Text Generation: A Systematic Literature Review of Tasks, Evaluation, and Challenges

JONAS BECKER*, University of Göttingen, Germany and LKA NRW, Germany
JAN PHILIP WAHLE, University of Göttingen, Germany
BELA GIPP, University of Göttingen, Germany
TERRY RUAS, University of Göttingen, Germany

Text generation has become more accessible than ever, and the growing interest in these systems, especially those using large language models, has spurred a surge in related publications. We provide a systematic literature review comprising 257 papers, covering the period from January 2017 to December 2025. This review categorizes text generation contributions into five main tasks: open-ended text generation, summarization, translation, paraphrasing, and question answering. For each task in our taxonomy, we review relevant characteristics and key subtasks. We assess current approaches for evaluating text generation systems, covering model-free, model-based, and human evaluation. Our investigation shows several task-specific challenges (e.g., missing datasets for multi-document summarization, lack of coherence in story generation, and difficulties in complex reasoning for question answering). We further discuss nine challenges common to all tasks and sub-tasks in recent text generation papers: bias, reasoning, hallucinations, misuse, privacy, interpretability, transparency, datasets, and computing. This systematic literature review targets two main audiences: early-career researchers in natural language processing seeking an overview of the field and promising research directions, and senior researchers who need a recent overview of the main tasks, evaluation, challenges, and mitigation strategies.

JAIR Track: Surveys

JAIR Associate Editor: Valerio Basile

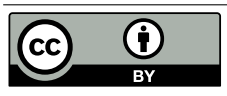
JAIR Reference Format:
Jonas Becker, Jan Philip Wahle, Bela Gipp, and Terry Ruas. 2026. Text Generation: A Systematic Literature Review of Tasks, Evaluation, and Challenges. *Journal of Artificial Intelligence Research* 86, Article 48 (August 2026), 49 pages. DOI: 10.1613/jair.1.22258

1 Introduction

Human language is a defining feature of human intelligence, enabling communication, reasoning, and shared understanding across individuals and time (Dennett, 2013; Premack, 2004). Text generation is a core task in natural language processing that involves producing coherent sequences of natural language tokens. Similar to how industrialization has automated numerous physical tasks and bodily routines, artificial intelligence (AI) systems are becoming human companions to solve cognitive problems and mental routine tasks (Ouyang et al., 2022; Achiam et al., 2024; Touvron et al., 2023; Comanici et al., 2025).

*Corresponding Author.

Authors' Contact Information: Jonas Becker, ORCID: 0009-0006-6438-1211, jonas.becker@uni-goettingen.de, University of Göttingen, Germany and LKA NRW, Düsseldorf, Germany; Jan Philip Wahle, ORCID: 0000-0002-2116-9767, wahle@uni-goettingen.de, University of Göttingen, Germany; Bela Gipp, ORCID: 0000-0001-6522-3019, gipp@uni-goettingen.de, University of Göttingen, Germany; Terry Ruas, ORCID: 0000-0002-9440-780X, ruas@uni-goettingen.de, University of Göttingen, Germany.



DOI: 10.1613/jair.1.22258

Major breakthroughs in AI have been achieved when models became capable of modeling natural language, specifically with large language models (LLMs), generating texts that are on par with human writing (Uchendu et al., 2021; Dou et al., 2022; Wahle et al., 2022a). Together with deep learning architectures, large-scale data, and accessible computing infrastructure, this has unveiled a new paradigm for training AI assistants at scale. Anyone with internet access can now have their own AI assistant to automate laborious, time-consuming tasks such as drafting emails, filling out forms, or developing software. This scenario is the result of sustained, collaborative, and incremental efforts by researchers and practitioners worldwide in AI, engineering, statistics, linguistics, and natural language processing (NLP), inter alia, over the years.

In the early days of distributional semantics (Harris, 1954), and rule-based question answering systems (McKeown, 1982), models would use structured data, grammatical rules, and templates to construct text (Winograd, 1972; McKeown, 1982). Today, most approaches use neural networks and backpropagation via gradient descent to estimate the probability of the next word in a sequence of words (Bengio et al., 2000). Modern LLMs like GPT-5 (Singh et al., 2025), LLaMA (Touvron et al., 2023), and Gemini (Comanici et al., 2025) solve problems in a text-to-text manner through natural language. This problem-solving method is inspired by how humans solve problems, formulating answers that convey a particular meaning, a process known as language production. In terms of machines, text generation is the process of creating natural-language text.

Text generation, also referred to as natural language generation (NLG), is present in a myriad of tasks, such as generating stories (Brown et al., 2020), engaging in dialogue (Li & Zhao, 2023), or solving reasoning problems (Comanici et al., 2025). Despite this rapid expansion, the literature on text generation remains fragmented across tasks, evaluation practices, and assumptions about what constitutes progress. As a result, contributions are often difficult to compare, open problems are inconsistently defined, and limitations are revisited in isolation rather than addressed systematically.

A systematic literature review is therefore essential not merely to summarize existing work, but to organize and contextualize contributions, identify recurring challenges and opportunities. By consolidating models, datasets, and research directions. Such a review can help establish shared understanding and guide future research toward more robust, interpretable, and responsible text generation.

In this paper, we provide an overview of recent text generation research up to December 2025, focusing on publications since the introduction of the Transformer architecture in 2017 (Vaswani et al., 2017), which has since become a central component of NLP. Our systematic literature review has three main focuses: tasks and sub-tasks (Section 3), evaluation metrics (Section 4), and challenges (Section 5). For this literature review, we exclude investigations of data-to-text and multimodal approaches (e.g., syntactic-semantic representations of images (Tian et al., 2024a) and tables (Ding & Xu, 2023)). This is done given the recent prominence of work on text-to-text tasks and the fact that evaluation and challenges inherently differ between text-to-text and data-to-text systems, exceeding the scope of this review. For multimodality, we refer to the surveys by Erdem et al. (2022); Barua et al. (2023). We devise the following key questions to organize our review:

(1) What constitutes the task of text generation? What are the main sub-tasks?
(2) How are text generation systems evaluated? What are the concomitant limitations?
(3) What are the open challenges in text generation?
(4) What are prominent research directions in text generation?

Figure 1 gives an overview of the most prominent tasks and associated challenges in text generation. We delimit text generation as the synthetic production of contextually relevant content in written natural language from an underlying non-linguistic representation. We identify five main tasks through our investigation: *open-ended text generation*, *summarization*, *translation*, *paraphrasing*, and *question answering* (Section 3). For each task, we review its relevant characteristics, sub-tasks, and specific challenges. Next, we assess commonly employed evaluation methodologies in the field (i.e., model-free and model-based metrics, LLM-as-a-judge, and human evaluation)

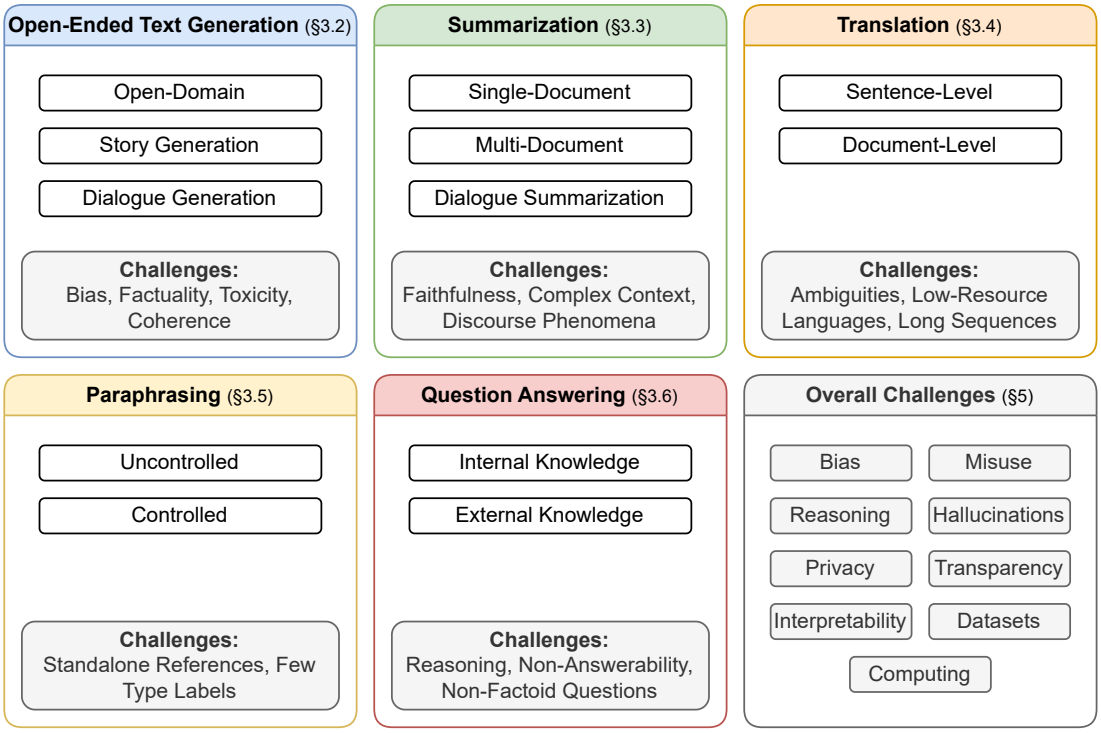


Fig. 1. Taxonomy of the main tasks in text generation. For each main task (e.g., translation), we identify common subtasks (e.g., sentence-level, document-level). We discuss the specific challenges of each task and the nine overall challenges in text generation.

and discuss their limitations (Section 4). In addition, we also identify nine prominent challenges common to all tasks and sub-tasks in recent text generation publications: *bias*, *reasoning*, *hallucinations*, *misuse*, *privacy*, *interpretability*, *transparency*, *datasets*, and *computing* (Section 5). Lastly, we revisit, summarize, and answer our research questions (Section 6). For reproducibility, we publicly share the details of our methodology (e.g., key phrases, subjective decisions, exclusion criteria, code) and metadata are publicly available[1].

1.1 Related Literature Reviews

Since 2017, the field of text generation has seen a marked increase in publications (Ramos et al., 2023). As a result, several works survey and organize the available literature. Most of these efforts focus on particular aspects of text generation, such as efficient Transformer architectures (Tay et al., 2020; Lin et al., 2021), adaptations of pre-trained models for unconditional text generation (Li et al., 2024a), the use of reinforcement learning for sequence-to-sequence models (Keneshloo et al., 2019), the use of external knowledge to improve the quality of generated text (Yu et al., 2022), and specific challenges (e.g., hallucination (Ji et al., 2023), bias (Sun et al., 2019; Gallegos et al., 2024), human evaluation (Clark et al., 2021), security (Yao et al., 2024)).

[1] https://github.com/jonas-becker/text-generation/

Only a few literature reviews provide a broad analysis of text generation. Gatt & Krahmer (2018a) survey the state-of-the-art in text generation in 2018, covering tasks, applications, and evaluation. Zhao et al. (2023) investigate the pre-training, fine-tuning, utilization, and evaluation of recent LLMs with more than ten billion parameters in selected tasks. Similarly, Raiaan et al. (2024) review the field of LLMs, covering a variety of NLP tasks including text generation. Fatima et al. (2022) survey 90 publications (2015-2021) on text generation using deep neural networks and organize their approaches, quality metrics, datasets, languages, and applications. Goyal et al. (2023) survey the technical application of text generation and concomitant challenges starting from the year 2011. The survey addresses table-to-text generation and multimodal tasks like image captioning.

Our literature review contributes to the existing literature by its recency, breadth of coverage across tasks, and systematic retrieval of papers. First, we include a large set of 257 resources resulting from subsequent exploration of the text generation field, covering the literature until December 2025. Second, unlike other literature reviews, such as Ji et al. (2023), we systematically select our primary publications and document each step in the selection process. Rather than choosing specific subfields preemptively (Li et al., 2022; Zhao et al., 2023), we adopt a bottom-up approach and infer the subtasks, evaluation procedures, and associated challenges in text generation through a systematic search, adding supplementary works at a later stage during writing. While other literature reviews focus on more technical characteristics (e.g., architectures (Goyal et al., 2023; Tay et al., 2020; Lin et al., 2021; Raiaan et al., 2024)), we provide a comprehensive investigation of text generation sub-tasks, their concomitant solutions, and remaining problems. Specifically, we differ from Goyal et al. (2023), who outline technical procedures for generating text and explain statistical methods, deep learning, transformer-based models, and encoding techniques. While they also mention some task-specific challenges, they do not provide a comprehensive overview focused on text-to-text tasks, including overall challenges of text generation as we do. We further add to their contribution by quantifying the impact and application of commonly employed evaluation metrics within the scientific community. Tay et al. (2020); Lin et al. (2021) exclusively review Transformers, whereas we focus specifically on text-generation sub-tasks and use a bottom-up approach to infer commonly employed techniques. Finally, we dedicate a specific analysis to open challenges in text generation that can be explored in the near future, thereby informing researchers in the field about which research gaps are most promising to pursue.

2 Methodology

This systematic literature review is inspired by the guidelines of Kitchenham & Charters (2007); Okoli (2015) and comprises three parts: *search and retrieval*, *automatic filtering*, and *manual assessment*. Figure 2 provides an overview of the process. We provide the source code for reproducing the literature retrieval on GitHub[1].

2.1 Search and Retrieval

Query construction. Our search queries use *primary* and *secondary* terms to select candidate papers. While primary terms aim to match papers in text generation, secondary terms focus our query on specific characteristics (e.g., training). As the two primary terms, we use *text generation* and *machine-generated text*. We generate secondary terms by manually inspecting existing field surveys and using suggestions from ChatGPT-4[2]. We manually review the list of search terms, merge overlapping terms, and conduct consensus voting among the authors in cases of ambiguity to finalize the selection. We consider the following 14 secondary terms: *paraphrase*, *nlp*, *natural language generation*, *language model*, *evaluation metrics*, *bias*, *privacy*, *controllable*, *creative*, *machine*, *automated*, *task*, *training*, *cost*, *co2 emission*, and *detection*.

Retrieval. We search Semantic Scholar for papers (including preprints from arXiv) by building all permutations of primary and secondary terms as queries, leaving us with 30 queries (e.g., "text generation evaluation metrics"). We

[2]version May 12, 2023

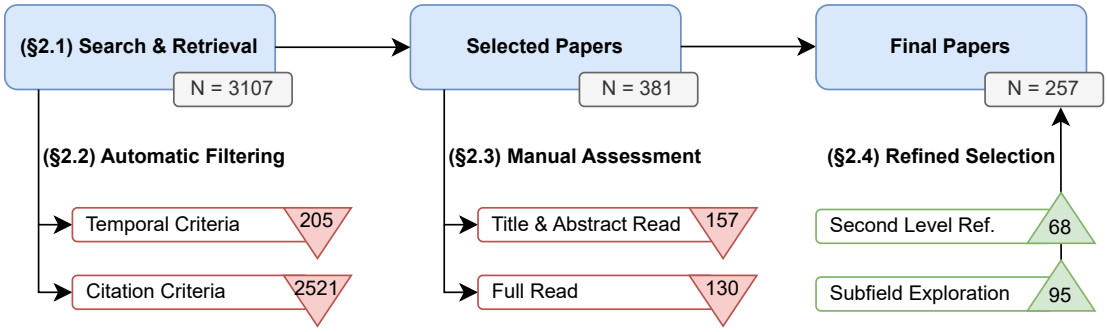


Fig. 2. Pipeline of this systematic review. Red-colored triangles show papers we excluded; Green-colored triangles show papers we added. We report the judgments on manually filtered papers in our GitHub repository.

restrict our search to Semantic Scholar's field of study to computer science to exclude articles on text generation in other disciplines (e.g., text generation tasks by humans in psychology studies)[3]. We retrieve the top 100 papers per query and year, ranked by Semantic Scholar's native ranking algorithm[4]. Semantic Scholar ranks search results using a learned relevance function that scores how well each document matches the query based on multiple signals, including semantic content, citation information, and metadata. The system combines an initial text-matching stage (e.g., keyword/semantic similarity) with a learned open-source reranker (s2search[5]) that reorders results to prioritize papers most likely relevant to the query. Our search results in 3107 publications published between 1966—2025.

2.2 Automatic Filtering

Temporal criteria. The Transformer architecture has fundamentally changed the landscape of text generation and now forms the basis of modern language models (Vaswani et al., 2017). While existing surveys predominantly examine approaches developed before the Transformer era (Gatt & Krahmer, 2018a) or focus on its early adoption (Fatima et al., 2022), this review considers work published since the Transformer's introduction in 2017. By this, we expect to retrieve works that directly or indirectly rely on the Transformer architecture. This cutoff is defined because most LLMs are Transformer-based, enabling a systematic analysis of the architecture's abilities, limitations, and emerging research opportunities.

Citation criteria. Next, we filter the results using Semantic Scholar's *influential citations* (Valenzuela-Escarcega et al., 2015). Influential citations, therefore, provide a proxy for measuring a publication's impact on other publications in the field. An influential citation traces key works that another work relies on, e.g., measured by citations in specific paper locations, research overlap, or similar work (Valenzuela-Escarcega et al., 2015). To even out the distribution of papers by year, we sample the top five papers per query and year by influential citations. We select papers with more than two influential citations until 2021 to estimate papers of strong influence in text generation. We treat recent works differently as older papers tend to have more citations than recent ones, although both can be impactful (Abdalla et al., 2023; Wahle et al., 2023a). Thus, we relax our restrictions and do

[3]According to Semantic Scholar, the Fields of Study attribute is assigned using a machine-learning classifier based on paper titles and abstracts; the S2FOS classifier achieved 86% correctness in an end-system human evaluation: https://medium.com/ai2-blog/announcing-s2fos-an-open-source-academic-field-of-study-classifier-9d2f641949e5

[4]https://api.semanticscholar.org/api-docs/

[5]https://github.com/allenai/s2search

not remove papers with two or fewer influential citations for the years 2022 to 2025. After the citation criteria, we are left with 381 works.

2.3 Manual Assessment

Title & Abstract Read. We manually assess the titles and abstracts of all 381 works from 2017 to 2025 to judge their relevance regarding text generation. We eliminate works that employ text generation within another field of study (e.g., medicine). Multimodality for text generation requires data-specific solutions (e.g., images (Tian et al., 2024a) and tables (Ding & Xu, 2023)). The focus of this review is text generation and tasks that can be solved through natural language. Therefore, we remove papers about multimodality in the process. We release a table of our relevance annotations to our GitHub repository[1].

Table 1. Number of papers per year after search and retrieval (Section 2.1), after automatic filtering (Section 2.2), and after refined selection (Section 2.4).

Number of papers after step...	**Prior**	**2017**	**2018**	**2019**	**2020**	**2021**	**2022**	**2023**	**2024**	**2025**	**Total**
§2.1 Search & Retrieval	205	65	113	188	334	438	268	52	1147	297	**3107**
§2.2 Automatic Filtering	0	47	64	48	48	41	41	48	29	15	**381**
§2.4 Refined Selection	34	12	20	26	37	27	28	42	18	13	**257**

Full Read. We read and summarize each relevant paper, considering its main contributions, applications, tasks, datasets, potential challenges, and limitations. The publications are tagged with the different main aspects of this review (e.g., dataset, metric) to facilitate their organization. We release a sampled table of our notes and tagged terms to our GitHub repository[1].

2.4 Refined Selection

Second Level References. To complement our review with other relevant work, we add auxiliary papers that are cited by the papers that undergo a full read. These auxiliary papers are direct references (second level). For example, this includes cases in which we find that method papers repeatedly use a dataset, architecture, or metric, indicating that the resource is relevant to the field and should be included in this literature review. We include 68 papers in the process.

Subfield Exploration. As we identify key areas of text generation, such as challenges, evaluation methods, and subtasks, we conduct a targeted search within these areas. This step serves two purposes. First, it complements citation-based retrieval by capturing recent advances in text generation that have not yet accumulated sufficient citations to be reliably surfaced by automated search engines such as Semantic Scholar. Second, it ensures comprehensive coverage across text-generation subtasks, evaluation, and challenges. After identifying core thematic areas, a targeted search facilitates the retrieval of relevant studies, such as architectural or task-specific analyses, that may not be captured by broad keyword-based queries alone. We note that the literature review adopts a bottom-up strategy, in which relevant tasks, evaluation methods, and challenges are systematically identified from earlier stages of the review process. This targeted subfield exploration is conducted within these inferred categories. For example, we search for the term "multi-document summarization" on scholarly platforms, such as Google Scholar, because this subtask was previously identified in the retrieved works. Ultimately, this step improves the recency, per-area coverage, and depth of our literature review. Adding 95 works in the process, our final selection of reviewed papers comprises 257 works as of December 2025.

3 Text Generation Tasks

Text generation, also called natural language generation, is a computational linguistic task that produces human-like text from a latent (underlying) non-linguistic representation of information (Mcdonald, 1980; Katz, 1980; Reiter & Dale, 1997).

