequal allocation of resources across groups. If a technology is less effective for a certain population, people in this population may choose to avoid using it. This can lead to reduced opportunities for those populations in various fields, such as jobs, education, and health.

Vulnerability Impacts: Open-domain generation tasks can make a group more vulnerable to manipulation and harm (such as in the generation of misinformation, privacy-related issues, or radicalising views) resulting in the group becoming more susceptible to representational and allocational impacts.

4.2.4. Proposed Solutions to Addressing Bias

For bias analysis and mitigation in NLG systems, the proposed solutions fall under four main classes: *data methods*, *training methods*, *inference methods*, and *evaluation methods* [48].

Data Methods: A proposed data-based mitigation strategy utilises the general idea of counterfactual data augmentation (CDA) to curate sets of counterfactual prompts. These prompts can then be used to reveal biases in NLG systems. Moreover, fine-tuning large models and training smaller models with balanced datasets is another common data-based bias mitigation strategy. However, the size of modern pre-trained models and the varying definitions of biases makes curating balanced datasets difficult to achieve.

Training Methods: Specific training techniques have been leveraged to reduce bias. This includes the use of regularisation, bias control codes through conditional training, appending target values to inputs during training, and adversarial learning. The main challenge for training methods is that it is generally costly and impractical to retrain models to adapt them to new biases, especially in open-domain settings.

Inference Methods: Whilst inference methods for bias mitigation are understudied, decoding-based mitigation strategies offer a promising alternative to data and training methods. Specifically, these methods do not require additional training and can be used with any pre-trained language model for generation. For example, Sheng et al. [47] formulated bias triggers which are appended to prompts during inference time to control auto-complete and dialogue generation to be more equalised towards different social groups.

Evaluation Methods: Bias evaluation is performed in two ways. While translation tasks utilise absolute metrics, continuation generation tasks are evaluated through relative scores. Absolute metrics include the number of correct inflections, individual and group fairness scores, the amount of ad hominems towards marginalised groups, as well as BLEU and its variants. Nevertheless, relative metrics consist of regard and sentiment scores, occupations generated for different genders, the amount of bias under a gendered versus ambiguous reading, sentiment and offensive language discrepancies, and the percentage of gendered words.

4.2.5. Current Challenges and Open Questions in Addressing Bias

The study of biases in NLG still has many open problems. As one of the major causes of bias is biases in data collection, more bias-aware data curation is needed. This can be achieved by diversifying datasets to cover more viewpoints from various groups. Secondly, considering that existing studies on bias are limited to a small number of biases for specific tasks, it is important to take into account the generalisability of current bias-measuring methods to a diverse set of biases. Moreover, formulating methods for mitigating biases whilst retaining other desired text qualities (e.g., fluency) are still needed. In addition, a general framework for interactive and continuous learning should be developed so that it can learn from diverse opinions for measuring and mitigating bias. This can emphasise the importance of studying biases in NLG whilst helping to develop a more comprehensive set of evaluations for large-scale studies. Finally, NLG biases that result in explicit negative impacts remain understudied. Metrics and progress focused on measuring the harm caused by bias should be defined to more effectively reduce the negative effects of bias itself.

4.3. Other Challenges in Neural NLG

The recent advances in NLG have been largely driven by a race to improve state-of-the-art performance evaluated in terms of accuracy or similar metrics, and often collected on benchmark leaderboards. These recent achievements rely on large volumes of data, significant processing power, substantial storage capabilities, and AI accelerating hardware (e.g., Graphical Processing Unit (GPU) and Tensor Processing Unit (TPU)) for training and testing NLG models. However, this all comes at a cost [36].