In this paper, we treat text generation as a central capability and use it to organize the discussion of downstream tasks. Depending on the input and the intended behavior, text generation includes, for instance, generating text from structured data, rewriting existing text (Radev et al., 2002), producing responses in interactive settings (Li & Zhao, 2023), and generating domain-specific text such as code (Roziere et al., 2022).

Text generation is relevant beyond its technical role. It can lower barriers to accessing information and interacting with technology, including for users with higher learning curves (Naumanen & Tukiainen, 2007), and it can support domain work by turning specialized texts into more accessible forms (Dagdelen et al., 2024). At the same time, the same capability can be misused to generate convincing misinformation at scale, for example via social media bots (Zellers et al., 2020). Whether text generation is considered an opportunity, a risk, or both, understanding the recent developments in this field is of the utmost importance. Therefore, we define the problem of text generation, its most prominent sub-tasks, concomitant characteristics, and open challenges.

3.1 Problem

Katz (1980) was one of the first to explore text generation as a process of grouping different feature representations for generating kernel sentences, deciding on transformations based on syntax and themes, and then combining these kernels with appropriate modifications and pronoun adjustments to produce cohesive text. Later, Reiter & Dale (1997) expanded the notion of text generation to signals other than text, such as weather forecast data from graphical weather maps (Goldberg et al., 1994) or the creation of technical documentation via a knowledge base (Reiter et al., 1995). Modern definitions advocate for a process that leverages knowledge from computational linguistics and artificial intelligence to automatically generate natural, cohesive, and plausible texts that meet defined requirements (Li et al., 2022). In our literature review, we delimit text generation as the synthetic production of contextually relevant content in written natural language using an underlying non-linguistic representation. While the input to a text generation system during training might be natural language, it is not directly converted to a textual output. The output is derived from an underlying non-linguistic representation (e.g., embeddings) displaying meaning.

At the time of writing, text generation is often performed via LMs (Li et al., 2022; Brown et al., 2020). LMs provide a probabilistic description of natural language and maximize the probability of the generated sequence, producing the most probable text relevant to the context (Bengio et al., 2000; Jurafsky & Martin, 2024). Language modeling and text generation are distinct yet interconnected fields. While text generation focuses on producing content, language modeling leverages the probabilistic structure of language. However, text generation does not necessarily require LMs, rule-based approaches from Katz (1980) and non-autoregressive methods (Xiao et al., 2023) can also generate text.

Our literature review identifies five prominent areas related to text generation, namely *open-ended text generation*, *summarization*, *translation*, *paraphrasing*, and *question answering* (Figure 1). Open-ended text generation is a task where newly generated text is iteratively conditioned on the previous context so that the final output appears coherent and fluent (Li et al., 2023b; Venkatraman et al., 2025). This also included modern, instruction-tuned AI assistants such as ChatGPT (Achiam et al., 2024; Ouyang et al., 2022). Summarization is the generation of a text from one or more texts to convey information in a shorter format (Radev et al., 2002). Translation means converting a source text in language A to a target language B (Lewis et al., 2020a). Paraphrasing is the task of generating text that has (approximately) identical meaning but uses different words or structures (Dolan & Brockett, 2005; Vila et al., 2015; Wahle et al., 2023b). Question answering takes a question as input text and

outputs a streamlined answer or a list of possible answers based on background knowledge (Cai et al., 2020). We investigate these areas based on their specific subtasks and challenges. We mention relevant datasets selectively but point to our GitHub repository for an elaborate list of available corpora[1].

3.2 Open-Ended Text Generation

Task definition. Open-Ended Text Generation (OETG) generates new text constrained by an existing source text (e.g., prompt) so that the final output appears coherent and fluent (Li et al., 2023b; He et al., 2023). Applications of OETG range from open-domain response generation by AI assistants (Achiam et al., 2024) to complex story generation (Liang et al., 2023).

Main characteristics. Compared to other text generation tasks like summarization, OETG provides more freedom during content generation due to the highly variable output length. For this, LMs that maximize the probability distribution over word sequences are used to generate fluent text. Conversely, human language mainly encompasses semantic and pragmatic characteristics beyond grammatical and syntactic rules (Gehrmann et al., 2019). Nevertheless, the fluency of machine-generated text can mislead human judgments of authorship (Adelani et al., 2020). We note that some of the following tasks and sub-tasks may also be considered OETG in an LLM-prompted setup. Previous advances in instruction-following (Ouyang et al., 2022) have led to greater task unification under general-purpose systems like ChatGPT (Achiam et al., 2024). We will not reiterate the possibility of prompting an OETG system like ChatGPT (Achiam et al., 2024) for each task. This is crucial for this review, as we focus on the task-specific modeling, datasets, and challenges.

Challenges. Since the modern days of LMs (Brown et al., 2020; Devlin et al., 2019) and their capacity to produce fluent texts, other desired features emerged for OETG. As sequences generated by OETG systems can grow large, the risk of producing logical inconsistencies increases. This makes coherence a key challenge to solve in OETG (Guan et al., 2021; Liang et al., 2023). In addition, recent LMs are subject to bias, prejudice, and toxicity from human language. Thus, controlling attributes like sentiment (Kawamae, 2023; Adelani et al., 2020; Liu et al., 2021) during OETG is an active research topic (Yang et al., 2023a).

3.2.1 Sub-tasks and Datasets in Open-Ended Text Generation.

We identify three popular sub-tasks of OETG, namely *open-domain* OETG, *story generation*, and *dialogue generation*. Niche sub-tasks, such as review generation (Costa et al., 2018; Ni & McAuley, 2018) and data-to-document generation (Lu et al., 2023a), often use open-domain OETG methods because they generalize well to these problems.

Open-domain:

Description. Given minimal or underspecified input, the model generates coherent, contextually appropriate text across a wide range of subjects and scenarios, allowing for high variability in content, style, and structure. The task is not constrained to a predefined topic, format, or domain.

Modeling. Open-domain OETG differs from traditional NLP tasks in that it is not defined by a fixed domain or output structure, but rather by the system's ability to generate coherent and appropriate text under specified conditions (Li et al., 2023c). Thus, open-domain OETG systems must generalize not only across topics and domains but also across tasks, discourse structures, and styles, which are provided implicitly by natural-language prompts rather than by explicit task specifications. To achieve this, the dominant approaches for open-domain OETG use LLMs like GPT (Achiam et al., 2024; Brown et al., 2020; Radford et al., 2019) or LLaMA (Touvron et al., 2023), which support a wide range of downstream objectives through natural language prompting. This includes applications such as ChatGPT (Achiam et al., 2024; Ouyang et al., 2022), where users prompt the system to solve a variety of tasks (e.g., instruction following (Ouyang et al., 2022), question generation (Mulla & Gharpure, 2023)). Prompting reframes diverse tasks as instances of conditional text generation, in which instructions serve as soft constraints rather than fixed optimization targets, thereby avoiding the need for task-specific architectures or supervision.

Improvements in open-domain OETG are largely attributed to training strategies, including large-scale pre-training on diverse data, instruction tuning, and alignment to human preferences, rather than to the introduction of task-specific model architectures (Ouyang et al., 2022). Large-scale pre-training on datasets such as Common Crawl[6] (Brown et al., 2020; Touvron et al., 2023) or knowledge bases (Yu et al., 2022) improves coverage across domains and formats, but does not guarantee correctness, particularly in rare or complex cases. Thus, evaluation plays a central role in open-domain OETG. To evaluate pre-trained LLMs for OETG, He et al. (2025) provide an error-annotated dataset and benchmark tasks for text generation. Their benchmark covers errors from a taxonomy of 24 error types, organized into main and subtypes in linguistics and knowledge. Frequent errors of generated text are commonsense errors, inappropriate combinations, missing words, discourse errors, and redundancies. Open-domain OETG inherently trades increased coverage for greater exposure to rare and complex cases, where errors are more likely to arise (Liu et al., 2022). Consequently, effective open-domain OETG systems must balance broad generalization with mechanisms that promote linguistic and factual correctness.

Challenges of open-domain OETG. Open-domain OETG requires balancing multiple competing objectives—such as coherence, relevance, factuality, and diversity—that are often in tension with one another (Holtzman et al., 2020). Improvements along one dimension (e.g., diversity) may degrade others (e.g., coherence or factual accuracy), complicating both modeling and evaluation. Contrastive decoding selects each next token by favoring what a strong model prefers over what a weaker "generic" model prefers, steering generation away from bland high-frequency continuations (Li et al., 2023c). This usually yields more specific, less repetitive text. Technology companies often propose the most capable open-domain models as they have the resources to train them on large amounts of data (Abdalla et al., 2023). Many of these models are closed-source. This makes the creation and reproducibility of modern LLMs particularly challenging, as many companies tend not to disclose technical details about their models, such as used data or hyperparameters (Achiam et al., 2024). Few companies follow an open-source approach when it comes to LLM development (Touvron et al., 2023). This can hinder the scientific study of OETG, as these systems are difficult to reproduce and might not generalize to widely available open-source models. We note that open-domain OETG is also challenged by biased datasets, interpretability issues, limited evaluation, and other factors. As these challenges are not exclusive to OETG and are prevalent across several text generation tasks, we address them generally in Section 4 and Section 5.

Story Generation:

Description. Story generation is a conditional text generation task that requires producing narrative text that is coherent, logically consistent, and engaging, typically involving characters, events, and a meaningful temporal progression for the reader (Guan et al., 2021; Liang et al., 2023).

Modeling. We find that most approaches employ an LM with additional modules to enhance planning, temporal consistency, or coherence. Unlike short-form text generation, story generation requires maintaining a consistent narrative structure, character development, and causal progression over extended contexts. Thus, several works explicitly introduce planning mechanisms. Planning the generation process hierarchically can enhance perceived creativity (Peng, 2022). To achieve planning, Yang et al. (2022b) employ a module that produces a story premise with a setting, characters, and outline. Additionally, they use a dedicated module to make local edits, ensuring that the generated story remains consistent. Other approaches focus on improving coherence through prompt construction and conditioning. Fan et al. (2018) use a convolutional LM to generate a specially designed prompt and enhance the consistency of story generation with a text-to-text model. Their work also introduces the WritingPrompts dataset, which contains over 300k human-written prompt–story pairs. CollabStory uses several LLMs that collaborate at the segment level to form a coherent storyline (Venkatraman et al., 2025).

Challenges of story generation. The production of long and coherent stories with a consistent narrative and developing characters (i.e., unique and changing properties over the temporal shift in the story) is one of the

[6]https://commoncrawl.org/

main challenges in story generation (Alabdulkarim et al., 2021). Commonly employed decoding algorithms like beam search produce incoherent and repetitive text for story generation (Wiher et al., 2022). While effective for constrained tasks (e.g., translation), beam search performs poorly in open-ended story generation where many plausible continuations exist. This is because beam search maximizes sequence likelihood, favoring high-probability and generic continuations. In open-ended story generation, this can lead to degeneration phenomena such as repetition loops and loss of global coherence. To mitigate this, Lu et al. (2022b) propose a decoding algorithm with lookahead heuristics that enables foresight planning. This requires that the constraints of the conditioned story can be formulated as logical expressions.

Dialogue generation:

Description. Dialogue generation is a multi-turn conditional text generation task of modeling verbal or written communication among two or more participants, producing contextually appropriate and coherent utterances in a conversational exchange (Zhang et al., 2020d; Li & Zhao, 2023). While a two-party setup (i.e., training data and architecture) is often used in AI assistants, a multi-party setup is more common in, e.g., movie dialogue.

Modeling. Dialogue contains one-to-many relationships, where a single dialogue context may correspond to multiple appropriate replies. As bidirectional LMs (e.g., BERT (Devlin et al., 2019)) and autoregressive LMs (e.g., GPT (Achiam et al., 2024; Brown et al., 2020; Radford et al., 2019)) are not directly optimized on dialogue datasets, and the one-to-many relationships make fine-tuning inherently difficult, dialogue generation is modeled differently than open-domain OETG or story generation. Bao et al. (2020) and Zhang et al. (2020c) avoid fine-tuning by further pre-training the LM on conversational data. They jointly integrate uni- and bidirectional processing and model the one-to-many relationships with a latent variable. For pre-training and testing conversational models, mostly synthetic data exists. For example, the AMI corpus (Mccowan et al., 2005) comprises 100 hours of transcribed meeting dialogue, partly real and mostly staged between actors. The proposal of more authentic, human-authored dialogue at a sufficiently large scale could be a meaningful contribution to the field of dialogue generation.

Challenges of dialogue generation. We identify key challenges on the technical level and the availability of datasets. Both two- and multi-party dialogue generation are challenging tasks due to semantics, consistency, and interactivity when applied to open-domain problems (Huang et al., 2020a). Huang et al. (2020a) identify that these challenges are influenced by long context, character personalities, sentiment, and dialogue policies. Bingli & Vargas (2025) study persona-driven dialogues inspired by role-playing games, using a card-based framework that includes scene settings and character personas. When guided by detailed scene settings, their small model (seven billion parameters) could craft context-aware dialogues for even unseen characters and scenarios. While research on two-party dialogue is common, multi-party dialogue generation remains underexplored (Kann et al., 2022), possibly because natural multi-party dialogue often occurs in confidential settings, such as business meetings. This makes the available datasets for training or evaluating multi-party dialogue generation scarce. Investigating the dynamics of multi-party dialogue could support the understanding of the dynamics behind turn-taking (Skantze, 2021) and multi-agent interaction (Wang et al., 2023b; Becker et al., 2025a).

3.3 Summarization

Task definition. Summarization describes the generation of a short text, distilling one or more references (Radev et al., 2002). These references are usually more than double the length of the produced summaries (Radev et al., 2002). Many works differentiate between *extractive*, *abstractive*, and *hybrid* summarization (Fabbri et al., 2021; Bražinskas et al., 2020; Taunk et al., 2023). Extractive summarization concatenates spans of the input text to produce the final output text (Moratanch & Chitrakala, 2017). Abstractive summarization generates new phrases and sentences that can differ from the original text (Zhang et al., 2020a). Moreover, hybrid approaches aim to combine the benefits of both extractive and abstractive methods (Taunk et al., 2023).

Main characteristics. The characteristics of the task are different when looking at either extractive or abstractive summarization. Extractive systems use statistics to identify relevant text and are usually simple to implement, e.g., by fine-tuning an LM to select relevant text snippets (Liu et al., 2020) or ranking sentences from a source (Xiao & Carenini, 2019). Abstractive systems, conversely, use semantic features like word embeddings directly (Zhang et al., 2020a) to determine the meaning of texts and summarize them accordingly. Hence, abstractive summarization can generate text with low lexical overlap with the source, whereas extractive summarization always has overlap in the extracted text spans (Paulus et al., 2017; Zhang et al., 2020a).
Challenges. Extractive summarization often produces duplicated information and inconsistent summaries sensitive to the statistical features of the source text (Munot & S. Govilkar, 2014) while abstractive methods suffer from large contexts when summarizing large documents (Gambhir & Gupta, 2017). In addition, abstractive systems lack faithfulness, often hallucinating (Maynez et al., 2020).

3.3.1 Sub-tasks and Datasets in Summarization.

According to our investigations, the most popular summarization sub-tasks are *single-document summarization*, *multi-document summarization*, and *dialogue summarization*.

Single-Document Summarization (SDS):

Description: SDS is a sequence-to-sequence text generation task that aims to produce an accurate and concise summary from a single input document, e.g., a news article (Zhang et al., 2019, 2020a).
Modeling. Extractive SDS can be performed using encoders such as BERT, a bidirectional masked LM (Liu & Lapata, 2019). Copy mechanisms employed by a pointer-generator network leverage relevant tokens from the source as part of the output (Kumar et al., 2019). A decoder equipped with a pointer-generator network makes a soft choice at each decoding step between copying from the source and generating from the vocabulary (Zhang et al., 2019). This way, the encoder can predict out-of-vocabulary words by selecting appropriate tokens from the source text. Xiao & Carenini (2019) show that separately accounting for local and global context can improve summary quality. Abstractive summarization has seen a surge of techniques due to its ability to generate lexically independent summaries (Wang et al., 2018; Zhang et al., 2019). Some efforts combine extractive and abstractive methods for summarization to address the long computation times associated with purely abstractive approaches for long documents (Taunk et al., 2023). Today, LLMs are used for abstractive summarization based on a query or topic (Yang et al., 2023b). Prompted LLMs can match the performance of traditional fine-tuning approaches, but the possibility of factual errors or biased summaries exists. Meanwhile, LLMs can adapt their output to different target audiences (e.g., expertise, writing style), but still lack the stylistic diversity of human-authored text (Liu & Demberg, 2023).

Most SDS datasets we identify originate from the news domain (Narayan et al., 2018; Nallapati et al., 2016). This may be due to the fact that these datasets can conveniently leverage headlines or first sentences from news websites as summaries (Narayan et al., 2018) and are thus comparatively easy to create. For example, the XSum dataset (Narayan et al., 2018) includes 227k single-sentence summaries of BBC articles. The CNN/Daily Mail dataset (Nallapati et al., 2016) is a corpus with 312k multi-sentence summaries obtained by concatenating human summary bullets. The reliance on the news domain yields data that is genre-specific and not necessarily representative of an appropriate summary for other domains.
Challenges of SDS. Very long documents are inherently more challenging to summarize than shorter ones, as the relevance of more text sections needs to be taken into account. The quadratic time and memory complexity of commonly employed Transformers hinders their direct use for the task (Vaswani et al., 2017). This limitation is mitigated by specialized architectures like Longformer (Beltagy et al., 2020), which employ a linearly scaling attention mechanism. Besides this, abstractive summarization systems often lack faithfulness to the source text and tend to hallucinate information (Maynez et al., 2020). Some work addresses the faithfulness of pretrained LMs for document-grounded text generation (e.g., summarization), which we cover in Section 5.3.

Multi-Document Summarization (MDS):
Description. MDS is a sequence-to-sequence text generation task in which a model synthesizes information from multiple topic-related documents, often originating from different sources, to produce a single coherent and non-redundant executive summary (Ma et al., 2023).
Modeling. Common approaches for MDS specify distinct mechanisms to select or attend to relevant documents in a large collection. Maynez et al. (2023) identify several of these strategies for architecture design, ranging from ensemble networks or hierarchical approaches to graph neural networks or fine-tuning of pre-trained LMs. Liu et al. (2019) represent cross-document relationships via an attention mechanism. If faced with many (hundreds or more) documents, an LM can be employed with retrieval augmented generation (RAG) to combine document vectorization with retrieval of relevant text blocks (Lewis et al., 2020b). We find only few datasets for MDS. The Multi-News dataset (Fabbri et al., 2019) provides 46k news articles with an average number of 3.5 documents per summarized cluster. DiverseSumm includes 245 news stories, with each story comprising 10 news articles (Huang et al., 2024).
Challenges of MDS. MDS suffers from similar challenges as SDS but comes with an additional set of issues. Cross-document relations and conflicting, duplicate, and complementary information increase the challenges of mapping similar entities, events, and other related information (Ma et al., 2023; Gambhir & Gupta, 2017). Modern LLMs often hallucinate information in multi-document settings (Belém et al., 2025). They are also limited in their coverage of information across documents (Huang et al., 2024). We also find that the diversity of available MDS datasets is low, possibly due to the costs of creating them.
Dialogue Summarization:
Description. Dialogue summarization is a sequence-to-sequence text generation task that condenses a multi-party, multi-turn dialogue into a concise and informative summary while preserving key decisions, intents, and salient points (Kirstein et al., 2025). This sub-task is also important for meeting summarization goals (Feng et al., 2021), as it can reduce costs and time spent on manual note-taking.
Modeling. Dialogue contains one-to-many relationships, where a single dialogue context may correspond to multiple appropriate replies. Zhu et al. (2020) capture the unique dialogue structure by a hierarchical approach. They employ a word-level Transformer that processes the sequence of a single turn, while a turn-level Transformer processes the information of all turns in a meeting. Feng et al. (2021) leverage a relational graph encoder to model discourse relations. Mixture of Experts is with role-oriented routing via an LLM-based module to summarize a dialogue, with each expert processing different information from the dialogue (Tian et al., 2024b). This avoids anticipation bias and the potential loss of information inherent to a standard single-LLM approach. Most datasets for dialogue summarization consist of synthetic data. The AMI corpus (Mccowan et al., 2005) comprises 279 hours of synthetic meeting dialogue textualized from speech with annotations of extractive and abstractive summaries. Its participants play different roles in a team responsible for hardware design. The SAMSum corpus (Gliwa et al., 2019) includes 16k chat dialogues paired with human-written abstractive summaries. The dialogues are deliberately created by linguists, and the summaries are written in the third person.
Challenges of dialogue summarization. Dialogue summarization is difficult because of discourse phenomena like coreference errors (e.g., ambiguous pronouns) or discourse link errors (e.g., temporal ordering) (Pagnoni et al., 2021).This is why researchers use distinct architectures to account for these discourse phenomena, such as relational graph encoders (Feng et al., 2021) or hierarchical processing (Zhu et al., 2020). We identify that datasets in dialogue and meeting summarization are scarce, especially when looking for non-synthetic data. We suppose that this is because of confidentiality issues between the participants and the discussed topics. Training and evaluation on synthetic data could cause unexpected problems, such as a performance drop in real conversations, because naturally occurring dialogue might differ structurally or linguistically.

3.4 Translation

Task definition. Translation, also referred to as machine translation, is a task in which a (textual) source in language A is converted into a target language B (Lewis et al., 2020a).
Main characteristics. Unlike many natural language processing tasks, translation requires modeling two languages simultaneously and learning systematic correspondences between them. The publications identified by our search concern text-to-text problems (e.g., German-to-English translation).
Challenges. One of the main challenges in translation is to guarantee that the generated translation conveys the same meaning as its source (i.e., avoiding semantic shift). Most contributions in translation often focus on high-resource languages (e.g., English and Chinese). Consequently, training and evaluation are more difficult for low-resource languages due to the lack of accessible data covering them (Taunk et al., 2023). The mismatch between train and test data (i.e., the model being trained on data different from the data it is evaluated or used on) can significantly impact the performance of any translation system (Currey et al., 2020; Artetxe et al., 2023). To mitigate this problem, Artetxe et al. (2023) adapt the human training data by back-translation (e.g., Portoguese to English to Portoguese), making it more similar to a machine-generated model input during test time. They assume that applying machine translation twice induces a similar error and thus reduces the gap between train and test data. A well-selected dataset can also help reduce the train/test mismatch (Vilar et al., 2023).

3.4.1 Sub-tasks and Datasets in Translation.

Among our retrieved papers, the most popular translation sub-tasks are *sentence-level* and *document-level* translation. Most datasets in translation consider short sequences for their tasks (*sentence-level*) as they are easier to annotate than longer sequences (*document-level*). Thus, current systems are often more capable of handling short sequences as they are largely trained on corresponding data. However, *document-level* information might help to solve word ambiguity as the longer context can allow better handling of discourse phenomena like coreference, omissions, and coherence (Zhang & Zong, 2020).
Sentence-Level:
Description. Sentence-level translation is a sequence-to-sequence text generation task in which maps an individual source sentence to its equivalent in the target language, maintaining semantic meaning and grammatical correctness.
Modeling. Sentence-level translation involves sentence-aligned datasets, typically sampled from full-sized documents. Systems must process relatively short sequences. As the complexity of short sequences is low, corresponding translation systems are abundant and capable. Works on sentence-level translation propose a variant of the encoder-decoder Transformer, predicting tokens with a classifier over a large database of translation examples (Khandelwal et al., 2021), or prompt existing pre-trained LLMs (Zhang et al., 2023) to perform machine translation. The most popular datasets in the translation field come from the WMT General Machine Translation task (Kocmi et al., 2023). Since 2008, there have been regular releases and updates to the WMT dataset, which comprises several other corpora (Callison-Burch et al., 2008; Kocmi et al., 2023). Each corpus covers a different set of languages (e.g., Chinese, German, Japanese) and domains (e.g., e-commerce, conversation, news), with a few thousand examples per language. Together, these models and benchmarks have established sentence-level translation as a largely saturated setting, providing a foundation for exploring more challenging scenarios, such as document-level translation.
Challenges of sentence-level translation. Sentence-level translation is specifically limited due to a lack of context. For example, when referring to out-of-context personalities by pronouns or when encountering words that can have multiple translations depending on the language (Zhang & Zong, 2020). Word ambiguities are difficult to handle when only a single sentence is provided as a source. In addition, translation models trained on short sequences perform significantly worse when tested on long sequences (60+ words) (Koehn & Knowles, 2017; Pang et al., 2025). Thus, success in semantic preservation for longer sequences is especially influenced by the

variability, context, and ambiguity of different languages. Further challenges for modern LLMs in translation include inference efficiency, translation of low-resource languages during pretraining, and evaluation according to human-aligned translation quality (Pang et al., 2025).

Document-Level:

Description. Document-level translation is a sequence-to-sequence text generation task that leverages in-document context to resolve ambiguities, maintain coherence, and ensure consistency across sentences that are difficult to handle in isolated sentence-level translation (Maruf et al., 2022).

Modeling. The exact length of the segmented sequences heavily differs between works, ranging from small paragraphs to complete books. Thus, we reference efforts to translate text into sequences longer than sentence-level as document-level translation. Most document-level translation approaches leverage the long-context modeling capabilities of recent LLMs (Wang et al., 2023a). In general, LLMs are more suitable for translating documents than single sentences (Pang et al., 2025). Data for document-level translation is scarce. Many of the available datasets are paid resources that require a fee to access, such as the NIST corpora (NIST Multimodal Information Group, 2010, 2013). They are relatively small (100 samples per language) but provide reference translations for longer news articles. NIST datasets also only cover a few languages and are not uniformly structured. For example, NIST 2005 (NIST Multimodal Information Group, 2010) contains 4-way reference translations of Newswire articles in Chinese and Arabic (100 samples per language), while NIST 2012 (NIST Multimodal Information Group, 2013) contains 222 documents with Chinese-to-English pairs from Newswire and web data. Our investigation suggests that the lack of human datasets for this sub-task entails high upfront costs for research, as it requires creating high-quality translations of longer documents.

Challenges of document-level translation. Research on document-level translation remains limited, largely due to the combined challenges posed by long input sequences and the scarcity of large-scale, annotated datasets. Discourse phenomena such as coreference, omissions, and coherence can occur in a manner that is specific to long sequences, leading to performance differences between translating shorter and longer sequences (Zhang & Zong, 2020). While sentence-level models struggle to capture such phenomena, modern LLMs can handle more context at the cost of interpretability (Wang et al., 2023a). Further studies of document-level translation could yield interpretable improvements in handling ambiguities and discourse phenomena.

3.5 Paraphrasing

Task definition. Paraphrases are texts expressing (approximately) identical meanings that use different words or structures (Dolan & Brockett, 2005; Vila et al., 2015; Wahle et al., 2023b). Thus, paraphrasing aims towards the generated text having a high semantic similarity to the source text.

Main characteristics. In paraphrasing, the output length is not a relevant criterion, which is different from other tasks like translation. Thus, a paraphrased text can be the same length, shorter, or longer than the original text. Paraphrases can be generated in several ways, such as by replacing or adding words, making grammatical changes, or changing the order of words (Kovatchev et al., 2018).

Challenges. A key challenge of paraphrasing is the lack of suitable evaluation metrics. The similarity of paraphrases is difficult to measure, with different lexical or semantic changes taking place. Most authors use BLEU or other n-gram-based metrics to test their systems' performance (Grusky, 2023). As Shen et al. (2022b) point out, this has issues because their correlation with human judgments is low, and there are many ways to paraphrase a text. Consequently, multiple references for each data point would facilitate the training of paraphrasing systems (Zheng et al., 2018), but we find that most corpora still only include one reference (Kovatchev et al., 2018; Wang et al., 2017). Model-based metrics offer an alternative to overlap-based evaluation, but come at the cost of reduced interpretability and potential model bias (He et al., 2023). Machine-generated paraphrases are semantically closer to their original source text when compared to human paraphrases (Becker et al., 2023). This indicates a gap

between human and machine paraphrases that is not captured by model-free quality metrics. Additionally, some paraphrasing datasets are thematically balanced, whereas others exhibit concentrated or unbalanced coverage of domains such as politics (Becker et al., 2023).

3.5.1 Sub-tasks and Datasets in Paraphrasing.

We identify two major sub-fields of paraphrasing research, namely *uncontrolled* paraphrasing and *controlled paraphrasing*. As *uncontrolled* paraphrasing does not come with additional constraints on the generation process, it is inherently characterized by its output variability. The other branch of paraphrasing studies aims to *control* the variability of paraphrases during generation. As recent models demonstrate satisfactory performance on *uncontrolled* paraphrasing tasks, the focus has shifted toward developing systems that provide greater *control* to the user. Thus, *uncontrolled* paraphrasing is well-established compared to *controlled* paraphrasing.

Uncontrolled:

Description. Uncontrolled paraphrasing is a sequence-to-sequence text generation task in which a model rewrites an input text into a semantically equivalent alternative, without enforcing specific lexical, syntactic, or stylistic constraints on the output.

Modeling. Common approaches use an encoder-decoder architecture that first creates a semantic representation of the input text and then outputs a paraphrase (Egonmwan & Chali, 2019). Garg et al. (2021) fine-tune an autoregressive LM to generate paraphrases. Uniquely, their approach does not rely on parallel paraphrase datasets. It instead uses lexical control signals to guide how words or phrases should change in the paraphrase. A reward with reinforcement learning encourages semantic similarity to the original sentence while promoting lexical and syntactic diversity. Several datasets are available for paraphrasing considering uncontrolled scenarios. For example, TURL (Lan et al., 2017) comprises 2.8m multi-rater annotated sentence pairs from Twitter news. Quora Question Pairs (Wang et al., 2017) comprises 400k similar questions from the Quora forum with binary paraphrase annotations.

Challenges of uncontrolled paraphrasing. Since many possible paraphrases can be produced, the n-gram-based comparison to a single reference is questionable. To encourage diversity in paraphrases, we, therefore, suggest the use of multiple references per source text (Lan et al., 2017; Zheng et al., 2018). However, these are often not provided in today's paraphrasing datasets. We suggest that new corpora should come with multiple reference paraphrases as a standard.

Controlled:

Description. Controlled paraphrasing is a conditional sequence-to-sequence text generation task in which a model produces semantically equivalent text while adhering to predefined control signals, such as paraphrase type (e.g., modal verb changes) (Wahle et al., 2023b; Vila et al., 2014; Kovatchev et al., 2018), target characteristics like quality (Bandel et al., 2022), or structural constraints such as syntactic form (Chen et al., 2019; Kumar et al., 2020).

Modeling. To generate paraphrases in a controlled manner, most prior work focuses on the encoder side of text-to-text systems. Wahle et al. (2023b) show that paraphrases of a specific type can be generated using autoregressive models like BART (Lewis et al., 2020a) or PEGASUS (Zhang et al., 2020a) by conditioning the encoder on task-specific prompts or labels. Several approaches introduce explicit structural control. Chen et al. (2019) and Kumar et al. (2020) employ semantic and syntactic encoders for syntax-controllable paraphrase generation, while Yang et al. (2022a) encode constituency trees with Transformers to capture parent–child and sibling relations, constraining outputs to a given syntactic template. Beyond structural control, Bandel et al. (2022) directly regulate paraphrase quality by conditioning generation on semantic similarity as well as syntactic and lexical distance. This enables the model to trade off meaning preservation and lexical and syntactic variation during decoding. Finally, Zhang et al. (2021) leverage manifold sentiment information to guide paraphrase generation.

The ETPC dataset (Kovatchev et al., 2018) comes with the extended paraphrase typology. This includes granular annotations of the categories morphology-based, lexicon-based, lexico-syntactic-based, syntax-based, discourse-based, and other changes. Though it only includes 5k samples and shows a balanced distribution of covered topics (Becker et al., 2023), other datasets usually lack such granular type annotations. However, some other datasets focus on one specific paraphrase type like sentence splitting (Botha et al., 2018) or entailments (Williams et al., 2018). Overall, we observe that datasets are abundant for specific paraphrase types, such as sentence splitting, whereas other types, such as polarity changes, syntax control, or sentiment control, require more specialized datasets. An additional avenue is to adapt non-paraphrasing research on controllability (Ghosh et al., 2017; Huang et al., 2020b; Yang et al., 2023a; Seo et al., 2023; Yang & Klein, 2021).

Challenges of controlled paraphrasing. While uncontrolled paraphrasing is a well-established task, we find that controlled paraphrasing tasks such as paraphrase type generation are incipient (Wahle et al., 2023b). Thus, many architectural characteristics that make controlled paraphrasing systems superior remain unclear, as they also depend on the controlled attribute. We could not identify many datasets that provide a comprehensive selection of annotated paraphrase types or controlled characteristics like sentiment. Therefore, we encourage other researchers in the field to construct comprehensive datasets that yield controllable attributes, such as paraphrase types (Kovatchev et al., 2018). This would foster further research on controlled paraphrasing.

3.6 Question Answering

Task definition. Question answering (QA) takes a question as input and outputs a streamlined answer or a list of possible answers based on background knowledge (Cai et al., 2020). This background knowledge can either be provided as a supplementary document, a knowledge base, or leveraged from training data when no additional context is provided.

Main characteristics. Background knowledge of a QA system can be *internal* or *external* (Yu et al., 2022). *Internal knowledge* takes place within the training data and input text(s), including but not limited to keywords, topics, linguistic features, and internal graph structure. *External knowledge* acquisition occurs when knowledge is provided from outside sources, such as a knowledge base, knowledge graph, or grounded text. While there exist QA systems (Cai et al., 2020) for topic-restricted domains (e.g., math QA), most research focuses on open-domain QA, i.e., QA systems with capabilities that are not constrained on a topic because they generalize better to practical problems.

Challenges. The difficulties of QA stem from open-domain evaluation and the interpretability of the system's answers. Evaluation of generated answers is difficult for very abstractive settings like open-domain QA where the source does not directly indicate what the correct answer could be (Durmus et al., 2020). This contrasts with multiple-choice settings, in which evaluation is comparatively simple, e.g., via regular expression text matching. To bridge the gap of interpretability, researchers draw inspiration from how humans synthesize a chain of thought for reasoning. Although Chain-of-Thought Prompting (Wei et al., 2024) and Solo Performance Prompting (Wang et al., 2023b) are competent approaches to improve reasoning in QA systems, their implementation is not trivial because they require a specific prompt design and large models that are capable of mimicking human-like reasoning this way.

3.6.1 Sub-tasks and Datasets in Question and Answering.

According to our investigations, the most popular QA sub-tasks are *internal knowledge-grounded QA* (e.g., open-domain QA) and *external knowledge-grounded QA* (e.g., reading comprehension).

Internal Knowledge:

Description. Internal-knowledge question answering is a text generation task in which a model produces an answer solely based on the input question and knowledge implicitly stored in its parameters or other underlying

representations (Yu et al., 2022). The system can not leverage additional resources to answer a question. Some works refer to internal knowledge as parametrized knowledge (Ma et al., 2025).

Modeling. Corresponding systems are mostly employed with pre-trained LLMs that utilize the internal knowledge memorized from large web corpora (Tan et al., 2023). Such internalized knowledge supports open-domain question answering by enabling the model to retrieve and combine relevant information without explicit access to external databases at inference time. Large and diverse pre-training corpora like Common Crawl[7] increase the amount of internal knowledge that can be leveraged for QA, which is particularly important for handling rare entities, long-tail queries, and heterogeneous question formulations. Recent work suggests that the factual knowledge encoded in a model's internal computations can significantly exceed what the model actually generates in its outputs, revealing that some correct answers may be present internally yet not always produced by repeated sampling methods (Gekhman et al., 2025). Moreover, this hidden knowledge gap constrains the practical gains achievable by increasing test-time compute in open-domain QA, because some of the model's factual information remains inaccessible when relying solely on token-level generation. Beyond pre-training, domain-specific datasets can be used for fine-tuning to adapt the model to specialized domains.

Challenges of internal knowledge-grounded QA. A core challenge that comes with QA systems is their tendency to hallucinate information (Zhuang et al., 2023), a characteristic that heavily countervails the goal of factual QA. Thus, there exist many efforts in mitigating hallucinations, making LMs more reliable for QA tasks (Zhou et al., 2021; Ji et al., 2023). We discuss hallucinations thoroughly in Section 5.3.

External Knowledge:

Description. External-knowledge question answering is a text generation task in which a model produces an answer by jointly considering the input question and retrieved or provided external evidence. External information is provided at inference time, such as a knowledge base, a knowledge graph, or grounded text. When grounded text is used as the external knowledge source, the task is commonly referred to as reading comprehension.

Modeling. To leverage external knowledge, most approaches focus on LMs with sufficient context modeling capabilities (Tan et al., 2023). They can be prompted to answer a question based on a provided source text (Tan et al., 2023). Other knowledge, like knowledge graphs, is leveraged by additional modules, e.g., a graph neural network (Lu et al., 2023b). Nishida et al. (2020) improve the performance of their QA system on specialized domains by stacking a reading comprehension layer and an LM layer on top of BERT (Devlin et al., 2019). Given specific user requirements, some studies employ custom tools to facilitate or ensure accurate responses. Zhuang et al. (2023) employs 13 external tools, like a code interpreter or a mathematical computer, to enhance the performance of an LLM on the QA task. Retrieval augmented generation (RAG) combines document vectorization with retrieval (Lewis et al., 2020b). Queries are embedded, matched against a vector database, and the retrieved context is used to generate grounded responses. QA datasets with external knowledge are abundant, especially in the subfield of reading comprehension (Rogers et al., 2023). The widely used SQuAD dataset (Rajpurkar et al., 2016) contains crowdsourced reading comprehension questions with reference answers supported by a paragraph. SQuAD 2.0 (Rajpurkar et al., 2018) adds 50k unanswerable questions to the base dataset that challenge models to predict whether the generation of an answer is feasible at all. Du et al. (2017) and Yuan et al. (2017) provide a model that can generate synthetic reading comprehension questions. Gao et al. (2019) propose an advanced system with controllable difficulty levels. It can be used to generate more samples on specific domains if crowdsourcing is not an option.

Challenges of external knowledge-grounded QA. The sub-tasks' difficulty rises with the complexity of the questions. Despite recent advances, LLMs still exhibit limitations in addressing questions that require complex, multi-step reasoning (Tan et al., 2023), making effective reasoning with LLMs an active area of research. Additional challenges arise with non-answerable or non-factoid questions, as LMs are typically trained to always generate some text as

[7] https://commoncrawl.org/

an answer (Rajpurkar et al., 2018). Ideally, a model should be able to tell the user when a competent answer seems unrealistic, given the provided context. QA datasets like SQuAD (Rajpurkar et al., 2016) have been criticized because they are overly dependent on the similarity of question/answer sentences rather than on human-like reasoning, meaning they require only superficial reading skills (Uto et al., 2023). Datasets like ELI5 (Fan et al., 2019) can provide alternatives that require more reasoning and multi-step thinking.

4 Evaluation

Several metrics have been proposed or adopted from other research fields to evaluate, compare, and discuss text generation systems. These metrics serve as proxy measures of desired text-generation characteristics, such as coherence, fluency, and semantic diversity. Table 2 shows the summary of all metrics identified. We identify 16 distinct automated metrics from our systematically retrieved papers (excluding auxiliary papers added during writing) and organize them into three major types: *model-free*, *model-based*, and *human* evaluation. *Model-free* metrics are usually based on static, rule-based algorithms, such as text overlap (e.g., BLEU (Papineni et al., 2002)). *Model-based* metrics are automated metrics that rely on learned models rather than only surface-level rules. They include embedding- or encoder-based similarity metrics (e.g., BERTScore (Zhang et al., 2020b)) or sequence-to-sequence scoring metrics (e.g., BARTScore (Yuan et al., 2021)) to derive a final score. Another field of evaluation concerns *human evaluation*, which measures characteristics like fluency, faithfulness, coherence, and others in text generation systems (Ghosh et al., 2017; Ma et al., 2018; Qader et al., 2018; Feng et al., 2021; Lu et al., 2022b). We also discuss LLM-as-a-judge (Zheng et al., 2023a), which uses an instruction-tuned LLM to predict a decision or score. In the retrieved papers, we find four distinct methods used for human *labeling* (e.g., Likert Scale) and four metrics for annotator *agreement* (e.g., Krippendorff Alpha).

4.1 Desired Characteristics in Text Generation

All metrics are used to estimate the desired characteristics of generated texts. We identify several characteristics, e.g., fluency and coherence, across the papers we considered. Although there are many metrics to assess generated texts, they are proxies for *quality*. Other characteristics, such as bias, faithfulness, and reasoning, also have their own evaluation procedures and will be covered in Section 5.

4.1.1 Quality.

A high-quality text has to fulfill various criteria, such as having a correct grammatical structure or being coherent and fluent. Sonntag (2004) decomposes textual quality into the three general parts: intrinsic, contextual, and representational quality. We find that this definition partly omits task-specific characteristics. For example, some tasks like dialogue generation (Li & Zhao, 2023; Zhang et al., 2020d) specifically require diverse outputs to avoid repetitions and encourage creativity (Chung et al., 2023; Alihosseini et al., 2019). We detail our expanded definitions below. First, *intrinsic quality* includes the need for a text to be accurate, unbiased, and generated from a data source with a high reputation. Secondly, *contextual quality* comprises the amount of text being produced, the completeness of the output, the coherence of timeline events, and the relevance of the text to the prompt or document (Deng et al., 2021). Third, *representational quality* requires a consistent and concise output that is easily understood. This also involves the fluency of the output, and the text should be interpretable by the reader. Lastly, *task-oriented quality* captures requirements specific to the downstream task, such as diversity and creativity in dialogue generation (Chung et al., 2023), stylistic adherence in creative writing (Peng, 2022), or faithfulness and coverage in summarization (Maynez et al., 2020; Huang et al., 2024).