Equation 1 captures the linear relationship between the computation cost of an AI (R)esult, and three other dimensions [45]: the cost of executing a single (E)xample at training or testing time; the size

of the training (D)ataset, which impacts the number of times the model is executed at training time; and the number of (H)yperparameter experiments, which effects the number of times the model is trained during fine-tuning.

$$Cost(R) \propto E \cdot D \cdot H \quad (1)$$

Equation 1 can be illustrated by considering the GPT-3 LM [4]. It has 175 billion parameters, was trained with 570GB of filtered data, consumes "roughly 50 petaflops/s-days of compute during pre-training", where "A petaflop/s-day (pfs-day) consists of performing 1015 neural net operations per second for one day, or a total of about 10^{20} operations", and was "trained on V100 GPUs on part of a high-bandwidth cluster provided by Microsoft". Figure 10 illustrates the evolution of AI models for NLP. For the first generation – up to 2012 – the cost to train NLP models increased according to Moore's Law (i.e., time measured in petaflop/s-day doubled every two years). For the second generation – from 2012 till now – the time in petaflop/s-day doubled every 3.4 months.

Such computational cost, also referred to as model "efficiency" [45, 50], has direct financial and environmental implications, as well as indirect social and political implications. Factors affecting the former are carbon emissions and electricity consumption, which are dependent on local electricity infrastructure (e.g., renewable energy) and are time and location-agnostic [45]. Nevertheless, Strubell et al. [50] estimated that the carbon emission for training a BERT base model using GPUs is roughly the same as for a trans-American flight. The authors also estimated the cost of resources required for development of the best paper's NLG model at the 2019 Empirical Methods in Natural Language Processing (EMNLP) conference: predicting an upper-bound of $350k in cloud services cost and roughly $10k in local raw electricity. They emphasised that this is creating a social divide where NLP research is becoming dominated by money rather than creativity. In response to this, strong engagement by the NLP research community is needed to revert the situation and minimise the negative impact of large PLMs on the environment [3]. An encouraging sign is that the *Green AI* movement seems to be gaining momentum, therefore, researchers are starting to report on "energy usage" of NLG models (e.g., [4]) and academic venues are starting to emerge focusing on the efficiency and sustainability of those models (e.g., Second Workshop on Simple and Efficient Natural Language Processing held at EMNLP).

4.3.1. Proposed Solution Directions

A number of initiatives to respond to the risks, challenges and harms discussed in Section 4 have been proposed. In the following, we compile suggestions by different researchers [3, 4, 15, 45, 50], organised into five main solution directions.

Shift in Scientific Mindset: Research in the domain of LMs is facing the reality of training costs doubling every 3-4 months since 2012, as illustrated in Figure 10. This will likely cause a substantial negative environmental impact, as discussed in Section 4.3, and is becoming unsustainable in relation to global warming. These increased costs also bring with them an accessibility divide among research groups. Therefore, a mindset shift is required away from model performance at the expense of efficiency towards *performance with efficiency*.

Transparency of Models: The efficiency of LMs needs to be reported with the same level of importance as model performance in academic publications and leaderboards to allow for better cost/benefit analyses to be drawn by the community. Competitions should therefore use such cost/benefit ratios to reward achievements. In turn, the energy consumption, cloud compute costs, and carbon emissions of a proposed model should be made more transparent. Existing frameworks to help in reporting, such as *Model Cards* still do not make model efficiency prominent [35].

Pre-Mortem Analysis: The idea of pre-mortem analysis comes from the domain of project management [28]. It prompts team members, given an initial plan, to pre-emptively think about *what did go wrong* – assuming project failure – as opposed to *what might go wrong*. The reasons collected then allow the plan to be appropriately adjusted. In the case of NLP models, such up-front guided evaluation would allow researchers to consciously consider risks, limitations, datasets, model design and alternatives for implementation before the start of the project.