Challenges of quality evaluation. Considering these dimensions of text quality (i.e., intrinsic, contextual, representational, and task-oriented), evaluation in text generation research is commonly performed by assessing the similarity of a machine-generated text to a reference text written by humans (Papineni et al., 2002; Lin, 2004). To consider more specific dimensions of quality or additional attributes, but especially task-oriented

Table 2. Automated evaluation metrics in the text generation field. We consider systematically retrieved Semantic Scholar documents from 2017 to 2023 after the eligibility criteria (cf. Section 2.3). "Used" marks the number of papers that propose, survey, or apply the metric in their publication.

Type	Category	Metric	Description	Used
Model-free	N-gram	BLEU	Textual overlap between source and reference (precision).	69
		ROUGE	Textual overlap between source and reference (recall).	46
		METEOR	Textual overlap between source and reference (precision and recall).	32
		CIDEr	Measures consensus on multiple reference texts.	15
		chrF++	Character-based F-score computed using n-grams.	13
		Dist-n	Measures generation diversity by the percentage of distinct n-grams.	8
		NIST	Alters BLEU to also consider n-gram informativeness.	6
		Self-BLEU	Measures generation diversity by calculating BLEU between generated samples.	2
	Statistical	Perplexity	Fluency metric based on the likelihood of word sequences.	23
		Word Error Rate	The rate of words that are different from a reference sequence based on the Levenshtein distance.	11
	Graph	SPICE	Measures the semantic similarity of two texts by the distance of their scene graphs.	6
Model-based	Hybrid	BERTScore	Contextual token similarity to measure textual overlap.	13
		MoverScore	Uses contextualized embeddings and captures both intersection and deviation from the reference for a similarity score.	6
		Word Mover Distance	Distance metric to measure the dissimilarity of two texts.	2
	Trained	BLEURT	Models human judgment on text quality.	4
		BARTScore	Likelihood metric for assessing candidate consistency with the source or reference text.	3

quality, supplementary metrics are needed. For example, Jagfeld et al. (2018) measure the textual diversity by the number of unique sentences and words in the output. Xu et al. (2017) measure textual novelty by the number of infrequent words in the generated responses, excluding the top 2,000 most frequent words. Unlike traditional classification and regression tasks, where error rates can be clearly defined, generative tasks have a complex answer space and, in many cases, are subjective. Most datasets and tasks (e.g., the summarization dataset XSum (Narayan et al., 2018) or the paraphrasing dataset ETPC (Kovatchev et al., 2018)) provide only a single reference text. As a result, automatic evaluation is implicitly bound to one surface realization of the target output. This limitation causes semantically equivalent but lexically divergent generations to receive low scores due to insufficient token-level overlap, despite being judged as valid by humans. Consequently, many commonly used metrics, particularly n-gram-based metrics, show limited correlation with human judgments (Wahle et al., 2023b;

Jagfeld et al., 2018). Moreover, n-gram-based metrics rely on hard or soft subsequence alignments, which do not measure the interdependence between consecutive sentences well (Fabbri et al., 2021) and show an unsatisfying correlation with human judgments (Gatt & Krahmer, 2018b). We outline metrics to measure textual quality and other attributes, and to understand the limitations of current evaluation methodologies.

4.2 Model-Free

The most used evaluation methods for text generation are model-free metrics because they are easy to adapt and have low computational costs (Table 2). The most common metrics used for evaluating the quality of machine-generated text rely on n-gram-based comparisons, such as BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and METEOR (Banerjee & Lavie, 2005). Other metrics compare the semantic differences using graphs (e.g., SPICE (Anderson et al., 2016)) or the probability of word sequences (e.g., perplexity (Jelinek et al., 2005)). In general, a metric tries to align well with human judgments, although various human biases and imperfections exist. As noted earlier, many model-free metrics (e.g., BLEU, ROUGE) do not correlate well with human judgments, as they sometimes assign high scores to low-quality texts (Gatt & Krahmer, 2018b). Still, these metrics are widely used in text generation research as cost-efficient proxies. Thus, these metrics should be used only under constrained circumstances, i.e., for extractive tasks, and are very limited for tasks with diverse outputs such as abstractive summarization (Junadhi et al., 2025). We further detail each metric, including its capabilities and limitations.

BLEU is the most employed model-free metric in our review (69 papers) (Papineni et al., 2002). It measures the word overlap between a candidate text and one or more references. The score indicates the precision of the generated text based on the overlapping n-grams. A high BLEU score indicates that the evaluated text is very similar to the reference in terms of wording. To mitigate highly precise yet very short texts, a brevity penalty is added. Notably, BLEU does not account for grammatical errors and only considers the n-grams of a text for comparison. NIST alters BLEU to also consider the informativeness of n-grams by assigning lower scores to more frequent n-grams than to less frequent ones (Doddington, 2002). Self-BLEU measures the diversity of generated samples by applying BLEU to pairs of generations (Zhu et al., 2018). As mentioned before, n-gram-based metrics like BLEU can be susceptible to noisy text (Vaibhav et al., 2019) and are unreliable when used for abstractive tasks (Junadhi et al., 2025). Thus, their use is limited and best accompanied by additional model-based metrics or human evaluation.

ROUGE measures the recall of n-gram overlap between a generated text and a single reference (46 papers) (Lin, 2004). It is also known as ROUGE-N, where N stands for the length of the n-grams used for the calculation (e.g., ROUGE-1 for unigrams, ROUGE-2 for bigrams, etc.). ROUGE-L specifically considers the longest common subsequence between the generated text and a reference. Although ROUGE was originally recall-oriented, it is now often reported as ROUGE-F1, which additionally penalizes outputs that achieve high recall by adding irrelevant or excessive text. As an n-gram-based metric, ROUGE is inherently biased towards lexical similarity (Ng & Abrecht, 2015). As a countermeasure, ROUGE-WE (Ng & Abrecht, 2015) adds soft lexical matching through word embeddings (making it a hybrid model-based metric). Here, the individual vectors of constituent tokens are multiplied to produce the vector for an n-gram. Just as BLEU, ROUGE is also not reliable for abstractive tasks with diverse outputs (Junadhi et al., 2025) and is susceptible to noisy text (Vaibhav et al., 2019).

METEOR is another overlap metric (32 papers) (Banerjee & Lavie, 2005). While BLEU captures precision and ROUGE recall, METEOR combines both into a single metric. A reordering penalty term punishes texts with the correct words that appear in the wrong order. Abstractive settings that introduce word-level changes from input to output, such as synonyms, still pose a limitation for the metric. This is because METEOR is also an overlap-based metric, similar to BLEU and ROUGE.

CIDEr is a consensus-based metric that matches a single sentence to a set of reference sentences (15 papers) (Vedantam et al., 2015). While originally developed for the image captioning task, CIDEr is often used in various

text generation tasks (Li & Liang, 2021; Fabbri et al., 2021; Lu et al., 2022b). It calculates cosine similarity between 1- to 4-gram co-occurrences and captures precision, recall, grammaticality, and word importance by down-weighting common n-grams.

chrF++ is an n-gram-based quality metric that computes n-grams on a character level instead of the token level (13 papers) (Popović, 2017). This makes chrF++ independent from language and tokenization. It improves correlation with human judgments by incorporating added unigrams and bigrams at the word level.

Distinct-n shows the number of distinct n-grams divided by the total number of generated tokens (eight papers) (Li et al., 2016). Researchers use this metric to measure the diversity of a sentence, penalizing sentences with repeated words. It is not capable of capturing text quality. Most works use $n \in 1, 2, 3$ (Ni & McAuley, 2018).

Perplexity in text generation is defined as the degree of uncertainty of a model in predicting a new sequence (23 papers) (Jelinek et al., 2005). Specifically, perplexity analyzes the likelihood of word sequences, comparing the generated probability distribution of words with the actual input. A lower perplexity score means the trained model is better at generating with varying input data because it is less perplexed/confused. Perplexity is only a measure of certainty and, therefore, does not necessarily correlate with the quality of the generated text. Still, it is a widely used metric that indicates how confident a model is during generation.

Word Error Rate measures the amount of edits required (substitutions, deletions, insertions) until a sequence equals the reference (11 papers). Therefore, the Word Error Rate measures the similarity of sentences, indicating textual quality when comparing a result with a reference text. As the Word Error Rate is a metric commonly used in image captioning tasks, we examined the papers that mention it. Most mentions of the Word Error Rate in our selection of works either review existing metrics (Fatima et al., 2022) or apply it to speech recognition (Huang et al., 2018), due to its higher sensitivity to specific error types (substitutions, deletions, insertions) compared to more flexible n-gram-based approaches. It does not capture semantic meaning and therefore cannot distinguish between contextually relevant and irrelevant words.

SPICE uses a unique approach to measure semantic similarity by implementing a distance measurement between two texts' scene graphs that encode objects, attributes, and relations (six papers) (Anderson et al., 2016). SPICE ignores fluency, and a highly similar text pair might not be well-formed, but it captures semantic similarity more effectively than n-gram-based metrics. This metric is also used for tasks like image captioning because it can capture relations between objects and attributes well (Keneshloo et al., 2019). In the papers we assessed, SPICE is often used alongside other metrics, primarily because it is one of the few model-free metrics that do not rely on n-gram comparisons (Lin et al., 2020; Lu et al., 2022b).

In summary, model-free metrics remain the most commonly used evaluation tools due to their simplicity, low computational cost, and ease of comparison across studies. However, our analysis confirms that these metrics primarily capture surface-level properties such as lexical overlap, diversity, or prediction confidence, and often fail to reflect semantic adequacy or overall text quality. Their weak correlation with human judgments, especially for abstractive and open-ended generation tasks, limits their interpretability when used as standalone measures. Consequently, model-free metrics are best suited to constrained or extractive settings and should be complemented by model-based metrics or human evaluation to yield a more reliable assessment of generated text.

4.3 Model-Based

With model-based metrics, we capture the semantic meaning and potentially other attributes of a sentence using neural models such as BERT (Devlin et al., 2019). Thus, model-based metrics provide a more flexible measurement of textual characteristics. This relatively novel category of metrics generally outperforms many well-established overlap metrics (e.g., BLEU (Papineni et al., 2002; Babakov et al., 2022)). Considering LMs' increasing capabilities, model-based metrics are employed 28 times in the papers from our list, but most studies still include model-free

metrics as a widely accepted rule-based measurement, as indicated in Table 2. We differentiate between fully end-to-end *trained metrics* that rely on handcrafted features and/or learned embeddings (e.g., BLEURT (Sellam et al., 2020), BARTScore (Yuan et al., 2021)) and *hybrid metrics* that combine trained elements (e.g., contextual embeddings) with computational logic (e.g., BERTScore (Zhang et al., 2020b)). Model-based metrics (*trained* or *hybrid*) take advantage of available rating data for their training and should be more robust to distribution drifts than model-free metrics (Sellam et al., 2020). Ji et al. (2023) point out that neural models are prone to producing errors in predicting textual similarity or other attributes. Thus, relying solely on model-based metrics can propagate errors that degrade the models' accuracy. Moreover, model-based metrics are inherently prone to insensitivities, biases, and loopholes (He et al., 2023). For example, BERTScore (Zhang et al., 2020b) is confused by truncation errors in summarization.

BERTScore is the most-used model-based metric in our review (13 papers) (Zhang et al., 2020b). It is a hybrid metric that leverages contextual token similarity to measure the overlap of a text with a reference. More specifically, BERTScore uses pairwise cosine similarity to measure the similarity between word embeddings of a source and a reference text. As it considers precision, recall, and F1 measure, BERTScore can be applied to various text generation tasks and correlates well with human judgments. The metric is highly expressive and inherently tunable for task-specific properties (e.g., fluency, style, and coherence), though reliability varies across these properties.

MoverScore uses contextualized representations (e.g., BERT embeddings (Devlin et al., 2019)) to measure the similarity between two texts (six papers) (Zhao et al., 2019). It uses the Word Mover Distance (Kusner et al., 2015) to determine how much one set of embeddings needs to be changed to transform into the reference embeddings. Thus, MoverScore considers not only the intersection but also the deviation from the reference. Sentence Mover's Similarity (Clark et al., 2019) extends this idea by incorporating sentence embeddings and improving correlation with human judgments.

Word Mover Distance calculates the dissimilarity of two texts (Kusner et al., 2015) using static word representations (e.g., word2vec (Mikolov et al., 2013)) (two papers). While Word Mover Distance relies on traditional static word embeddings that can be visualized for interpretability, some researchers prefer MoverScore as a metric due to its greater ability to capture context.

BLEURT is a fully learned metric that models human judgment on text quality (four papers) (Sellam et al., 2020). BLEURT is highly adaptable, end-to-end trained, and can be fine-tuned for other tasks. However, hybrid metrics like BERTScore (Zhang et al., 2020b) may yield better results with limited training data and do not rely on the assumption that train and test data are similarly distributed.

BARTScore leverages a text-to-text model to assess text from perspectives like informativeness, coherence, and factuality (three papers) (Yuan et al., 2021). While it can also capture precision and recall, BARTScore also considers faithfulness and can be prompted or fine-tuned to fit more specific tasks. BARTScore++ improves on BARTScore by considering the explicit and implicit error distances when detecting hallucinations (Lu et al., 2023a). The application of BARTScore depends on the task, requiring users to decide which variation to choose and how to design the prompt.

In summary, model-based metrics enable a richer evaluation of generated text by capturing semantic and higher-level textual properties that model-free metrics cannot address. Hybrid approaches such as BERTScore and MoverScore are widely adopted due to their balance between expressiveness and robustness, while fully trained metrics like BLEURT and BARTScore offer stronger alignment with human judgments at the cost of greater sensitivity to biases and distribution shifts. Despite their overall superior performance, model-based metrics exhibit systematic failure modes and inherited model biases, which limit their reliability when used in isolation. As reflected in the reviewed studies, they are therefore most effective when combined with model-free metrics to provide complementary and more stable evaluations.

Table 3. Human evaluation methods in the text generation field. We consider systematically retrieved Semantic Scholar documents from 2017 to 2023 after the eligibility criteria (cf. Section 2.3). "Used" indicates the number of papers that apply each method.

Type	Category	Metric	Description	Used
Human	Labeling	Likert Scale	Humans can choose on a scale, e.g., from 1 (horrible quality) to 5 (perfect quality).	22
		Pairwise Comparison	Humans choose the best example from two samples.	10
		Turing Test	Can quantify how distinguishable human text is from machine-generated text.	6
		Binary	Humans are answering binary questions with yes or no.	3
		Best-Worst Scaling	From a list of examples, humans are instructed to select the best and worst output.	2
	Agreement	Krippendorff Alpha	Measures the disagreement between annotators for nominal, ordinal, and metric data.	4
		Fleiss Kappa	Measures the agreement on nominal data between a fixed pair of annotators.	4
		Pearson Correlation	Displays the agreement between annotators by measuring linear correlation.	3
		Spearman Correlation	Displays the monotonic relationships on ranked data.	2

4.4 Human Evaluation

The standard in assessing the quality, fluency, faithfulness, and coherence of text generation systems is human evaluation. Methods for human evaluation can be divided into labeling (e.g., classification) and agreement measurements (e.g., correlations between annotator judgements), which can serve as indicators of the quality of the produced labels.

Usually, human evaluation is performed through crowdsourcing marketplaces (e.g., Amazon Mechanical Turk[8], Prolific[9]), where annotators are given the task of labeling samples based on their intuition. We report the used methods during human annotation in Table 3. Ultimately, the choice of evaluation method depends on the task to be solved. Most used labeling methods are Likert scales (22 papers), e.g., "How coherent is the text from 1 (least coherent) to 9 (most coherent)?". While the Likert scale is suitable for expressing text quality, ratings are often not explained and don't indicate the problematic text segment (Dou et al., 2022). Other works use pairwise comparisons (ten papers), i.e., directly comparing two systems' outputs to explain comparative improvements Dou et al. (2022).

We find that only a relatively small number of studies report standard agreement metrics, such as Krippendorff's alpha (13 papers). Especially in complex evaluation setups (e.g., emotion analysis (Nasution & Onan, 2024)), single annotators' judgments can vary substantially. Without agreement metrics, it is difficult to assess the reliability of the provided labels and to interpret model performance relative to human consistency. For this reason, we encourage evaluators of text generation systems to report agreement metrics alongside their labels in future

[8] https://www.mturk.com/
[9] https://www.prolific.com/

work. While this requires additional annotation effort, agreement metrics indicate the level of human consistency on a task and thus define a meaningful reference point for model performance, particularly when comparing systems on subjective or underspecified evaluations.

Human annotation and evaluation are costly, especially when consulting experts, so they are often avoided or done on a small scale. Reiter & Belz (2009) find that experts and non-experts correlate strongly on features like readability, clarity, and general appropriateness. Bhandari et al. (2020) use the pyramid method (Nenkova & Passonneau, 2004) to assess automated evaluation metrics themselves through human evaluation.

Some efforts have been made to provide easy-to-use frameworks to increase the effectiveness of human evaluations. For example, Scarecrow (Dou et al., 2022) is a framework for crowd-annotating several error types in machine-generated and human-generated text. Here, ten workers annotate each paragraph to provide an inter-annotator agreement score. The Multidimensional Quality Metric by Freitag et al. (2021) gets judgments for the translation task to improve the quality of human judgments through error analysis. The metric requires evaluators to identify errors and categorize them into severity levels. To assess how well a human evaluation and annotation are performed, authors can provide Kendall's Tau or Pearson Correlation. The TrueSkill algorithm also evaluates human annotations by creating clusters of systems for both quality and naturalness (Sakaguchi et al., 2014; Gehrmann et al., 2018).

The LLM-as-a-Judge paradigm uses an instruction-following model as a scalable alternative to human raters by prompting it with the task, example, and the candidate output(s), then extracting either a decision or a score (Zheng et al., 2023b). Methodologically, LLM-as-a-judge is model-based because the evaluator is a learned model. Conceptually, however, it approximates human evaluation protocols by prompting a model to produce judgments, preferences, or scores. We therefore discuss it in this section as a scalable alternative to human raters, while treating its outputs as model-generated annotations rather than human labels. Such judges can show high agreement with human preferences but exhibit systematic failure modes such as position/order bias, verbosity/length bias, and occasional self-enhancement or style biases. This motivates mitigations like response-order randomization, standardized criteria, and debiasing for confounders such as length (Zheng et al., 2023b). Nasution & Onan (2024) find that model-based annotations can be high-quality for simple tasks such as sentiment analysis. In contrast, human annotations consistently outperform LLM-generated ones in intricate NLP tasks such as emotion classification (Nasution & Onan, 2024). In practice, protocols for LLM-as-a-Judge commonly fall into three families: binary judgments (e.g., "Does the answer satisfy constraint X?"); preference judgments (e.g., "Which of A vs. B is better under criteria X?"); selection judgments (e.g., "Which candidate(s) of A, B, C are good answers?") (Li et al., 2025).

In summary, human evaluation remains the most reliable approach for assessing nuanced and subjective aspects of text generation. However, its high cost and inter-annotator variability limit scalability and reproducibility. The limited reporting of agreement metrics further complicates the interpretation of results. While LLM-as-a-Judge methods offer a scalable alternative and show promising alignment with human preferences in simple settings, they exhibit systematic biases and reduced reliability for complex tasks. Consequently, human evaluation should be complemented with agreement analysis and, where appropriate, automated judging to ensure both validity and scalability.

5 Related Challenges

Although text generation has seen much progress concerning its proposed techniques, related challenges remain that have not yet been solved. Aside from the specific challenges and limitations in Section 3, we identify nine main challenges related to text generation. The challenges are *bias*, *reasoning*, *factuality*, *misuse*, *datasets*, *interpretability*, *transparency*, *privacy*, and *computing*. This section describes each challenge while surveying their state-of-the-art mitigation methods and remaining research gaps. We note that some of the terms appear

anthropomorphic (e.g., hallucination, reasoning). As these terms are well-established and widely used in the scientific literature, we also use them. However, we aim to explain each term by a neutral description.

5.1 Bias

Bias, in the most general sense, describes systematically erroneous output that portrays scenarios distortedly (Sheng et al., 2019). In text generation, this distortion often leans towards or against certain demographic groups or concepts and amplifies biases in the training sets (Sun et al., 2019). As bias can occur in different shapes, this problem touches several other areas, such as toxicity in text generation (Zheng et al., 2023a) and sentiment-controlled text generation (Ghosh et al., 2017; Huang et al., 2020b; Kawamae, 2023; Liu et al., 2021). Bias towards groups or concepts is a pressing problem for machine-generated text (Sheng et al., 2019; Dhamala et al., 2021), and its mitigation can reduce the subliminal influence on perceived stereotypes by users of such systems.

While the characteristics of bias are manifold, we identify three main types: *exposure bias*, *allocation bias*, and *representation bias*. First, *exposure bias* occurs when a model is only exposed to the training data distribution instead of its prediction (Keneshloo et al., 2019). To avoid *exposure bias*, Keneshloo et al. (2019) remove the ground-truth dependency during training and use only the model distribution to minimize the loss function. Reinforcement learning concerning a metric like ROUGE (Lin, 2004) can mitigate *exposure bias* (Wang et al., 2018; Jin & Gildea, 2022). A Generative Adversarial Network can be employed as the generator constantly relies on its output during training (Shen et al., 2022a). Second, *allocation bias* appears when models perform better on data associated with groups or situations overrepresented in the training data (Sun et al., 2019). Zhao et al. (2018) show how gender-specific *allocation bias* can be mitigated by creating a second dataset with a swapped gender though it doubles the dataset size and is expensive to create. Third, *representation bias* appears when associations between groups with certain concepts are captured in word embeddings or model parameters. Thus, the representation (e.g., the embedding) reflects the bias (Sun et al., 2019). An intuitive way to tackle *representation bias* is to debias the embeddings directly by regularization (Huang et al., 2020b). This comes with a trade-off between generation quality and control because it can impose the target attribute on the LM on improper positions (Gu et al., 2022; Huang et al., 2020b). Debiasing can be effective while only fine-tuning 1% of parameters on an unbiased dataset, making learned techniques relatively cheap to adopt (Gira et al., 2022). Notably, Schick et al. (2021) show that pre-trained LMs are capable of recognizing their own biases to a considerable degree. Thus, they can be prompted to self-diagnose their output.

The quantification of bias is especially relevant when considering an LM for real-world applications and determining the effectiveness of mitigation techniques. BOLD (Dhamala et al., 2021) and FairPrim (Fleisig et al., 2023) both are helpful datasets to measure various biases (profession, gender, race, religious belief, and political ideology). BOLD metrics (Dhamala et al., 2021) further assess the texts' sentiment, toxicity, regard, psycholinguistic norms, and gender polarity. Notably, not all biases are easy to measure with current metrics due to either a lack of labeled data specific to the bias type or corresponding noise in the supposedly bias-free data.

5.2 Reasoning

Reasoning describes the capability of a text generation system to computationally infer its choices and writing from the source context logically and sensibly. A model should be able to make an understandable decision and explain its reasoning to a human. Several works point out that, even though recent years yielded significant improvements, LMs have difficulties with such commonsense inference (Lin et al., 2020; Zellers et al., 2019). Thus, eliciting reasoning with text generation is an active research topic. LLMs struggle with sequential decision-making that requires common-sense planning, as shown by evaluating their performance on reinforcement learning benchmarks (Li et al., 2024b). The lack of reasoning capabilities comes from misunderstanding facts in the source context and is also referred to as intrinsic hallucinations (Ji et al., 2023) (more details in Section 5.3). Text

generation systems require reasoning capabilities for complex tasks like story generation (Liang et al., 2023) or question answering (Fan et al., 2019).

One popular approach to improve reasoning is Chain-of-Thought Prompting (CoT) (Wei et al., 2024). Here, the model is prompted to think step by step and output its thought process in natural language. It improves performance for arithmetic, commonsense, and symbolic reasoning tasks. Multi-agent systems like Exchange-of-Thought (Yin et al., 2023) maintain the reasoning capabilities of CoT (Wei et al., 2024) while allowing for an iterative self-interaction that improves the output text. The approach imitates a multi-turn conversation between experts on a task. Nevertheless, the effectiveness of multi-agent systems is debated, as strong prompting of a single model can lead to similar task performance (Wang et al., 2024) and prolonged agentic interaction increases the risk of performance collapse (Becker et al., 2025b). Chen et al. (2020) survey another method to improve reasoning capabilities via path-finding strategies on a knowledge graph. We also find considerable research related to testing a model's reasoning capabilities. Stolfo et al. (2023) explain how to specifically test the robustness of mathematical reasoning by grounding a behavioral analysis in a causal graph. The CommonGen dataset (Lin et al., 2020) requires relational reasoning and compositional generalization. The HellaSwag dataset (Zellers et al., 2019) can also be used to test a model's ability of generative commonsense reasoning.

5.3 Hallucinations

Hallucinations describe outputs that are factually wrong or unfaithful to the source. We distinguish between *intrinsic* and *extrinsic hallucinations*, which refers to what content the generated text contradicts to (Ji et al., 2023). *Intrinsic hallucinations* contradict the source content, so they are unfaithful to the training data or input text. *Extrinsic hallucinations* describe outputs that can neither be supported nor contradicted by the source content (Maynez et al., 2020).

The main reasons for hallucinations stem from the inherent sampling in statistical language models and from the source data, because diversity-valuing tasks like dialogue generation inherently generate text that deviates from the factual information in the source data (Ji et al., 2023). Bias in the training data and text generation models can also lead to factually wrong outputs (Qader et al., 2018; Ji et al., 2023). Improving factuality in text generation makes systems more reliable in real-world applications (e.g., chat support assistants and summarization systems).

Overall, models pre-trained on large amounts of data tend to hallucinate less (Maynez et al., 2020; Zhou et al., 2021). Prabhumoye et al. (2021) mitigate hallucinations in document-grounded text generation by implementing a selective attention variant for the documents' contextualized representations. Faithfulness to a source can be improved by representing the knowledge in a simpler relational format and verifying it against relation tuples from the source or reference (Goodrich et al., 2019).

Other works employ mechanisms to detect hallucinations in existing text. Measuring hallucinations with LMs can be challenging they are designed to predict texts of high linguistic quality (e.g., fluency, grammaticality) rather than factual content (Gatt & Krahmer, 2018b). In addition, most automatic metrics do not correlate with faithfulness or factuality, making them often unreliable. Human evaluation on factuality datasets (Abhishek et al., 2022; Zhou et al., 2021) is the preferred method (Maynez et al., 2020; Goyal & Durrett, 2020; Falke et al., 2019; Ji et al., 2023). Goyal & Durrett (2020) present a textual entailment system that can fact-check generated text and localize the factual error. The model-based metric BARTScore++ can detect hallucinations to a certain degree because it imitates human-like error analysis (Lu et al., 2023a). Fabbri et al. (2022) propose QAFactEval to evaluate factual consistency for the QA task. Madaan et al. (2021) propose a model to automatically generate counterfactual text that can be leveraged to create a synthetic test set on factuality.

5.4 Misuse

It is important to name the adversarial risks of text-generative models. Only when knowing how these systems can be misused can countermeasures be researched. This is also referred to as threat modeling (Zellers et al., 2020), i.e., identifying threats and vulnerabilities from an adversary's point of view. We identify three main categories of how text generation systems are misused currently or in the near future. These are *non-disclosed*, *misaligned*, and *adversarial* usage.

Machine-generated systems can pose a threat if their usage is *not disclosed* properly. Convincing machine-generated text poses a significant challenge to academic integrity (Cotton et al., 2024). Additionally, highly controllable text generation systems can flood social media (Zellers et al., 2020; Tourille et al., 2022) or product pages (Adelani et al., 2020) with intentionally biased and non-factual information (fake news, propaganda, fake product reviews). The main mitigation methods for undisclosed usage concentrate on discriminating machine-generated text from human-written text (Tourille et al., 2022; Wahle et al., 2022b, 2021; Fagni et al., 2021; Munir et al., 2021), although attribution alone does not verify a text's factuality (Schuster et al., 2020). Classifiers for machine-generated text suffer from regular false positive detections for many reasons, ranging from shortness to incoherence (Jawahar et al., 2020). Perkins et al. (2024) rely on well-trained academic staff using intricate annotation software to identify generated content. A RoBERTa-based detector achieves notable results on a dataset comprising human- and machine-generated data (Maktabdar Oghaz et al., 2025). Still, reliable detection is especially difficult when advanced prompting techniques are used (Perkins et al., 2024). An adversary using multiple text generators within the same document might pose an additional challenge for detectors (Uchendu et al., 2021). Rodriguez et al. (2022) suggest that paragraph-level detectors can be used to detect the tampering of full-length documents. GLTR utilizes the fact that human writing includes more sequentially improbable tokens in a sequence than machine text and highlights tokens probabilities (Gehrmann et al., 2019). Zhong et al. (2020) specifically assess the factual structure of a text to detect machine samples. Zhang et al. (2024) employ a hybrid detector, merging conventional TF-IDF strategies with Bayesian classifiers, stochastic gradient descent, categorical gradient boosting, and 12 instances of LMs. Ni et al. (2025) develop a multi-view attention mechanism to detect fake news that attends to both the semantics and propagation structure of social media content.

Users of text generation systems might have harmful intentions and specifically tailor prompts that lead to text generation that is *not aligned* with human ethics or safety regulations. To mitigate *misaligned* use, many models like GPT-4 (Achiam et al., 2024), LLaMA 2 (Touvron et al., 2023), or Gemini (Comanici et al., 2025) include automated censorship mechanisms that block potentially harmful content. For example, Gemini (Comanici et al., 2025) leverages advanced Chain-of-Thought recipes to align with safety policies. GPT-4 (Achiam et al., 2024) provides an additional reward signal during RLHF fine-tuning (Ouyang et al., 2022) that targets correct behavior, such as refusing to give harmful responses. However, these mechanisms can be tricked. For example, Jiang et al. (2024) leverage the poor performance of many modern LMs in recognizing ASCII art to jailbreak safety-aligned LMs.

A more technical misuse of machine-generated text is *adversarial*. As text classification systems are sensitive to certain words, an adversary could intentionally generate text including certain words, ultimately hindering the effectiveness of classification models (Mosca et al., 2022; Gao et al., 2018; Jin et al., 2020). Due to the high instruction-following capabilities of modern LMs (Ouyang et al., 2022), an additional risk arises from prompt injections. Prompt injections describe secretly hidden text snippets (prompts) in any content used by an LM-based application (Greshake et al., 2023). They can lead to the extraction of sensitive information from LMs (e.g., personal data, credentials) (Greshake et al., 2023). Prompt injections can also be used for phishing, scams, or masquerading, and some systems might be vulnerable to unwanted API calls. To avoid tailored texts tricking classifiers, Mosca et al. (2022) propose a logits-based metric that captures words with a suspiciously high impact on a classifier. The risks of text-generative models are abundant, and mitigation methods must be tailored to

each problem. Meanwhile, we emphasize raising awareness of the diverse problems to foster the development of mitigation methods.

5.5 Datasets

The data scale has increased significantly since contemporary LMs need to be pre-trained on large amounts of web-scraped data. As corpora grow, finding and filtering open-source and high-quality data becomes more challenging, which might encourage exploring licensed data for training (e.g., transcribed YouTube videos). Also, many datasets and their sources are unstandardized (Fatima et al., 2022).

Fleisig et al. (2023) make several suggestions on how to improve the quality of datasets. Datasets should clarify the potential risks (e.g., toxicity (Gehman et al., 2020)) of the provided machine-generated text. With high-quality annotations, authors can measure or mitigate a diverse set of fairness-related harms, and identify the demographic groups of annotators (Fleisig et al., 2023). In addition, datasets should account for annotator disagreement, providing individual annotators' judgments rather than a final score. Context-dependent harms can only be identified when the context is provided in the dataset (e.g., perceived author of text, application), and a diverse set of annotators can ensure that various demographic groups are represented.

A key problem related to data scarcity concerns multilingual applications. Text generation works best for languages encountered frequently during pre-training. However, most text generation research takes place in the United States, followed by economically strong countries such as China, the United Kingdom, and India (Ramos et al., 2023; Abdalla et al., 2023). Thus, most research is performed on corresponding corpora, mostly in English or Chinese. Low-resource languages remain underexplored in machine-generated text, and multi-lingual models must be specifically trained to fit the niche (Shaham et al., 2024).

To enhance a model's performance on scarce data like low-resource languages, researchers apply variations to the training process. Shaham et al. (2024) show that just 40 multilingual samples in an English tuning set improve multi-lingual instruction following. This also generalizes to unseen languages during fine-tuning, indicating that including two to four languages in the training set significantly improves cross-lingual generalization. Datasets should not be used repeatedly as a benchmark, as overfitting reduces their efficiency long term (Fleisig et al., 2023). Thus, there exists a constant need for new datasets. One example dataset is XWikiRef (Taunk et al., 2023), a Wikipedia summarization dataset on eight low-resource languages across five domains (i.e., books, films, politicians, sportsmen, and writers). We highlight the importance of data-efficient methodologies to improve the utility of text generation systems across languages, dialects, and sociolects.

5.6 Interpretability

A text generation model is interpretable when the user can understand why a certain output was produced. Modern LMs inherently do not provide intuitive insight into how textual outputs are produced, as their weights and representations appear mathematically complicated (Lu et al., 2022a).

Interpretability is a desired goal when solving tasks because, without the reasoning behind the final output being clear, the correctness of the output can not be easily assessed, and users need to rely solely on trusting the model. Moreover, researchers rely on the intricate analysis of existing models to propose novel ideas.

Singh et al. (2024) point out that, overall, two aspects of interpretation research need to be explained. First, we require an intricate explanation of an LM's behavior. This comprises local characteristics like feature attribution and data grounding and global characteristics like attention head importance and data influence. Second, a clear explanation of used datasets is required. This includes interactive clarification using natural language and aiding data analysis. Tenney et al. (2020) propose a tool to visualize an LMs' behavior on the architectural level. They visualize the sentence embeddings, classification probabilities, attention, and confusion matrix. By providing the information in a visual format that is easy to understand, one can directly assess how an LM behaves when

specific data points are altered. Ultimately, tools like this provide a meaningful step towards interpretability, as models' inner processes usually remain hidden and their disclosure requires time-consuming development of custom solutions.

5.7 Transparency

Transparency involves two major aspects. First, transparency refers to the lucid explanation and documentation of proposed models and datasets. This includes the research along with their underlying methodologies, which many newly released models like GPT-4 (Achiam et al., 2024) and Gemini (Comanici et al., 2025) only provide superficially. As the quality of AI-authored scientific work remains subpar relative to human-authored research (Amirjalili et al., 2024), especially in terms of factual accuracy and complex aspects of authorship, the disclosure of its use in academic writing also becomes essential. The indiscriminate use of machine-generated text during training introduces irreparable defects into the resulting models, given the recursive incorporation of machine-generated data (Shumailov et al., 2024). So-called model collapse could be avoided by fully disclosing the use of AI, although this can be considered an ambitious solution.

Since big tech companies increasingly lead and, consequently, shift NLP research directions, the discussion of potential risks (e.g., toxicity, factuality) has become essential in recent years, especially when releasing LMs (Abdalla et al., 2023). To research and address the emerging challenges, the transparent reporting of model characteristics is of key importance.

For this, Mitchell et al. (2019) provide an extensive and standardized model card that can be shipped together with novel LMs. However, corporate organizations often have a monetary interest in releasing new models, and fully disclosing all model details can conflict with a company's interest in not having their product replicated or released to an open-source audience. Aczel & Wagenmakers (2023) describe the transparent and credible usage of text-generative models in scientific writing as threefold. First, researchers should disclose the model and its training data used. In addition, the generated text, or a summary of it, should be available to the readers. Second, the authors should specify for which aspects of a scientific work (outline, writing, code) a model was used. Third, researchers are required to always verify generated text which includes a check on plagiarism. Fulfilling all aspects described by Aczel & Wagenmakers (2023) can be difficult, but the guidelines give an idea of the ideal process for documenting AI usage. Wahle et al. (2023c) propose AI Usage Cards, a shorter card that can be included in research papers to document how text generation models are used for the work.

5.8 Privacy

The data collection process raises privacy concerns as there is apprehension about companies using human inputs to their system to continuously train their models (Wu et al., 2024). Modern LMs require large amounts of data crawled from the web for their pre-training (Touvron et al., 2023) and it is hard to control whether personal data is being used. These models have been found to output memorized passages from their training data (Yuan et al., 2022; Carlini et al., 2021). Thus, LMs could also leak private information they encountered during training (Li et al., 2021; Greshake et al., 2023).

Li et al. (2023a) show how multi-step jailbreaking prompts can be formulated to expose private information like names or emails, even bypassing safety modules employed in applications like ChatGPT (Yuan et al., 2022). Examples like this show that privacy-related research about text generation is crucial.

Yue et al. (2023) use differential privacy to mathematically guarantee that private information can neither be generated nor reconstructed from the output. The key idea of differential privacy is that including or excluding a single individual's data should not significantly affect the model's output. By understanding the sensitivity regarding single data points, the algorithm knows how much noise to add to a stochastic gradient descent algorithm. Once the noise has been added, further data processing will not diminish privacy protection. This

property is crucial because it ensures privacy is not compromised through additional analysis. However, their mathematical guarantee of privacy protection comes with a trade-off regarding output quality. Notably, tight differential privacy harms small classes like minority groups during training as rarely mentioned groups could be nullified in the process. Thus, one needs to find the right balance between privacy and quality in real-world applications. Song & Shmatikov (2019) develop an auditing model that lets users check whether their data has been used to train a text generation model.

5.9 Computing

LMs have been growing increasingly large (Zhao et al., 2023) and with more data and a higher number of model parameters, training times also surge. This requires more computing and energy to publish a novel state-of-the-art model. Nowadays, this becomes a problem for small companies and research organizations as only big tech companies like OpenAI, Google, or Meta have the resources to train a competitive language model from the ground up (Abdalla et al., 2023). Larger models also have a higher environmental cost, emitting more carbon during training than smaller models (Touvron et al., 2023; Fabbri et al., 2022). Reducing the computational requirements of recent LMs could potentially lower their environmental impact due to lower power consumption.

The scaling laws from (Kaplan et al., 2020) suggest that larger models will continue to perform better than smaller ones while reaching the same level of performance with fewer optimization steps. They also find that a model's performance depends strongly on the number of parameters, dataset size, and the amount of computing, while architectural parameters like layer depth or width are not as important. Thus, making text generation models lightweight and improving computing times is an active research topic. Liu et al. (2023) propose a transformer with ternary (i.e., -1, 0, 1) or binary (i.e., 0, 1) weights, which facilitate multiplication-free computations. Division-of-Thoughts distributes a reasoning process across a locally deployed, smaller-scale LM and cloud-based LLMs (Shao et al., 2025), reducing reasoning time and API costs. EdgeShard deploys LLMs across distributed services, accounting for device heterogeneity and bandwidth limitations (Zhang et al., 2025). Their variation of BART (Lewis et al., 2020a) is up to 16 times more efficient than its real-valued counterpart while achieving close to normal performance. However, binarization and ternarization require bit-packing to have actual memory savings and need dedicated hardware support for real-time acceleration. Li & Liang (2021) propose prefix-tuning, a light-weight alternative to fine-tuning where only 0.1% of a models parameters are optimized. Their model performs comparable to a full-data setting and is even better in low-data settings and extrapolation on unseen topics. Approaches to reduce LMs' environmental impact are often not included in the works we read. We identify that NLP-focused research on reducing environmental harm is scarce but necessary if the field wants sustainable future research.

6 Epilogue

In this literature review, we covered 257 publications, focusing on recent developments in text generation from the Transformer architecture in 2017 through December 2025. In that context, we systematically identified and discussed the core tasks and sub-tasks in the field, assessed how text generation systems are evaluated, and outlined relevant challenges that researchers can address in the near and mid-term future. To summarize this work, we revisit our initial research questions.

What constitutes the task of text generation? What are the main sub-tasks?

We categorized text generation into five main sub-tasks: *open-ended text generation*, *summarization*, *translation*, *paraphrasing*, and *question answering*. We identified prominent sub-tasks and unique challenges to display recent research within each of these. *Open-ended text generation* comprises open-domain applications such as prompted instruction-following systems, story generation, and dialogue generation. Open-ended tasks are inherently challenged by trying to consistently model long contexts, especially when considering sub-tasks that are lexically

and semantically diverse, like dialogue generation or story generation. In *summarization*, we focused on single documents, multiple documents, and dialogues. Recent studies concentrate on sub-tasks like multi-document summarization and dialogue summarization, which exhibit higher contextual complexity. While sentence-level *translation* is a well-established task, document-level translation poses a chance to mitigate limitations like word ambiguities within or between documents. *Paraphrasing* research shifts from uncontrolled generation towards more controllable scenarios. Here, we noticed a lack of aligned paraphrasing datasets that provide labels for such controllable attributes. *Question answering* relies on internal and external knowledge sources and is challenged by reasoning, non-answerability, and non-factoid questions.

How are text generation systems evaluated? What are the concomitant limitations?