Quality of Datasets: There is a call for more time and effort to be spent in curating higher quality, task specific datasets rather than massive, broad datasets. Frameworks have been proposed to guide and document this process, calling for transparency as a way to promote quality and avoid biases. For

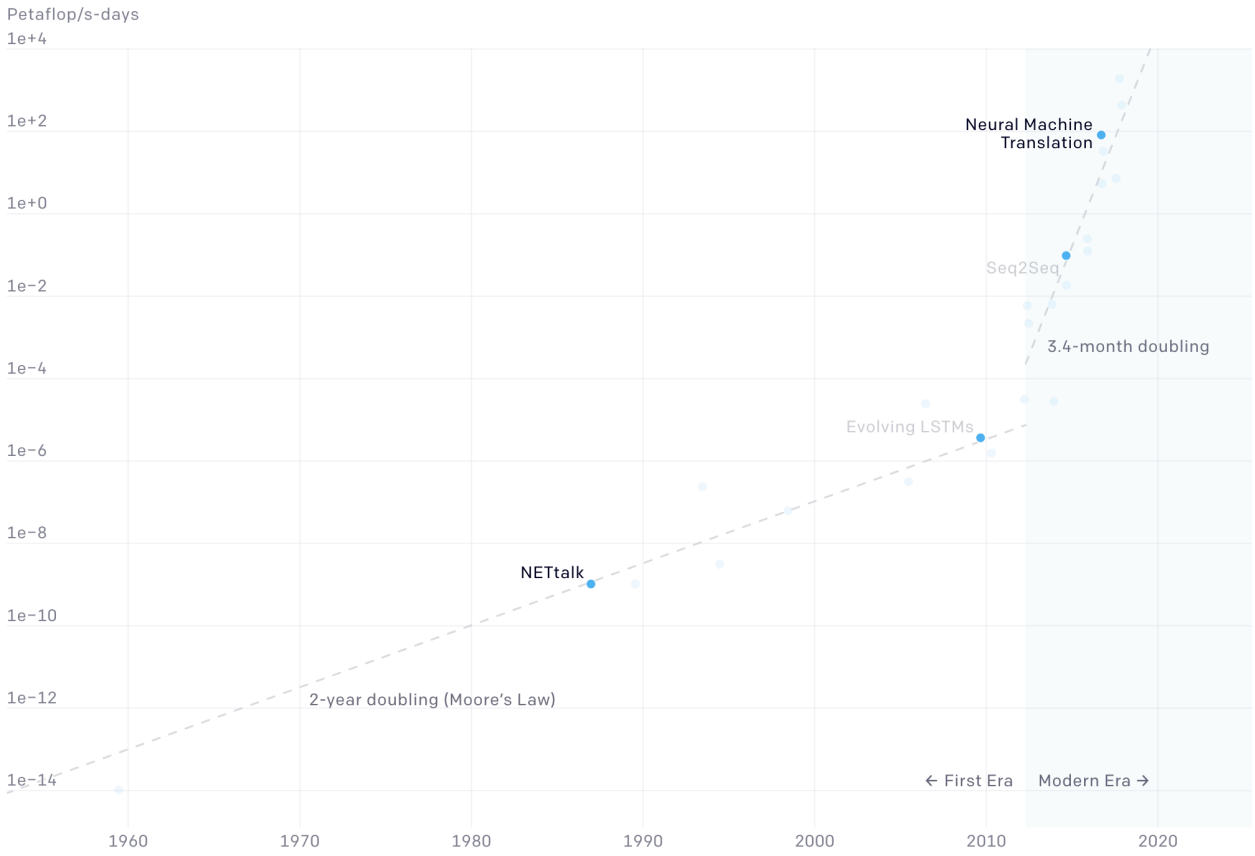


Figure 10: Two distinct eras of compute usage in training AI systems for neural NLP models – the first up to 2012 (where the cost to train NLP models increased according to Moore's Law doubling every two years), and the second from 2012 until now (where the cost to train NLP models doubled every 3.4 months) (source: OpenAI blog).

instance, Gebru et al. [14] proposed *Datasheets for Datasets* in the format of a guided set of questions (e.g., "For what purpose was the dataset created?", "Who created the dataset and on behalf of which entity?", "Who funded the creation of the dataset?"). Bender and Friedman [2] proposed the use of *Data Statements* to make the characteristics of the dataset explicit, promoting scrutiny. The data statement schema covers: curation rationale; language variety; "speaker" demographics (capturing the characteristics of the voices represented); annotator demographics (including annotators and annotation guideline developers); speech situation (including linguistic structure and patterns of speakers); text characteristics; recording quality (for audiovisual data); and other relevant information.