We identified two main evaluation methodologies: model-free and model-based. While model-free evaluations are based on static and rule-based algorithms, model-based ones leverage word or sentence embeddings of pre-trained models to produce a score. Most model-free metrics rely on n-gram overlap, but their correlation to quality as humans perceive it varies. Moreover, some textual characteristics, like factuality or bias, are difficult to measure using model-free metrics, which is why we emphasized the need for additional model-based metrics and human evaluation. In addition to quality, model-based metrics can capture attributes like factuality or writing style by fine-tuning or prompting. However, as neural models are subject to errors, model-based metrics might introduce various biases and fail to reliably quantify these attributes, while also lacking interpretability. LLM-as-a-Judge provides a scalable evaluation alternative with strong agreement to human judgments on simple tasks, but suffers from systematic biases and underperforms humans on complex NLP tasks. While many studies use human labels for evaluation, they often omit measures of inter-annotator agreement or label reliability. We encourage more researchers to consider agreement and reliability in the design phase of human studies. For this, we point out helpful metrics, methodologies, and labeling tools to improve the quality of produced annotations.

What are the open challenges in text generation?

We covered *bias*, *reasoning*, *hallucinations*, *misuse*, *datasets*, *interpretability*, *transparency*, *privacy*, and *computing*. In this work, we identified three prominent *bias* types relevant to text generation: exposure-, allocation-, and representation bias. Recent works are concerned with bias measurement and mitigation in generated texts. External modules that quantify and detect *hallucinations* can provide transparency and improve factuality in text generation. Specific prompting techniques like Chain-of-Thought stimulate language models' *reasoning* capabilities, but they require an intricate prompt design. We outlined various scenarios of how text generation could potentially be *misused*. These scenarios are categorized into non-disclosed, misaligned, and adversarial usage. Additionally, existing *datasets* for text generation tend to disregard low-resource languages and are sometimes not standardized into a uniform format or lack metadata like inter-annotator agreement scores. *Interpretability* in text generation should consider both the used models and associated datasets. Differential *privacy* can mitigate the risk of leaking private information from training data at the cost of output quality. To make capable language models more accessible to the community, we find incipient research about how to make the *computing* requirements of text generation systems lightweight without sacrificing substantial amounts of performance. This involves both the training process and the inference of models.

What are prominent research directions in text generation?

Recent publications have focused more on improving the intrinsic (e.g., bias, factuality) and contextual (e.g., relevance, completeness) quality of the machine-generated text. We suspect research in these areas will be particularly fruitful, and we highlighted various directions accordingly. For example, we identified that many bias mitigation methods depend on careful data selection and augmentation while only working for specific bias types. Research towards more universal mitigation techniques, possibly leveraging self-aware language models to quantify their own biases, would provide an impactful contribution to the community. Improvements

in factuality are challenged by ponderous evaluation. To foster additional research in that area, we suggested investigating the effective quantification of hallucinations using model-based methods. Aside from prompting strategies, enhancing language models' reasoning capabilities could be considered during a model's training instead of during inference. With language model applications permeating people's everyday lives, subordinate problems like its misuse become apparent. To ensure the safety and responsible use of these text-generative models, safety-aligned systems should be able to detect when they are about to be exploited by misaligned or adversarial prompts. In addition, responsible individuals and companies are required to establish safeguards to prevent the abuse of text-generation systems. Excluding private information from both the training process of language models and the generation of text poses another opportunity for research. Compared to more established tasks like uncontrolled paraphrasing or single-document summarization, contemporary proposals suggest that the community's general interest shifted towards more specific scenarios (e.g., controlled paraphrasing) and sub-tasks that are contextually more complex (e.g., dialogue summarization). We suggest that such scenarios will be more prominently researched in subsequent works. Additionally, we identified the potential to use large pre-trained models that can operate across a variety of domains and tasks. Future work could focus on more effectively leveraging the internal knowledge of such models for tasks like question answering while avoiding hallucinated content. In summary, we discussed several sub-tasks of text generation that leave space for future research and outlined key challenges that can be targeted in the near and mid-term future.

6.1 Limitations

Text generation can involve various other modalities that are not immediately natural language. For example, methods include table generation (Lu et al., 2022b), table-to-text generation (Wang et al., 2020), sql-to-text generation (Xu et al., 2018), or text generation from an abstract meaning representation (Mager et al., 2020). In this work, we focus on text generation from natural language input because modalities such as images (Tian et al., 2024a) require specialized techniques and are therefore out of scope for this review. Some recently published works may not have passed the filters of our systematic review because they might not have had enough time to accumulate enough citations. Ultimately, this is a limitation of our bottom-up approach, as some subfields might not be captured by this systematic search. We tried to mitigate this by mentioning related subfields where appropriate. We release the manual relevance judgments to the public through our GitHub repository[1]. When retrieving papers, we rely on Semantic Scholar's set of indexed papers and ranking algorithm, which might lead to some papers being left out, although Semantic Scholar's coverage, particularly in AI and NLP, is high. We detail the relevance function used by Semantic Scholar for proper documentation of the process.

Acknowledgments

This work was supported by the Lower Saxony Ministry of Science and Culture and the VW Foundation. This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 564661959.

References

Abdalla, M., Wahle, J. P., Ruas, T., Névéol, A., Ducel, F., Mohammad, S., & Fort, K. (2023). The Elephant in the Room: Analyzing the Presence of Big Tech in Natural Language Processing Research. In Rogers, A., Boyd-Graber, J., & Okazaki, N., editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13141–13160, Toronto, Canada. Association for Computational Linguistics.

Abhishek, T., Sagare, S., Singh, B., Sharma, A., Gupta, M., & Varma, V. (2022). XAlign: Cross-lingual Fact-to-Text Alignment and Generation for Low-Resource Languages. In *Companion Proceedings of the Web Conference 2022*. ACM.

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S. et al. (2024). GPT-4 Technical Report.

Aczel, B. & Wagenmakers, E.-J. (2023). Transparency Guidance for ChatGPT Usage in Scientific Writing.

Adelani, D. I., Mai, H., Fang, F., Nguyen, H. H., Yamagishi, J., & Echizen, I. (2020). Generating Sentiment-Preserving Fake Online Reviews Using Neural Language Models and Their Human- and Machine-Based Detection. In *Advanced Information Networking and Applications*, pp. 1341–1354. Springer International Publishing.

Alabdulkarim, A., Li, S., & Peng, X. (2021). Automatic Story Generation: Challenges and Attempts.

Alihosseini, D., Montahaei, E., & Soleymani Baghshah, M. (2019). Jointly measuring diversity and quality in text generation models. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pp. 90–98, Minneapolis, Minnesota. Association for Computational Linguistics.

Amirjalili, F., Neysani, M., & Nikbakht, A. (2024). Exploring the boundaries of authorship: A comparative analysis of ai-generated text and human academic writing in english literature. In *Frontiers in Education*, volume 9, p. 1347421. Frontiers Media SA.

Anderson, P., Fernando, B., Johnson, M., & Gould, S. (2016). SPICE: Semantic Propositional Image Caption Evaluation.

Artetxe, M., Goswami, V., Bhosale, S., Fan, A., & Zettlemoyer, L. (2023). Revisiting Machine Translation for Cross-lingual Classification.

Babakov, N., Dale, D., Logacheva, V., & Panchenko, A. (2022). A large-scale computational study of content preservation measures for text style transfer and paraphrase generation. In *Proc. of ACL*, pp. 300–321, Dublin, Ireland. Association for Computational Linguistics.

Bandel, E., Aharonov, R., Shmueli-Scheuer, M., Shnayderman, I., Slonim, N., & Ein-Dor, L. (2022). Quality Controlled Paraphrase Generation. In Muresan, S., Nakov, P., & Villavicencio, A., editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 596–609, Dublin, Ireland. Association for Computational Linguistics.

Banerjee, S. & Lavie, A. (2005). METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In Goldstein, J., Lavie, A., Lin, C.-Y., & Voss, C., editors, *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pp. 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Bao, S., He, H., Wang, F., Wu, H., & Wang, H. (2020). PLATO: Pre-trained Dialogue Generation Model with Discrete Latent Variable.

Barua, A., Ahmed, M. U., & Begum, S. (2023). A systematic literature review on multimodal machine learning: Applications, challenges, gaps and future directions. *IEEE Access*, 11:14804–14831.

Becker, J., Kaesberg, L. B., Bauer, N., Wahle, J. P., Ruas, T., & Gipp, B. (2025a). MALLM: Multi-agent large language models framework. In Habernal, I., Schulam, P., & Tiedemann, J., editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 418–439, Suzhou, China. Association for Computational Linguistics.

Becker, J., Kaesberg, L. B., Stephan, A., Wahle, J. P., Ruas, T., & Gipp, B. (2025b). Stay focused: Problem drift in multi-agent debate.

Becker, J., Wahle, J. P., Ruas, T., & Gipp, B. (2023). Paraphrase Detection: Human vs. Machine Content.

Belém, C. G., Pezeshkpour, P., Iso, H., Maekawa, S., Bhutani, N., & Hruschka, E. (2025). From single to multi: How LLMs hallucinate in multi-document summarization. In Chiruzzo, L., Ritter, A., & Wang, L., editors, *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 5276–5309, Albuquerque, New Mexico. Association for Computational Linguistics.

Beltagy, I., Peters, M. E., & Cohan, A. (2020). Longformer: The Long-Document Transformer.

Bengio, Y., Ducharme, R., & Vincent, P. (2000). A Neural Probabilistic Language Model. In *Advances in Neural Information Processing Systems*, volume 13. MIT Press.

Bhandari, M., Gour, P. N., Ashfaq, A., Liu, P., & Neubig, G. (2020). Re-evaluating evaluation in text summarization. In *Proc. of EMNLP*, pp. 9347–9359. Association for Computational Linguistics.

Bingli, L. & Vargas, D. V. (2025). Toward immersive computational storytelling: Card-framework for enhanced persona-driven dialogues. *IEEE Transactions on Games*, 17(2):384–396.

Botha, J. A., Faruqui, M., Alex, J., Baldridge, J., & Das, D. (2018). Learning To Split and Rephrase From Wikipedia Edit History. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 732–737, Brussels, Belgium. Association for Computational Linguistics.

Bražinskas, A., Lapata, M., & Titov, I. (2020). Few-shot learning for opinion summarization. In *Proc. of EMNLP*, pp. 4119–4135. Association for Computational Linguistics.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A. et al. (2020). Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc.

Cai, L.-Q., Wei, M., Zhou, S.-T., & Yan, X. (2020). Intelligent Question Answering in Restricted Domains Using Deep Learning and Question Pair Matching. *IEEE Access*, 8:32922–32934.

Callison-Burch, C., Fordyce, C., Koehn, P., Monz, C., & Schroeder, J. (2008). Further meta-evaluation of machine translation. In *Proceedings of the Third Workshop on Statistical Machine Translation - StatMT '08*, pp. 70–106, Columbus, Ohio. Association for Computational Linguistics.

Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U. et al. (2021). Extracting Training Data from Large Language Models.

Chen, M., Tang, Q., Wiseman, S., & Gimpel, K. (2019). Controllable paraphrase generation with a syntactic exemplar. In *Proc. of ACL*, pp. 5972–5984, Florence, Italy. Association for Computational Linguistics.

Chen, X., Jia, S., & Xiang, Y. (2020). A review: Knowledge reasoning over knowledge graph. *Expert Systems with Applications*, 141:112948.

Chung, J., Kamar, E., & Amershi, S. (2023). Increasing Diversity While Maintaining Accuracy: Text Data Generation with Large Language Models and Human Interventions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Clark, E., August, T., Serrano, S., Haduong, N., Gururangan, S., & Smith, N. A. (2021). All That's 'Human' Is Not Gold: Evaluating Human Evaluation of Generated Text. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 7282–7296. Association for Computational Linguistics.

Clark, E., Celikyilmaz, A., & Smith, N. A. (2019). Sentence Mover's Similarity: Automatic Evaluation for Multi-Sentence Texts. In Korhonen, A., Traum, D., & Màrquez, L., editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2748–2760, Florence, Italy. Association for Computational Linguistics.

Comanici, G., Bieber, E., Schaekermann, M., Pasupat, I., Sachdeva, N., Dhillon, I., Blistein, M., Ram, O., Zhang, D., Rosen, E. et al. (2025). Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities.

Costa, F., Ouyang, S., Dolog, P., & Lawlor, A. (2018). Automatic Generation of Natural Language Explanations. In *Proceedings of the 23rd International Conference on Intelligent User Interfaces Companion*. ACM.

Cotton, D., Cotton, P., & Shipway, J. R. (2024). Chatting and Cheating. Ensuring academic integrity in the era of ChatGPT.

Currey, A., Mathur, P., & Dinu, G. (2020). Distilling Multiple Domains for Neural Machine Translation. In Webber, B., Cohn, T., He, Y., & Liu, Y., editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4500–4511. Association for Computational Linguistics.

Dagdelen, J., Dunn, A., Lee, S., Walker, N., Rosen, A. S., Ceder, G., Persson, K. A., & Jain, A. (2024). Structured information extraction from scientific text with large language models. *Nature Communications*, 15(1):1418.

Deng, M., Tan, B., Liu, Z., Xing, E., & Hu, Z. (2021). Compression, transduction, and creation: A unified framework for evaluating natural language generation. In *Proc. of EMNLP*, pp. 7580–7605, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Dennett, D. C. (2013). The Role Of Language In Intelligence. In Burri, A., editor, *Sprache und Denken / Language and Thought*, pp. 42–55. De Gruyter.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dhamala, J., Sun, T., Kumar, V., Krishna, S., Pruksachatkun, Y., Chang, K.-W., & Gupta, R. (2021). BOLD. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. ACM.

Ding, H. & Xu, X. (2023). SAN-T2T: An automated table-to-text generator based on selective attention network. *Natural Language Engineering*, pp. 1–25.

Doddington, G. (2002). Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research* -, p. 138, San Diego, California. Association for Computational Linguistics.

Dolan, W. B. & Brockett, C. (2005). Automatically Constructing a Corpus of Sentential Paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Dou, Y., Forbes, M., Koncel-Kedziorski, R., Smith, N. A., & Choi, Y. (2022). Is GPT-3 text indistinguishable from human text? scarecrow: A framework for scrutinizing machine text. In *Proc. of ACL*, pp. 7250–7274, Dublin, Ireland. Association for Computational Linguistics.

Du, X., Shao, J., & Cardie, C. (2017). Learning to ask: Neural question generation for reading comprehension. In *Proc. of ACL*, pp. 1342–1352, Vancouver, Canada. Association for Computational Linguistics.

Durmus, E., He, H., & Diab, M. (2020). FEQA: A Question Answering Evaluation Framework for Faithfulness Assessment in Abstractive Summarization. In Jurafsky, D., Chai, J., Schluter, N., & Tetreault, J., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 5055–5070. Association for Computational Linguistics.

Egonmwan, E. & Chali, Y. (2019). Transformer and seq2seq model for Paraphrase Generation. In Birch, A., Finch, A., Hayashi, H., Konstas, I., Luong, T., Neubig, G., Oda, Y., & Sudoh, K., editors, *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pp. 249–255, Hong Kong. Association for Computational Linguistics.

Erdem, E., Kuyu, M., Yagcioglu, S., Frank, A., Parcalabescu, L., Plank, B., Babii, A., Turuta, O., Erdem, A., Calixto, I. et al. (2022). Neural natural language generation: A survey on multilinguality, multimodality, controllability and learning. *J. Artif. Int. Res.*, 73.

Fabbri, A., Li, I., She, T., Li, S., & Radev, D. (2019). Multi-News: A Large-Scale Multi-Document Summarization Dataset and Abstractive Hierarchical Model. In Korhonen, A., Traum, D., & Màrquez, L., editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1074–1084, Florence, Italy. Association for Computational Linguistics.

Fabbri, A., Wu, C.-S., Liu, W., & Xiong, C. (2022). QAFactEval: Improved QA-based factual consistency evaluation for summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2587–2601, Seattle, United States. Association for Computational Linguistics.

Fabbri, A. R., Kryściński, W., McCann, B., Xiong, C., Socher, R., & Radev, D. (2021). SummEval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Fagni, T., Falchi, F., Gambini, M., Martella, A., & Tesconi, M. (2021). TweepFake: About detecting deepfake tweets. *PLOS ONE*, 16(5):e0251415.

Falke, T., Ribeiro, L. F. R., Utama, P. A., Dagan, I., & Gurevych, I. (2019). Ranking Generated Summaries by Correctness: An Interesting but Challenging Application for Natural Language Inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2214–2220, Florence, Italy. Association for Computational Linguistics.

Fan, A., Jernite, Y., Perez, E., Grangier, D., Weston, J., & Auli, M. (2019). ELI5: Long Form Question Answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3558–3567, Florence, Italy. Association for Computational Linguistics.

Fan, A., Lewis, M., & Dauphin, Y. (2018). Hierarchical neural story generation. In *Proc. of ACL*, pp. 889–898, Melbourne, Australia. Association for Computational Linguistics.

Fatima, N., Imran, A. S., Kastrati, Z., Daudpota, S. M., & Soomro, A. (2022). A Systematic Literature Review on Text Generation Using Deep Neural Network Models. *IEEE Access*, 10:53490–53503.

Feng, X., Feng, X., Qin, L., Qin, B., & Liu, T. (2021). Language model as an annotator: Exploring DialoGPT for dialogue summarization. In *Proc. of ACL*, pp. 1479–1491. Association for Computational Linguistics.

Fleisig, E., Amstutz, A., Atalla, C., Blodgett, S. L., III, H. D., Olteanu, A., Sheng, E., Vann, D., & Wallach, H. (2023). FairPrism: Evaluating Fairness-Related Harms in Text Generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Freitag, M., Foster, G., Grangier, D., Ratnakar, V., Tan, Q., & Macherey, W. (2021). Experts, Errors, and Context: A Large-Scale Study of Human Evaluation for Machine Translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

Gallegos, I. O., Rossi, R. A., Barrow, J., Tanjim, M. M., Kim, S., Dernoncourt, F., Yu, T., Zhang, R., & Ahmed, N. K. (2024). Bias and Fairness in Large Language Models: A Survey. *Computational Linguistics*, pp. 1–83.

Gambhir, M. & Gupta, V. (2017). Recent automatic text summarization techniques: a survey. *Artificial Intelligence Review*, 47(1):1–66.

Gao, J., Lanchantin, J., Soffa, M. L., & Qi, Y. (2018). Black-Box Generation of Adversarial Text Sequences to Evade Deep Learning Classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE.

Gao, Y., Bing, L., Chen, W., Lyu, M. R., & King, I. (2019). Difficulty controllable generation of reading comprehension questions. In Kraus, S., editor, *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pp. 4968–4974. ijcai.org.

Garg, S., Prabhu, S., Misra, H., & Srinivasaraghavan, G. (2021). Unsupervised Contextual Paraphrase Generation using Lexical Control and Reinforcement Learning.

Gatt, A. & Krahmer, E. (2018a). Survey of the state of the art in natural language generation: core tasks, applications and evaluation. *J. Artif. Int. Res.*, 61(1):65–170.

Gatt, A. & Krahmer, E. (2018b). Survey of the State of the Art in Natural Language Generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 61:65–170.

Gehman, S., Gururangan, S., Sap, M., Choi, Y., & Smith, N. A. (2020). RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 3356–3369. Association for Computational Linguistics.

Gehrmann, S., Dai, F., Elder, H., & Rush, A. (2018). End-to-end content and plan selection for data-to-text generation. In *Proceedings of the 11th International Conference on Natural Language Generation*, pp. 46–56, Tilburg University, The Netherlands. Association for Computational Linguistics.

Gehrmann, S., Strobelt, H., & Rush, A. (2019). GLTR: Statistical detection and visualization of generated text. In *Proc. of ACL*, pp. 111–116, Florence, Italy. Association for Computational Linguistics.

Gekhman, Z., David, E. B., Orgad, H., Ofek, E., Belinkov, Y., Szpektor, I., Herzig, J., & Reichart, R. (2025). Inside-out: Hidden factual knowledge in llms.

Ghosh, S., Chollet, M., Laksana, E., Morency, L.-P., & Scherer, S. (2017). Affect-LM: A neural language model for customizable affective text generation. In *Proc. of ACL*, pp. 634–642, Vancouver, Canada. Association for Computational Linguistics.

Gira, M., Zhang, R., & Lee, K. (2022). Debiasing pre-trained language models via efficient fine-tuning. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, pp. 59–69, Dublin, Ireland. Association for Computational Linguistics.

Gliwa, B., Mochol, I., Biesek, M., & Wawer, A. (2019). SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In Wang, L., Cheung, J. C. K., Carenini, G., & Liu, F., editors, *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pp. 70–79, Hong Kong, China. Association for Computational Linguistics.

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

Goodrich, B., Rao, V., Liu, P. J., & Saleh, M. (2019). Assessing the factual accuracy of generated text. In Teredesai, A., Kumar, V., Li, Y., Rosales, R., Terzi, E., & Karypis, G., editors, *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pp. 166–175. ACM.

Goyal, R., Kumar, P., & Singh, V. P. (2023). A Systematic survey on automated text generation tools and techniques: application, evaluation, and challenges. *Multimedia Tools and Applications*.

Goyal, T. & Durrett, G. (2020). Evaluating factuality in generation with dependency-level entailment. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 3592–3603. Association for Computational Linguistics.

Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Holz, T., & Fritz, M. (2023). Not What You've Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*, pp. 79–90, Copenhagen Denmark. ACM.

Grusky, M. (2023). Rogue Scores. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Gu, Y., Feng, X., Ma, S., Wu, J., Gong, H., & Qin, B. (2022). Improving controllable text generation with position-aware weighted decoding. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 3449–3467, Dublin, Ireland. Association for Computational Linguistics.

Guan, J., Mao, X., Fan, C., Liu, Z., Ding, W., & Huang, M. (2021). Long text generation by modeling sentence-level and discourse-level coherence. In *Proc. of ACL*, pp. 6379–6393. Association for Computational Linguistics.

Harris, Z. S. (1954). Distributional Structure. *WORD*, 10(2-3):146–162.

He, J., Peng, B., Liao, Y., Liu, Q., & Xiong, D. (2025). Tgea: An error-annotated dataset and benchmark tasks for text generation from pretrained language models.

He, T., Zhang, J., Wang, T., Kumar, S., Cho, K., Glass, J., & Tsvetkov, Y. (2023). On the Blind Spots of Model-Based Evaluation Metrics for Text Generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Holtzman, A., Buys, J., Du, L., Forbes, M., & Choi, Y. (2020). The curious case of neural text degeneration. In *International Conference on Learning Representations (ICLR)*.

Huang, J., Li, Y., Ping, W., & Huang, L. (2018). Large margin neural language model. In *Proc. of EMNLP*, pp. 1183–1191, Brussels, Belgium. Association for Computational Linguistics.

Huang, K.-H., Laban, P., Fabbri, A., Choubey, P. K., Joty, S., Xiong, C., & Wu, C.-S. (2024). Embrace divergence for richer insights: A multi-document summarization benchmark and a case study on summarizing diverse information from news articles. In Duh, K., Gomez, H., & Bethard, S., editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 570–593, Mexico City, Mexico. Association for Computational

Linguistics.
Huang, M., Zhu, X., & Gao, J. (2020a). Challenges in Building Intelligent Open-domain Dialog Systems.
Huang, P.-S., Zhang, H., Jiang, R., Stanforth, R., Welbl, J., Rae, J., Maini, V., Yogatama, D., & Kohli, P. (2020b). Reducing sentiment bias in language models via counterfactual evaluation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 65–83. Association for Computational Linguistics.
Jagfeld, G., Jenne, S., & Vu, N. T. (2018). Sequence-to-sequence models for data-to-text natural language generation: Word- vs. character-based processing and output diversity. In *Proceedings of the 11th International Conference on Natural Language Generation*, pp. 221–232, Tilburg University, The Netherlands. Association for Computational Linguistics.
Jawahar, G., Abdul-Mageed, M., & Lakshmanan, V.S., L. (2020). Automatic detection of machine generated text: A critical survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 2296–2309, Barcelona, Spain (Online). International Committee on Computational Linguistics.
Jelinek, F., Mercer, R. L., Bahl, L. R., & Baker, J. K. (2005). Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1):S63.
Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., & Fung, P. (2023). Survey of Hallucination in Natural Language Generation. *ACM Computing Surveys*, 55(12):1–38.
Jiang, F., Xu, Z., Niu, L., Xiang, Z., Ramasubramanian, B., Li, B., & Poovendran, R. (2024). ArtPrompt: ASCII Art-based Jailbreak Attacks against Aligned LLMs.
Jin, D., Jin, Z., Zhou, J. T., & Szolovits, P. (2020). Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 8018–8025. AAAI Press.
Jin, L. & Gildea, D. (2022). Rewarding semantic similarity under optimized alignments for AMR-to-text generation. In *Proc. of ACL*, pp. 710–715, Dublin, Ireland. Association for Computational Linguistics.
Junadhi, J., Agustin, A., Efrizoni, L., Okmayura, F., Habibie, D., & Muslim, M. (2025). Improving evaluation metrics for text summarization: A comparative study and proposal of a novel metric. *Journal of Applied Data Sciences*, 6(2):885–896.
Jurafsky, D. & Martin, J. H. (2024). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.* 3 edition.
Kann, K., Ebrahimi, A., Koh, J., Dudy, S., & Roncone, A. (2022). Open-domain Dialogue Generation: What We Can Do, Cannot Do, And Should Do Next. In Liu, B., Papangelis, A., Ultes, S., Rastogi, A., Chen, Y.-N., Spithourakis, G., Nouri, E., & Shi, W., editors, *Proceedings of the 4th Workshop on NLP for Conversational AI*, pp. 148–165, Dublin, Ireland. Association for Computational Linguistics.
Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., & Amodei, D. (2020). Scaling Laws for Neural Language Models.
Katz, B. (1980). A Three-Step Procedure for Language Generation.
Kawamae, N. (2023). Friendly Conditional Text Generator. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. ACM.
Keneshloo, Y., Shi, T., Ramakrishnan, N., & Reddy, C. K. (2019). Deep Reinforcement Learning for Sequence-to-Sequence Models. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21.
Khandelwal, U., Fan, A., Jurafsky, D., Zettlemoyer, L., & Lewis, M. (2021). Nearest Neighbor Machine Translation.
Kirstein, F., Wahle, J. P., Gipp, B., & Ruas, T. (2025). Cads: A systematic literature review on the challenges of abstractive dialogue summarization. *J. Artif. Int. Res.*, 82.
Kitchenham, B. & Charters, S. (2007). Guidelines for performing Systematic Literature Reviews in Software Engineering. 2.

Kocmi, T., Avramidis, E., Bawden, R., Bojar, O., Dvorkovich, A., Federmann, C., Fishel, M., Freitag, M., Gowda, T., Grundkiewicz, R. et al. (2023). Findings of the 2023 Conference on Machine Translation (WMT23): LLMs Are Here but Not Quite There Yet. In Koehn, P., Haddow, B., Kocmi, T., & Monz, C., editors, *Proceedings of the Eighth Conference on Machine Translation*, pp. 1–42, Singapore. Association for Computational Linguistics.

Koehn, P. & Knowles, R. (2017). Six Challenges for Neural Machine Translation.

Kovatchev, V., Martí, M. A., & Salamó, M. (2018). ETPC - A Paraphrase Identification Corpus Annotated with Extended Paraphrase Typology and Negation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Kumar, A., Ahuja, K., Vadapalli, R., & Talukdar, P. (2020). Syntax-guided controlled generation of paraphrases. *Transactions of the Association for Computational Linguistics*, 8:329–345.

Kumar, V., Ramakrishnan, G., & Li, Y.-F. (2019). Putting the horse before the cart: A generator-evaluator framework for question generation from text. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pp. 812–821, Hong Kong, China. Association for Computational Linguistics.

Kusner, M., Sun, Y., Kolkin, N., & Weinberger, K. (2015). From Word Embeddings To Document Distances. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 957–966. PMLR.

Lan, W., Qiu, S., He, H., & Xu, W. (2017). A Continuously Growing Dataset of Sentential Paraphrases. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1224–1234, Copenhagen, Denmark. Association for Computational Linguistics.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2020a). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proc. of ACL*, pp. 7871–7880. Association for Computational Linguistics.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T. et al. (2020b). Retrieval-augmented generation for knowledge-intensive nlp tasks. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., & Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474. Curran Associates, Inc.

Li, D., Jiang, B., Huang, L., Beigi, A., Zhao, C., Tan, Z., Bhattacharjee, A., Jiang, Y., Chen, C., Wu, T. et al. (2025). From generation to judgment: Opportunities and challenges of LLM-as-a-judge. In Christodoulopoulos, C., Chakraborty, T., Rose, C., & Peng, V., editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 2757–2791, Suzhou, China. Association for Computational Linguistics.

Li, H., Guo, D., Fan, W., Xu, M., Huang, J., Meng, F., & Song, Y. (2023a). Multi-step Jailbreaking Privacy Attacks on ChatGPT.

Li, J., Galley, M., Brockett, C., Gao, J., & Dolan, B. (2016). A Diversity-Promoting Objective Function for Neural Conversation Models. In Knight, K., Nenkova, A., & Rambow, O., editors, *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 110–119, San Diego, California. Association for Computational Linguistics.

Li, J., Tang, T., Zhao, W. X., Nie, J.-Y., & Wen, J.-R. (2022). Pretrained Language Models for Text Generation: A Survey.

Li, J., Tang, T., Zhao, W. X., Nie, J.-Y., & Wen, J.-R. (2024a). Pre-Trained Language Models for Text Generation: A Survey. *ACM Comput. Surv.*, 56(9):230:1–230:39.

Li, J., Tang, T., Zhao, W. X., & Wen, J.-R. (2021). Pretrained Language Model for Text Generation: A Survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization.

Li, X. L., Holtzman, A., Fried, D., Liang, P., Eisner, J., Hashimoto, T., Zettlemoyer, L., & Lewis, M. (2023b). Contrastive Decoding: Open-ended Text Generation as Optimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Li, X. L., Holtzman, A., Fried, D., Liang, P., Eisner, J., Hashimoto, T., Zettlemoyer, L., & Lewis, M. (2023c). Contrastive decoding: Open-ended text generation as optimization. In Rogers, A., Boyd-Graber, J., & Okazaki, N., editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12286–12312, Toronto, Canada. Association for Computational Linguistics.

Li, X. L. & Liang, P. (2021). Prefix-tuning: Optimizing continuous prompts for generation. In *Proc. of ACL*, pp. 4582–4597. Association for Computational Linguistics.

Li, Y. & Zhao, H. (2023). EM Pre-training for Multi-party Dialogue Response Generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Li, Z., Cao, Y., Xu, X., Jiang, J., Liu, X., Teo, Y. S., Lin, S.-W., & Liu, Y. (2024b). Llms for relational reasoning: How far are we? In *Proceedings of the 1st International Workshop on Large Language Models for Code*, LLM4Code '24, p. 119–126, New York, NY, USA. Association for Computing Machinery.

Liang, X., Tang, Z., Li, J., & Zhang, M. (2023). Open-ended Long Text Generation via Masked Language Modeling. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Lin, B. Y., Zhou, W., Shen, M., Zhou, P., Bhagavatula, C., Choi, Y., & Ren, X. (2020). CommonGen: A constrained text generation challenge for generative commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 1823–1840. Association for Computational Linguistics.

Lin, C.-Y. (2004). ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain. Association for Computational Linguistics.

Lin, T., Wang, Y., Liu, X., & Qiu, X. (2021). A Survey of Transformers. *arXiv:2106.04554 [cs]*.

Liu, A., Sap, M., Lu, X., Swayamdipta, S., Bhagavatula, C., Smith, N. A., & Choi, Y. (2021). DExperts: Decoding-time controlled text generation with experts and anti-experts. In *Proc. of ACL*, pp. 6691–6706. Association for Computational Linguistics.

Liu, B., Wei, H., Niu, D., Chen, H., & He, Y. (2020). Asking questions the human way: Scalable question-answer generation from text corpus. In Huang, Y., King, I., Liu, T., & van Steen, M., editors, *Proc. of WWW*, pp. 2032–2043. ACM / IW3C2.

Liu, D. & Demberg, V. (2023). ChatGPT vs human-authored text: Insights into controllable text summarization and sentence style transfer. In Padmakumar, V., Vallejo, G., & Fu, Y., editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pp. 1–18, Toronto, Canada. Association for Computational Linguistics.

Liu, L., Lewis, P., Riedel, S., & Stenetorp, P. (2022). Challenges in generalization in open domain question answering. In Carpuat, M., de Marneffe, M.-C., & Meza Ruiz, I. V., editors, *Findings of the Association for Computational Linguistics: NAACL 2022*, pp. 2014–2029, Seattle, United States. Association for Computational Linguistics.

Liu, Y. & Lapata, M. (2019). Text Summarization with Pretrained Encoders.

Liu, Z., Oguz, B., Pappu, A., Shi, Y., & Krishnamoorthi, R. (2023). Binary and Ternary Natural Language Generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Liu, Z., Xia, X., Treude, C., Lo, D., & Li, S. (2019). Automatic Generation of Pull Request Descriptions. In *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE.

Lu, P., Mishra, S., Xia, T., Qiu, L., Chang, K.-W., Zhu, S.-C., Tafjord, O., Clark, P., & Kalyan, A. (2022a). Learn to Explain: Multimodal Reasoning via Thought Chains for Science Question Answering. *Advances in Neural Information Processing Systems*, 35:2507–2521.

Lu, Q., Ding, L., Xie, L., Zhang, K., Wong, D. F., & Tao, D. (2023a). Toward Human-Like Evaluation for Natural Language Generation with Error Analysis. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Lu, X., Welleck, S., West, P., Jiang, L., Kasai, J., Khashabi, D., Le Bras, R., Qin, L., Yu, Y., Zellers, R. et al. (2022b). NeuroLogic a*esque decoding: Constrained text generation with lookahead heuristics. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 780–799, Seattle, United States. Association for Computational Linguistics.

Lu, Y., Ouyang, S., & Zhou, K. (2023b). Structured Knowledge Grounding for Question Answering.

Ma, C., Zhang, W. E., Guo, M., Wang, H., & Sheng, Q. Z. (2023). Multi-document Summarization via Deep Learning Techniques: A Survey. *ACM Computing Surveys*, 55(5):1–37.

Ma, S., Sun, X., Li, W., Li, S., Li, W., & Ren, X. (2018). Query and output: Generating words by querying distributed word representations for paraphrase generation. In *Proc. of NAACL-HLT*, pp. 196–206, New Orleans, Louisiana. Association for Computational Linguistics.

Ma, W., Du, X., Feng, X., Huang, L., Huang, Y., Zhang, H., Yang, X., Li, B., Feng, X., Liu, T. et al. (2025). One for all: Update parameterized knowledge across multiple models with once edit. In Che, W., Nabende, J., Shutova, E., & Pilehvar, M. T., editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16021–16034, Vienna, Austria. Association for Computational Linguistics.

Madaan, N., Padhi, I., Panwar, N., & Saha, D. (2021). Generate your counterfactuals: Towards controlled counterfactual generation for text. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pp. 13516–13524. AAAI Press.

Mager, M., Fernandez Astudillo, R., Naseem, T., Sultan, M. A., Lee, Y.-S., Florian, R., & Roukos, S. (2020). GPT-too: A language-model-first approach for AMR-to-text generation. In *Proc. of ACL*, pp. 1846–1852. Association for Computational Linguistics.

Maktabdar Oghaz, M., Babu Saheer, L., Dhame, K., & Singaram, G. (2025). Detection and classification of chatgpt-generated content using deep transformer models. *Frontiers in Artificial Intelligence*, 8:1458707.

Maruf, S., Saleh, F., & Haffari, G. (2022). A Survey on Document-level Neural Machine Translation: Methods and Evaluation. *ACM Computing Surveys*, 54(2):1–36.

Maynez, J., Agrawal, P., & Gehrmann, S. (2023). Benchmarking Large Language Model Capabilities for Conditional Generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Maynez, J., Narayan, S., Bohnet, B., & McDonald, R. (2020). On faithfulness and factuality in abstractive summarization. In *Proc. of ACL*, pp. 1906–1919. Association for Computational Linguistics.

Mccowan, I., Carletta, J., Kraaij, W., Ashby, S., Bourban, S., Flynn, M., Guillemot, M., Hain, T., Kadlec, J., Karaiskos, V. et al. (2005). The AMI meeting corpus. *Int'l. Conf. on Methods and Techniques in Behavioral Research*.

Mcdonald, D. (1980). Natural Language Generation as a Process of Decision Making Under Constartints.

McKeown, K. R. (1982). The Text System for Natural Language Generation: An Overview. In *20th Annual Meeting of the Association for Computational Linguistics*, pp. 113–120, Toronto, Ontario, Canada. Association for Computational Linguistics.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space.

Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., Spitzer, E., Raji, I. D., & Gebru, T. (2019). Model Cards for Model Reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pp. 220–229.

Moratanch, N. & Chitrakala, S. (2017). A survey on extractive text summarization. In *2017 International Conference on Computer, Communication and Signal Processing (ICCCSP)*, pp. 1–6.

Mosca, E., Agarwal, S., Rando Ramírez, J., & Groh, G. (2022). "that is a suspicious reaction!": Interpreting logits variation to detect NLP adversarial attacks. In *Proc. of ACL*, pp. 7806–7816, Dublin, Ireland. Association for Computational Linguistics.

Mulla, N. & Gharpure, P. (2023). Automatic question generation: A review of methodologies, datasets, evaluation metrics, and applications. *Progress in Artificial Intelligence*, 12(1):1–32.

Munir, S., Batool, B., Shafiq, Z., Srinivasan, P., & Zaffar, F. (2021). Through the looking glass: Learning to attribute synthetic text generated by language models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 1811–1822. Association for Computational Linguistics.

Munot, N. & S. Govilkar, S. (2014). Comparative Study of Text Summarization Methods. *International Journal of Computer Applications*, 102(12):33–37.

Nallapati, R., Zhou, B., dos Santos, C., Gulcehre, C., & Xiang, B. (2016). Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pp. 280–290, Berlin, Germany. Association for Computational Linguistics.

Narayan, S., Cohen, S. B., & Lapata, M. (2018). Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Nasution, A. H. & Onan, A. (2024). Chatgpt label: Comparing the quality of human-generated and llm-generated annotations in low-resource language nlp tasks. *IEEE Access*, 12:71876–71900.

Naumanen, M. & Tukiainen, M. (2007). Guiding the elderly into the use of computers and internet–lessons taught and learnt. *Proceedings of cognition and exploratory learning in digital age*, pp. 19–27.

Nenkova, A. & Passonneau, R. (2004). Evaluating Content Selection in Summarization: The Pyramid Method. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pp. 145–152, Boston, Massachusetts, USA. Association for Computational Linguistics.

Ng, J.-P. & Abrecht, V. (2015). Better Summarization Evaluation with Word Embeddings for ROUGE. In Màrquez, L., Callison-Burch, C., & Su, J., editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1925–1930, Lisbon, Portugal. Association for Computational Linguistics.

Ni, J. & McAuley, J. (2018). Personalized review generation by expanding phrases and attending on aspect-aware representations. In *Proc. of ACL*, pp. 706–711, Melbourne, Australia. Association for Computational Linguistics.

Ni, S., Li, J., & Kao, H.-Y. (2025). Mvan: Multi-view attention networks for fake news detection on social media.

Nishida, K., Nishida, K., Saito, I., Asano, H., & Tomita, J. (2020). Unsupervised Domain Adaptation of Language Models for Reading Comprehension. In Calzolari, N., Béchet, F., Blache, P., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Isahara, H., Maegaard, B., Mariani, J. et al., editors, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pp. 5392–5399, Marseille, France. European Language Resources Association.

NIST Multimodal Information Group (2010). NIST 2005 Open Machine Translation (OpenMT) Evaluation.

NIST Multimodal Information Group (2013). NIST 2012 Open Machine Translation (OpenMT) Evaluation.

Okoli, C. (2015). A Guide to Conducting a Standalone Systematic Literature Review. *Communications of the Association for Information Systems*, 37.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A. et al. (2022). Training language models to follow instructions with human feedback.

Pagnoni, A., Balachandran, V., & Tsvetkov, Y. (2021). Understanding Factuality in Abstractive Summarization with FRANK: A Benchmark for Factuality Metrics. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4812–4829.

Association for Computational Linguistics.

Pang, J., Ye, F., Wong, D. F., Yu, D., Shi, S., Tu, Z., & Wang, L. (2025). Salute the classic: Revisiting challenges of machine translation in the age of large language models. *Transactions of the Association for Computational Linguistics*, 13:73–95.

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). Bleu: a Method for Automatic Evaluation of Machine Translation. In Isabelle, P., Charniak, E., & Lin, D., editors, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Paulus, R., Xiong, C., & Socher, R. (2017). A Deep Reinforced Model for Abstractive Summarization.

Peng, N. V. (2022). Controllable Text Generation for Open-Domain Creativity and Fairness. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization.

Perkins, M., Roe, J., Postma, D., McGaughran, J., & Hickerson, D. (2024). Detection of GPT-4 Generated Text in Higher Education: Combining Academic Judgement and Software to Identify Generative AI Tool Misuse. *Journal of Academic Ethics*, 22(1):89–113.

Popović, M. (2017). chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pp. 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

Prabhumoye, S., Hashimoto, K., Zhou, Y., Black, A. W., & Salakhutdinov, R. (2021). Focused attention improves document-grounded generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4274–4287. Association for Computational Linguistics.

Premack, D. (2004). Is Language the Key to Human Intelligence? *Science*, 303(5656):318–320.

Qader, R., Jneid, K., Portet, F., & Labbé, C. (2018). Generation of company descriptions using concept-to-text and text-to-text deep models: dataset collection and systems evaluation. In *Proceedings of the 11th International Conference on Natural Language Generation*, pp. 254–263, Tilburg University, The Netherlands. Association for Computational Linguistics.

Radev, D. R., Hovy, E., & McKeown, K. (2002). Introduction to the Special Issue on Summarization. *Computational Linguistics*, 28(4):399–408.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language Models are Unsupervised Multitask Learners.