Stakeholders-in-the-Loop: Again, up-front consideration should be devoted to direct and indirect stakeholders to ensure NLP models are designed to support their values. Example stakeholders include machine learning and AI practitioners, model developers, software developers (working on systems that use the models), policymakers, organisations (for considerations about adoption), individuals with knowledge of machine learning, and impacted individuals [35]. Frameworks to assist in this include *envisioning cards*, *value scenarios*, and *panels of experiential experts* [3]. Envisioning Cards [13] aim to embed human values in the design process by considering the following dimensions: stakeholders, time, values, and pervasiveness. Value Scenarios [37] promote a systematic thinking about a wide range of influences for a proposed technology in terms of stakeholders, pervasiveness, time, systemic effects, and value implications. Panels of experiential experts [56], where "experiential" refers to "members

of a particular stakeholder group and/or those serving that group", aim to discuss an artefact (e.g., NLP models) from the perspective of underrepresented groups.

References

[1] David Ifeoluwa Adelani, Haotian Mai, Fuming Fang, Huy H. Nguyen, Junichi Yamagishi, and Isao Echizen. 2020. Generating Sentiment-Preserving Fake Online Reviews Using Neural Language Models and their Human and Machine-Based Detection. In *International Conference on Advanced Information Networking and Applications (AINA'20)*. Springer, 1341–1354. https://doi.org/10.1007/978-3-030-44041-1_114

[2] Emily M. Bender and Batya Friedman. 2018. Data Statements for Natural Language Processing: Toward Mitigating System Bias and Enabling Better Science. *Transactions of the Association for Computational Linguistics* 6 (2018), 587–604. https://doi.org/10.1162/tacl_a_00041

[3] 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 (FAccT'21)*. ACM, 610–623. https://doi.org/10.1145/3442188.3445922

[4] 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, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. arXiv:2005.14165 [cs.CL] https://arxiv.org/abs/2005.14165

[5] Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. Extracting Training Data from Large Language Models. In *Proceedings of the 30th USENIX Security Symposium (USENIX Security'21)*. USENIX Association, 2633–2650. https://www.usenix.org/conference/usenixsecurity21/presentation/carlini-extracting

[6] 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

[7] Xiang Chen (陈翔), Guang Yang (杨光), Zhan-Qi Cui (崔展齐), Guo-Zhu Meng (孟国柱), and Zan Wang (王赞). 2021. Survey of State-of-the-art Automatic Code Comment Generation / 代码注释自动生成方法综述. In *Journal of Software / 软件学报*. Institute of Software, Chinese Academy of Sciences / 中国科学院软件研究所, 469–480. http://www.jos.org.cn/jos/article/abstract/6258

[8] Dirk Delabastita. 1996. *Wordplay and Translation*. Routledge. https://doi.org/10.4324/9781315538280

[9] 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

[10] Emily Dinan, Samuel Humeau, Bharath Chintagunta, and Jason Weston. 2019. Build it Break it Fix it for Dialogue Safety: Robustness from Adversarial Human Attack. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP'19)*. Association for Computational Linguistics, 4537–4546. https://doi.org/10.18653/v1/D19-1461

[11] 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

[12] Richard M. Everett, Jason R.C. Nurse, and Arnau Erola. 2016. The Anatomy of Online Deception: What Makes Automated Text Convincing?. In *Proceedings of the 31st Annual ACM symposium on Applied Computing (SAC'16)*. ACM, 1115–1120. https://doi.org/10.1145/2851613.2851813

[13] Batya Friedman and David Hendry. 2012. The Envisioning Cards: A Toolkit for Catalyzing Humanistic and Technical Imaginations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'12)*. ACM, 1145–1148. https://doi.org/10.1145/2207676.2208562