Raiaan, M. A. K., Mukta, M. S. H., Fatema, K., Fahad, N. M., Sakib, S., Mim, M. M. J., Ahmad, J., Ali, M. E., & Azam, S. (2024). A review on large language models: Architectures, applications, taxonomies, open issues and challenges. *IEEE Access*, 12:26839–26874.

Rajpurkar, P., Jia, R., & Liang, P. (2018). Know What You Don't Know: Unanswerable Questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 784–789, Melbourne, Australia. Association for Computational Linguistics.

Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, Austin, Texas. Association for Computational Linguistics.

Ramos, L., Marquez, R., & Rivas, F. (2023). AI's next frontier: The rise of ChatGPT and its implications on society, industry, and scientific research. 44:131–148.

Reiter, E. & Belz, A. (2009). An Investigation into the Validity of Some Metrics for Automatically Evaluating Natural Language Generation Systems. *Computational Linguistics*, 35(4):529–558.

Reiter, E. & Dale, R. (1997). Building applied natural language generation systems. *Natural Language Engineering*, 3(1):57–87.

Reiter, E., Mellish, C., & Levine, J. (1995). Automatic Generation of Technical Documentation. *Applied Artificial Intelligence*, 9(3):259–287.

Rodriguez, J., Hay, T., Gros, D., Shamsi, Z., & Srinivasan, R. (2022). Cross-domain detection of GPT-2-generated technical text. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1213–1233, Seattle, United States. Association for Computational Linguistics.

Rogers, A., Gardner, M., & Augenstein, I. (2023). QA Dataset Explosion: A Taxonomy of NLP Resources for Question Answering and Reading Comprehension. *ACM Computing Surveys*, 55(10):1–45.

Roziere, B., Zhang, J. M., Charton, F., Harman, M., Synnaeve, G., & Lample, G. (2022). Leveraging automated unit tests for unsupervised code translation.

Sakaguchi, K., Post, M., & Van Durme, B. (2014). Efficient Elicitation of Annotations for Human Evaluation of Machine Translation. In Bojar, O., Buck, C., Federmann, C., Haddow, B., Koehn, P., Monz, C., Post, M., & Specia, L., editors, *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pp. 1–11, Baltimore, Maryland, USA. Association for Computational Linguistics.

Schick, T., Udupa, S., & Schütze, H. (2021). Self-diagnosis and self-debiasing: A proposal for reducing corpus-based bias in NLP. *Transactions of the Association for Computational Linguistics*, 9:1408–1424.

Schuster, T., Schuster, R., Shah, D. J., & Barzilay, R. (2020). The limitations of stylometry for detecting machine-generated fake news. *Computational Linguistics*, 46(2):499–510.

Sellam, T., Das, D., & Parikh, A. (2020). BLEURT: Learning robust metrics for text generation. In *Proc. of ACL*, pp. 7881–7892. Association for Computational Linguistics.

Seo, H., Jung, S., Jung, J., Hwang, T., Namgoong, H., & Roh, Y.-H. (2023). Controllable Text Generation Using Semantic Control Grammar. *IEEE Access*, 11:26329–26343.

Shaham, U., Herzig, J., Aharoni, R., Szpektor, I., Tsarfaty, R., & Eyal, M. (2024). Multilingual Instruction Tuning With Just a Pinch of Multilinguality.

Shao, C., Hu, X., Lin, Y., & Xu, F. (2025). Division-of-thoughts: Harnessing hybrid language model synergy for efficient on-device agents. In *Proceedings of the ACM on Web Conference 2025*, WWW '25, p. 1822–1833, New York, NY, USA. Association for Computing Machinery.

Shen, L., Li, S., & Chen, Y. (2022a). KATG: keyword-bias-aware adversarial text generation for text classification. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pp. 11294–11302. AAAI Press.

Shen, L., Liu, L., Jiang, H., & Shi, S. (2022b). On the Evaluation Metrics for Paraphrase Generation. In Goldberg, Y., Kozareva, Z., & Zhang, Y., editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 3178–3190, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Sheng, E., Chang, K.-W., Natarajan, P., & Peng, N. (2019). The woman worked as a babysitter: On biases in language generation. In *Proc. of EMNLP*, pp. 3407–3412, Hong Kong, China. Association for Computational Linguistics.

Shumailov, I., Shumaylov, Z., Zhao, Y., Papernot, N., Anderson, R., & Gal, Y. (2024). Ai models collapse when trained on recursively generated data. *Nature*, 631(8022):755–759.

Singh, A., Fry, A., Perelman, A., Tart, A., Ganesh, A., El-Kishky, A., McLaughlin, A., Low, A., Ostrow, A., Ananthram, A. et al. (2025). Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*.

Singh, C., Inala, J. P., Galley, M., Caruana, R., & Gao, J. (2024). Rethinking Interpretability in the Era of Large Language Models.

Skantze, G. (2021). Turn-taking in Conversational Systems and Human-Robot Interaction: A Review. *Computer Speech & Language*, 67:101178.

Song, C. & Shmatikov, V. (2019). Auditing data provenance in text-generation models. In Teredesai, A., Kumar, V., Li, Y., Rosales, R., Terzi, E., & Karypis, G., editors, *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pp. 196–206. ACM.

Sonntag, D. (2004). Assessing the quality of natural language text data. pp. 259–263. Gesellschaft für Informatik e.V.

Stolfo, A., Jin, Z., Shridhar, K., Schoelkopf, B., & Sachan, M. (2023). A Causal Framework to Quantify the Robustness of Mathematical Reasoning with Language Models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Sun, T., Gaut, A., Tang, S., Huang, Y., ElSherief, M., Zhao, J., Mirza, D., Belding, E., Chang, K.-W., & Wang, W. Y. (2019). Mitigating gender bias in natural language processing: Literature review. In *Proc. of ACL*, pp. 1630–1640, Florence, Italy. Association for Computational Linguistics.

Tan, Y., Min, D., Li, Y., Li, W., Hu, N., Chen, Y., & Qi, G. (2023). Can ChatGPT Replace Traditional KBQA Models? An In-depth Analysis of the Question Answering Performance of the GPT LLM Family.

Taunk, D., Sagare, S., Patil, A., Subramanian, S., Gupta, M., & Varma, V. (2023). XWikiGen: Cross-lingual Summarization for Encyclopedic Text Generation in Low Resource Languages. In *Proceedings of the ACM Web Conference 2023*. ACM.

Tay, Y., Dehghani, M., Bahri, D., & Metzler, D. (2020). Efficient Transformers: A Survey. *arXiv:2009.06732 [cs]*.

Tenney, I., Wexler, J., Bastings, J., Bolukbasi, T., Coenen, A., Gehrmann, S., Jiang, E., Pushkarna, M., Radebaugh, C., Reif, E. et al. (2020). The Language Interpretability Tool: Extensible, Interactive Visualizations and Analysis for NLP Models.

Tian, C., Zhu, X., Xiong, Y., Wang, W., Chen, Z., Wang, W., Chen, Y., Lu, L., Lu, T., Zhou, J. et al. (2024a). MM-Interleaved: Interleaved Image-Text Generative Modeling via Multi-modal Feature Synchronizer.

Tian, Y., Xia, F., & Song, Y. (2024b). Dialogue summarization with mixture of experts based on large language models. In Ku, L.-W., Martins, A., & Srikumar, V., editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7143–7155, Bangkok, Thailand. Association for Computational Linguistics.

Tourille, J., Sow, B., & Popescu, A. (2022). Automatic Detection of Bot-generated Tweets. In *Proceedings of the 1st International Workshop on Multimedia AI against Disinformation*. ACM.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F. et al. (2023). LLaMA: Open and Efficient Foundation Language Models.

Uchendu, A., Ma, Z., Le, T., Zhang, R., & Lee, D. (2021). TURINGBENCH: A benchmark environment for Turing test in the age of neural text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 2001–2016, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Uto, M., Tomikawa, Y., & Suzuki, A. (2023). Difficulty-Controllable Neural Question Generation for Reading Comprehension using Item Response Theory. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pp. 119–129, Toronto, Canada. Association for Computational Linguistics.

Vaibhav, V., Singh, S., Stewart, C., & Neubig, G. (2019). Improving robustness of machine translation with synthetic noise. In *Proc. of NAACL-HLT*, pp. 1916–1920, Minneapolis, Minnesota. Association for Computational Linguistics.

Valenzuela-Escarcega, M., Ha, V. A., & Etzioni, O. (2015). Identifying Meaningful Citations.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Vedantam, R., Zitnick, C. L., & Parikh, D. (2015). CIDEr: Consensus-based Image Description Evaluation.

Venkatraman, S., Tripto, N. I., & Lee, D. (2025). CollabStory: Multi-LLM collaborative story generation and authorship analysis. In Chiruzzo, L., Ritter, A., & Wang, L., editors, *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 3665–3679, Albuquerque, New Mexico. Association for Computational Linguistics.

Vila, M., Bertran, M., Martí, M. A., & Rodríguez, H. (2015). Corpus annotation with paraphrase types: new annotation scheme and inter-annotator agreement measures. *Language Resources and Evaluation*, 49(1):77–105.

Vila, M., Martí, M. A., & Rodríguez, H. (2014). Is This a Paraphrase? What Kind? Paraphrase Boundaries and Typology. *Open Journal of Modern Linguistics*, 04(01):205–218.

Vilar, D., Freitag, M., Cherry, C., Luo, J., Ratnakar, V., & Foster, G. (2023). Prompting PaLM for Translation: Assessing Strategies and Performance.

Wahle, J., Ruas, T., Abdalla, M., Gipp, B., & Mohammad, S. (2023a). We are Who We Cite: Bridges of Influence Between Natural Language Processing and Other Academic Fields. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12896–12913, Singapore. Association for Computational Linguistics.

Wahle, J. P., Gipp, B., & Ruas, T. (2023b). Paraphrase Types for Generation and Detection.

Wahle, J. P., Ruas, T., Foltýnek, T., Meuschke, N., & Gipp, B. (2022a). Identifying Machine-Paraphrased Plagiarism. In *Information for a Better World: Shaping the Global Future: 17th International Conference, iConference 2022, Virtual Event, February 28 – March 4, 2022, Proceedings, Part I*, pp. 393–413, Berlin, Heidelberg. Springer-Verlag.

Wahle, J. P., Ruas, T., Kirstein, F., & Gipp, B. (2022b). How Large Language Models are Transforming Machine-Paraphrase Plagiarism. In Goldberg, Y., Kozareva, Z., & Zhang, Y., editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 952–963, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Wahle, J. P., Ruas, T., Meuschke, N., & Gipp, B. (2021). Are Neural Language Models Good Plagiarists? A Benchmark for Neural Paraphrase Detection. In *2021 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pp. 226–229.

Wahle, J. P., Ruas, T., Mohammad, S. M., Meuschke, N., & Gipp, B. (2023c). AI Usage Cards: Responsibly Reporting AI-generated Content.

Wang, L., Lyu, C., Ji, T., Zhang, Z., Yu, D., Shi, S., & Tu, Z. (2023a). Document-Level Machine Translation with Large Language Models.

Wang, L., Yao, J., Tao, Y., Zhong, L., Liu, W., & Du, Q. (2018). A reinforced topic-aware convolutional sequence-to-sequence model for abstractive text summarization. In Lang, J., editor, *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pp. 4453–4460. ijcai.org.

Wang, Q., Wang, Z., Su, Y., Tong, H., & Song, Y. (2024). Rethinking the bounds of LLM reasoning: Are multi-agent discussions the key? In Ku, L.-W., Martins, A., & Srikumar, V., editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6106–6131, Bangkok, Thailand. Association for Computational Linguistics.

Wang, Z., Hamza, W., & Florian, R. (2017). Bilateral Multi-Perspective Matching for Natural Language Sentences.

Wang, Z., Mao, S., Wu, W., Ge, T., Wei, F., & Ji, H. (2023b). Unleashing Cognitive Synergy in Large Language Models: A Task-Solving Agent through Multi-Persona Self-Collaboration.

Wang, Z., Wang, X., An, B., Yu, D., & Chen, C. (2020). Towards faithful neural table-to-text generation with content-matching constraints. In *Proc. of ACL*, pp. 1072–1086. Association for Computational Linguistics.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E. H., Le, Q. V., & Zhou, D. (2024). Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International*

Conference on Neural Information Processing Systems, NIPS '22, pp. 24824–24837, Red Hook, NY, USA. Curran Associates Inc.

Wiher, G., Meister, C., & Cotterell, R. (2022). On decoding strategies for neural text generators. *Transactions of the Association for Computational Linguistics*, 10:997–1012.

Williams, A., Nangia, N., & Bowman, S. (2018). A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Winograd, T. (1972). The Automatic Generation of Natural Language Texts. *Artificial Intelligence*, 3(3-4):185–231.

Wu, X., Duan, R., & Ni, J. (2024). Unveiling security, privacy, and ethical concerns of ChatGPT. *Journal of Information and Intelligence*, 2(2):102–115.

Xiao, W. & Carenini, G. (2019). Extractive Summarization of Long Documents by Combining Global and Local Context.

Xiao, Y., Wu, L., Guo, J., Li, J., Zhang, M., Qin, T., & Liu, T.-y. (2023). A Survey on Non-Autoregressive Generation for Neural Machine Translation and Beyond.

Xu, K., Wu, L., Wang, Z., Feng, Y., & Sheinin, V. (2018). SQL-to-text generation with graph-to-sequence model. In *Proc. of EMNLP*, pp. 931–936, Brussels, Belgium. Association for Computational Linguistics.

Xu, Z., Liu, B., Wang, B., Sun, C., Wang, X., Wang, Z., & Qi, C. (2017). Neural response generation via GAN with an approximate embedding layer. In *Proc. of EMNLP*, pp. 617–626, Copenhagen, Denmark. Association for Computational Linguistics.

Yang, E., Bai, C., Xiong, D., Zhang, Y., Meng, Y., Xu, J., & Chen, Y. (2022a). Learning Structural Information for Syntax-Controlled Paraphrase Generation. In Carpuat, M., de Marneffe, M.-C., & Meza Ruiz, I. V., editors, *Findings of the Association for Computational Linguistics: NAACL 2022*, pp. 2079–2090, Seattle, United States. Association for Computational Linguistics.

Yang, K. & Klein, D. (2021). FUDGE: Controlled text generation with future discriminators. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3511–3535. Association for Computational Linguistics.

Yang, K., Liu, D., Lei, W., Yang, B., Xue, M., Chen, B., & Xie, J. (2023a). Tailor: A Soft-Prompt-Based Approach to Attribute-Based Controlled Text Generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Yang, K., Tian, Y., Peng, N., & Klein, D. (2022b). Re3: Generating Longer Stories With Recursive Reprompting and Revision.

Yang, X., Li, Y., Zhang, X., Chen, H., & Cheng, W. (2023b). Exploring the limits of chatgpt for query or aspect-based text summarization.

Yao, Y., Duan, J., Xu, K., Cai, Y., Sun, Z., & Zhang, Y. (2024). A survey on large language model (LLM) security and privacy: The Good, The Bad, and The Ugly. *High-Confidence Computing*, 4(2):100211.

Yin, Z., Sun, Q., Chang, C., Guo, Q., Dai, J., Huang, X., & Qiu, X. (2023). Exchange-of-Thought: Enhancing Large Language Model Capabilities through Cross-Model Communication.

Yu, W., Zhu, C., Li, Z., Hu, Z., Wang, Q., Ji, H., & Jiang, M. (2022). A Survey of Knowledge-enhanced Text Generation. *ACM Computing Surveys*, 54(11s):1–38.

Yuan, A., Coenen, A., Reif, E., & Ippolito, D. (2022). Wordcraft: Story Writing With Large Language Models. In *27th International Conference on Intelligent User Interfaces*. ACM.

Yuan, W., Neubig, G., & Liu, P. (2021). BARTScore: Evaluating Generated Text as Text Generation.

Yuan, X., Wang, T., Gulcehre, C., Sordoni, A., Bachman, P., Zhang, S., Subramanian, S., & Trischler, A. (2017). Machine comprehension by text-to-text neural question generation. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pp. 15–25, Vancouver, Canada. Association for Computational Linguistics.

Yue, X., Inan, H., Li, X., Kumar, G., McAnallen, J., Shajari, H., Sun, H., Levitan, D., & Sim, R. (2023). Synthetic Text Generation with Differential Privacy: A Simple and Practical Recipe. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., & Choi, Y. (2019). HellaSwag: Can a machine really finish your sentence? In *Proc. of ACL*, pp. 4791–4800, Florence, Italy. Association for Computational Linguistics.

Zellers, R., Holtzman, A., Rashkin, H., Bisk, Y., Farhadi, A., Roesner, F., & Choi, Y. (2020). Defending Against Neural Fake News.

Zhang, B., Haddow, B., & Birch, A. (2023). Prompting Large Language Model for Machine Translation: A Case Study.

Zhang, H., Cai, J., Xu, J., & Wang, J. (2019). Pretraining-based natural language generation for text summarization. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pp. 789–797, Hong Kong, China. Association for Computational Linguistics.

Zhang, J., Zhao, Y., Saleh, M., & Liu, P. J. (2020a). PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization.

Zhang, J. & Zong, C. (2020). Neural machine translation: Challenges, progress and future. *Science China Technological Sciences*, 63(10):2028–2050.

Zhang, M., Shen, X., Cao, J., Cui, Z., & Jiang, S. (2025). Edgeshard: Efficient llm inference via collaborative edge computing. *IEEE Internet of Things Journal*, 12(10):13119–13131.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2020b). BERTScore: Evaluating Text Generation with BERT.

Zhang, W., Deng, Y., Li, X., Yuan, Y., Bing, L., & Lam, W. (2021). Aspect sentiment quad prediction as paraphrase generation. In *Proc. of EMNLP*, pp. 9209–9219, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhang, Y., Leng, Q., Zhu, M., Ding, R., Wu, Y., Song, J., & Gong, Y. (2024). Enhancing text authenticity: A novel hybrid approach for ai-generated text detection. In *2024 IEEE 4th International Conference on Electronic Technology, Communication and Information (ICETCI)*, pp. 433–438.

Zhang, Y., Sun, S., Galley, M., Chen, Y.-C., Brockett, C., Gao, X., Gao, J., Liu, J., & Dolan, B. (2020c). DIALOGPT : Large-Scale Generative Pre-training for Conversational Response Generation. In Celikyilmaz, A. & Wen, T.-H., editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 270–278. Association for Computational Linguistics.

Zhang, Y., Wang, G., Li, C., Gan, Z., Brockett, C., & Dolan, B. (2020d). POINTER: Constrained progressive text generation via insertion-based generative pre-training. In *Proc. of EMNLP*, pp. 8649–8670. Association for Computational Linguistics.

Zhao, J., Wang, T., Yatskar, M., Ordonez, V., & Chang, K.-W. (2018). Gender Bias in Coreference Resolution: Evaluation and Debiasing Methods.

Zhao, W., Peyrard, M., Liu, F., Gao, Y., Meyer, C. M., & Eger, S. (2019). MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. In *Proc. of EMNLP*, pp. 563–578, Hong Kong, China. Association for Computational Linguistics.

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z. et al. (2023). A Survey of Large Language Models.

Zheng, C., Shi, C., Vafa, K., Feder, A., & Blei, D. (2023a). An Invariant Learning Characterization of Controlled Text Generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E. P. et al. (2023b). Judging llm-as-a-judge with mt-bench and chatbot arena. In *Proceedings of the 37th International Conference*

on Neural Information Processing Systems, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Zheng, R., Ma, M., & Huang, L. (2018). Multi-reference training with pseudo-references for neural translation and text generation. In *Proc. of EMNLP*, pp. 3188–3197, Brussels, Belgium. Association for Computational Linguistics.

Zhong, W., Tang, D., Xu, Z., Wang, R., Duan, N., Zhou, M., Wang, J., & Yin, J. (2020). Neural deepfake detection with factual structure of text. In *Proc. of EMNLP*, pp. 2461–2470. Association for Computational Linguistics.

Zhou, C., Neubig, G., Gu, J., Diab, M., Guzmán, F., Zettlemoyer, L., & Ghazvininejad, M. (2021). Detecting hallucinated content in conditional neural sequence generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 1393–1404. Association for Computational Linguistics.

Zhu, C., Xu, R., Zeng, M., & Huang, X. (2020). A Hierarchical Network for Abstractive Meeting Summarization with Cross-Domain Pretraining. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 194–203. Association for Computational Linguistics.

Zhu, Y., Lu, S., Zheng, L., Guo, J., Zhang, W., Wang, J., & Yu, Y. (2018). Texygen: A Benchmarking Platform for Text Generation Models.

Zhuang, Y., Yu, Y., Wang, K., Sun, H., & Zhang, C. (2023). ToolQA: A Dataset for LLM Question Answering with External Tools.

Received 18 March 2026; accepted 15 July 2026