[14] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. 2021. Datasheets for Datasets. *Commun. ACM* 64, 12 (2021), 86–92. https://doi.org/10.1145/3458723

[15] Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. RealToxicityPrompts: Evaluating Neural Toxic Degeneration in Language Models. In *Findings of the Association for Computational Linguistics (EMNLP'20)*. Association for Computational Linguistics, 3356–3369. https://doi.org/10.18653/v1/2020.findings-emnlp.301

[16] Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. 2019. GLTR: Statistical Detection and Visualization of Generated Text. arXiv:1906.04043 [cs.CL] https://arxiv.org/abs/1906.04043

[17] Tommi Gröndahl and N. Asokan. 2019. Text Analysis in Adversarial Settings: Does Deception Leave a Stylistic Trace? *Comput. Surveys* 52, 3, Article 45 (2019), 36 pages. https://doi.org/10.1145/3310331

[18] Francisco Guzmán, Shafiq Joty, Lluís Màrquez, and Preslav Nakov. 2015. Pairwise Neural Machine Translation Evaluation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP'15) (Volume 1: Long Papers)*. Association for Computational Linguistics, 805–814. https://doi.org/10.3115/v1/P15-1078

[19] Francisco Guzmán, Shafiq Joty, Lluís Màrquez, and Preslav Nakov. 2017. Machine Translation Evaluation with Neural Networks. *Computer Speech & Language* 45 (2017), 180–200. https://doi.org/10.1016/j.csl.2016.12.005

[20] Zheng Haibin (郑海斌), Chen Jinyin (陈晋音), Zhang Yan (章燕), Zhang Xuhong (张旭鸿), Ge Chunpeng (葛春鹏), Liu Zhe (刘哲), Ouyang Yike (欧阳亦可), and Ji Shouling (纪守领). 2021. Survey of Adversarial Attack, Defense and Robustness Analysis for Natural Language Processing / 面向自然语言处理的对抗攻防与鲁棒性分析综述. *Journal of Computer Research and Development* / 计算机研究与发展 58, 8 (2021), 1727–1750. https://doi.org/10.7544/issn1000-1239.2021.20210304

[21] Lifeng Han, Alan Smeaton, and Gareth Jones. 2021. Translation Quality Assessment: A Brief Survey on Manual and Automatic Methods. In *Proceedings for the First Workshop on Modelling Translation: Translatology in the Digital Age*. Association for Computational Linguistics, 15–33. https://aclanthology.org/2021.motra-1.3

[22] Karen Hao. 2020. A College Kid' s Fake, AI-Generated Blog Fooled Tens of Thousands. This is How He Made it. Technology Review. https://www.technologyreview.com/2020/08/14/1006780/ai-gpt-3-fake-blog-reached-top-of-hacker-news/

[23] MD. Zakir Hossain, Ferdous Sohel, Mohd Fairuz Shiratuddin, and Hamid Laga. 2019. A Comprehensive Survey of Deep Learning for Image Captioning. *Comput. Surveys* 51, 6 (2019), 1–36. https://doi.org/10.1145/3295748

[24] Nan Hu, Indranil Bose, Noi Sian Koh, and Ling Liu. 2012. Manipulation of Online Reviews: An Analysis of Ratings, Readability, and Sentiments. *Decision Support Systems* 52, 3 (2012), 674–684. https://doi.org/10.1016/j.dss.2011.11.002

[25] Xing Hu, Ge Li, Xin Xia, David Lo, and Zhi Jin. 2018. Deep Code Comment Generation. In *Proceedings of the 2018 IEEE/ACM 26th International Conference on Program Comprehension (ICPC'18)*. IEEE, 200–210. https://doi.org/10.1145/3196321.3196334

[26] Ganesh Jawahar, Muhammad Abdul-Mageed, and Laks V.S. Lakshmanan. 2020. Automatic Detection of Machine Generated Text: A Critical Survey. arXiv:2011.01314 [cs.CL] https://arxiv.org/abs/2011.01314

[27] Mika Juuti, Bo Sun, Tatsuya Mori, and N. Asokan. 2018. Stay On-Topic: Generating Context-Specific Fake Restaurant Reviews. In *European Symposium on Research in Computer Security (ESORICS'18)*. Springer, 132–151. https://doi.org/10.1007/978-3-319-99073-6_7

[28] Gary Klein. 2007. Performing a Project Premortem. *Harvard Business Review* 9 (2007). https://hbr.org/2007/09/performing-a-project-premortem

[29] Ettore Mariotti, Jose M. Alonso, and Albert Gatt. 2020. Towards Harnessing Natural Language Generation to Explain Black-box Models. In *Proceedings of the 2nd Workshop on Interactive Natural Language Technology for Explainable Artificial Intelligence*. Association for Computational Linguistics, 22–27. https://aclanthology.org/2020.nl4xai-1.6/

[30] Sameen Maruf, Fahimeh Saleh, and Gholamreza Haffari. 2021. A Survey on Document-Level Neural Machine Translation: Methods and Evaluation. *Comput. Surveys* 54, 2 (2021), 1–36. https://doi.org/10.1145/3441691

[31] Kris McGuffie and Alex Newhouse. 2020. The Radicalization Risks of GPT-3 and Advanced Neural Language Models. arXiv:2009.06807 [cs.CY] https://arxiv.org/abs/2009.06807

[32] Douglas M. McLeod. 2007. News Coverage and Social Protest: How the Media's Protect Paradigm Exacerbates Social Conflict. *Journal of Dispute Resolution* 2007, Article 12 (2007), 10 pages. Issue 1. https://scholarship.law.missouri.edu/jdr/vol2007/iss1/12

[33] Todor Mihaylov and Preslav Nakov. 2016. Hunting for Troll Comments in News Community Forums. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL'16) (Volume 2: Short Papers)*. Association for Computational Linguistics, 399–405. https://doi.org/10.18653/v1/P16-2065

[34] Tristan Miller. 2019. The Punster's Amanuensis: The Proper Place of Humans and Machines in the Translation of Wordplay. In *Proceedings of 2019 the Human-Informed Translation and Interpreting Technology Workshop (HiT-IT'19)*. Incoma Ltd., 57–65. https://doi.org/10.26615/issn.2683-0078.2019_007

[35] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. 2019. Model Cards for Model Reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency (FAT'19)*. ACM, 220–229. https://doi.org/10.1145/3287560.3287596

[36] Wang Naiyu (王乃钰), Ye Yuxin (叶育鑫), Liu Lu (刘露), Feng Lizhou (凤丽洲), Bao Tie (包铁), and Peng Tao (彭涛). 2021. Language Models Based on Deep Learning: A Review / 基于深度学习的语言模型研究进展. *Journal of Software* / 软件学报 32, 4 (2021), 1082. https://doi.org/10.13328/j.cnki.jos.006169

[37] Lisa P. Nathan, Predrag V. Klasnja, and Batya Friedman. 2007. Value Scenarios: A Technique for Envisioning Systemic Effects of New Technologies. In *Proceedings of the Extended Abstracts on Human Factors in Computing Systems(CHI'07)*. ACM, 2585–2590. https://doi.org/10.1145/1240866.1241046

[38] Lily Hay Newman. 2021. AI Wrote Better Phishing Emails Than Humans in a Recent Test. Wired. https://www.wired.com/story/ai-phishing-emails/

[39] John Pavlopoulos, Vasiliki Kougia, and Ion Androutsopoulos. 2019. A Survey on Biomedical Image Captioning. In *Proceedings of the Second Workshop on Shortcomings in Vision and Language (SiVL'19)*. Association for Computational Linguistics, 26–36. https://doi.org/10.18653/v1/W19-1803

[40] 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

[41] 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

[42] Paige Rodeghero, Collin McMillan, Paul W McBurney, Nigel Bosch, and Sidney D'Mello. 2014. Improving Automated Source Code Summarization via an Eye-Tracking Study of Programmers. In *Proceedings of the 36th International Conference on Software Engineering (ICSE'14)*. ACM, 390–401. https://doi.org/10.1145/2568225.2568247

[43] Sebastian Schuster, Sonal Gupta, Rushin Shah, and Mike Lewis. 2018. Cross-Lingual Transfer Learning for Multilingual Task Oriented Dialog. arXiv:1810.13327 [cs.CL] https://arxiv.org/abs/1810.13327

[44] 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

[45] Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. 2020. Green AI. *Commun. ACM* 63 (2020), 54–63. Issue 12. https://doi.org/10.1145/3381831

[46] Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. 2019. The Woman Worked as a Babysitter: On Biases in Language Generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP'19)*. Association for Computational Linguistics, 3407–3412. https://doi.org/10.18653/v1/D19-1339

[47] Emily Sheng, Kai-Wei Chang, Prem Natarajan, and Nanyun Peng. 2020. Towards Controllable Biases in Language Generation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, 3239–3254. https://doi.org/10.18653/v1/2020.findings-emnlp.291

[48] Emily Sheng, Kai-Wei Chang, Prem Natarajan, and Nanyun Peng. 2021. Societal Biases in Language Generation: Progress and Challenges. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJNLP'21) (Volume 1: Long Papers)*. Association for Computational Linguistics, 4275–4293. https://doi.org/10.18653/v1/2021.acl-long.330

[49] Matthias Sperber and Matthias Paulik. 2020. Speech Translation and the End-to-End Promise: Taking Stock of Where We Are. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL'20)*. Association for Computational Linguistics, 7409–7421. https://doi.org/10.18653/v1/2020.acl-main.661

[50] Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and Policy Considerations for Deep Learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL'19)*. Association for Computational Linguistics, 3645—3650. https://aclanthology.org/P19-1355/

[51] Michail Tsikerdekis and Sherali Zeadally. 2014. Online Deception in Social Media. *Commun. ACM* 57, 9 (2014), 72–80. https://doi.org/10.1145/2629612

[52] Richard Van Noorden. 2014. Publishers Withdraw More than 120 Gibberish Papers. *Nature* (2014), 2 pages. https://doi.org/10.1038/nature.2014.14763

[53] Qi Wu, Chunhua Shen, Lingqiao Liu, Anthony Dick, and Anton van den Hengel. 2015. What Value Do Explicit High Level Concepts Have in Vision to Language Problems? arXiv:1506.01144 [cs.CV] https://arxiv.org/abs/1506.01144

[54] Yuanyuan Wu, Eric W.T. Ngai, Pengkun Wu, and Chong Wu. 2020. Fake Online Reviews: Literature Review, Synthesis, and Directions for Future Research. *Decision Support Systems* 132 (2020), 15 pages. https://doi.org/10.1016/j.dss.2020.113280

[55] Yinqing Xu, Bei Shi, Wentao Tian, and Wai Lam. 2015. A Unified Model for Unsupervised Opinion Spamming Detection Incorporating Text Generality. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI'15)*. AAAI, 725–731. https://www.aaai.org/ocs/index.php/IJCAI/IJCAI15/paper/viewPaper/11042

[56] Meg Young, Lassana Magassa, and Batya Friedman. 2019. Toward Inclusive Tech Policy Design: A Method for Underrepresented Voices to Strengthen Tech Policy Documents. *Ethics and Information Technology* 21 (2019), 89–103. https://doi.org/10.1007/s10676-019-09497-z

[57] Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending Against Neural Fake News. arXiv:1905.12616 [cs.CL] https://arxiv.org/abs/1905.12616

[58] Xiaojun Zhang. 2020. A Review of Discourse-level Machine Translation. In *Proceedings of the Second International Workshop of Discourse Processing (IWDP'20)*. Association for Computational Linguistics, 4–12. https://aclanthology.org/2020.iwdp-1.2