Innovations in Neural Data-to-text Generation: A Survey

MANDAR SHARMA, Virginia Tech, USA
AJAY KUMAR GOGINENI, Virginia Tech, USA
NAREN RAMAKRISHNAN, Virginia Tech, USA

The neural boom that has sparked natural language processing (NLP) research throughout the last decade has similarly led to significant innovations in data-to-text generation (D2T). This survey offers a consolidated view into the neural D2T paradigm with a structured examination of the approaches, benchmark datasets, and evaluation protocols. This survey draws boundaries separating D2T from the rest of the natural language generation (NLG) landscape, encompassing an up-to-date synthesis of the literature, and highlighting the stages of technological adoption from within and outside the greater NLG umbrella. With this holistic view, we highlight promising avenues for D2T research that not only focus on the design of linguistically capable systems but also systems that exhibit fairness and accountability.

CCS Concepts:
• Computing methodologies → Natural language generation; Learning paradigms; Machine learning algorithms.

Additional Key Words and Phrases: narration, data-to-text, data-to-text generation, natural language generation

1 INTRODUCTION

Textual Representations of Information: A picture is worth a thousand words - isn’t it? And hence graphical representation is by its nature universally superior to text - isn’t it? Why then isn’t the anecdote itself represented graphically? - Petre [246], in his advocacy for textual representation of information, challenges the notion that graphical representations of information are inherently more memorable, comprehensible, and accessible than their textual counterparts. Gershon and Page [104] note that the transformation of information from a textual to visual domain, in certain instances, requires further addition of information rendering textual representations more economical. Similarly, knowing where to look may not be obvious in visual representations of information - as validated through reading comprehension experiments [31, 115, 116] where participants were significantly slower in interpreting visual representations of the nested conditional structures within a program compared to their textual representations. This being said, these studies do not intend to dissuade the use of visual representations but rather establish the importance of textual representation of information. Often, the interplay of these paradigms brings out the best of both [289]. Thus, having established the importance of textual representations of information, we next explore how these notions tie into Data-to-text (D2T) generation.

1.1 Defining Data-to-text Generation and Scope of the Survey

Textual representations of information, for easier assimilation, are often presented as annotations outlining different behaviours of the underlying data stitched together. These stitched annotations, as showcased in Fig. 1, are referred to as narratives. The automated generation of such narratives, although serving several niches (see below), are most prevalent in the public eye through the practice of robo-journalism [66, 176]. Bloomberg News generates a third of its content with Cyborg, their in-house automation system that can dissect tedious financial reports and churn out news articles within seconds [243]. Also prevalent are the use of such systems in business intelligence.
The practice of automating the translation of data to user-consumable narratives through such systems is known as data-to-text generation, as depicted in Fig. 1. Although encompassed by the general umbrella of Natural Language Generation, the nuance that differentiates D2T from the rest of the NLG landscape is that the input to the system has to qualify as a data instance. Reiter and Dale (1997) [270] describe the instance as a non-linguistic representation of information, and although narration of images and videos [70] has garnered interest in the NLG community, the definition of data-to-text employed by this survey follows that established by the seminal works prior [100, 270]: an entity that is not exclusively linguistic - tabular databases, graphs and knowledge bases, time-series, and charts. Using this clause, we limit the scope of our analysis and exclude examination of all other NLG systems that either both ingest and expel linguistic entities for downstream tasks such as machine translation [145, 350] or summarization [188, 228] or ingest non-conventional data such as images [131] and videos [328].

Outside of dataset specific tasks, practical applications of D2T include, but are not limited to:

- Weather forecasts [20, 271]
- Sport summaries [15, 279, 316]
- Healthcare [241, 249]
- Virtual dietitians [9]
- Stock market comments [10, 220]
- Video-game dialogues [149] and Driving feedback [35]

---

1This survey exclusively focuses on academic innovations for data-to-text generation as the technologies underlying commercial frameworks are often proprietary.
2https://www.arria.com/
3https://narrativescience.com/
4https://automatedinsights.com/
With our scope defined, below we outline the rationale for this survey, followed by a structured examination of approaches, benchmark datasets, and evaluation protocols that constitute the D2T landscape with the intent to outline promising avenues for further research.

1.2 Survey Rationale
Following the seminal work by Reiter and Dale [270], the most comprehensive survey on D2T to-date has been that by Gatt and Krahmer [100]. Although several articles have taken a close examination of NLG sub-fields such as dialogue systems [282], poetry generation [234], persuasive text generation [76], social robotics [94], or exclusively focus on issues central to NLG such as faithfulness [181] and hallucination [142], a detailed break-down of the last half-decade of innovations has been missing since the last exhaustive body of work. The need for a close and consolidated examination of developments in neural D2T is more pertinent now than ever. Further, D2T distinguishes itself from other NLG tasks as it blends the generation of narratives with numerical reasoning between data points. Outside of the D2T niche, there are research communities focused on solving these individual problems - NLG [100, 235, 320] and numerical reasoning [295, 317, 334, 349]. Thus, neural D2T is uniquely positioned such that it either has to incorporate innovations from these seemingly disparate niches or jointly innovate on both fronts. We believe this provides added justification for D2T requiring its own comprehensive literature review.

As such, neural D2T borrows heavily from advances in other facets of NLG such as neural machine translation (NMT) [11, 350] and spoken dialogue systems (SDS) [79, 342, 343]. As such, the pertinence of such a survey also spans highlighting the stages of technological adoptions in the D2T paradigm and drawing distinctions between its NMT and SDS neighbors. Further, the adoption of such technologies brings about the adoption of shared pitfalls - inconsistencies in evaluation metrics [268] and meaningful inter-model comparisons [265]. Thus, in addition to an exhaustive examination of neural D2T frameworks, a consolidated resource on approaches to its evaluation is also necessary. Also crucial, is the discussion of benchmark datasets across shared tasks. The above considerations motivate our survey on the neural D2T paradigm intended to serve the following goals:

- Structured examination of innovations in neural D2T in the last half-decade spanning relevant frameworks, datasets, and evaluation measures.
- Outlining the technological adoptions in D2T from within and outside of the greater NLG umbrella with the distinctions and shared pitfalls that lie therein.
- Highlighting promising avenues for further D2T research and exploration that promote fairness and accountability along with linguistic prowess.

2 DATASETS FOR DATA-TO-TEXT GENERATION
The first set of technological adoptions from NLP takes the form of dataset design: parallel corpora that align the data to their respective narratives are crucial for end-to-end learning, analogous to any neural-based approach to text processing. The initial push towards building such datasets began with database-text pairs of weather forecasts [20, 271] and sport summaries [15]. These datasets, and the convention that currently follows, use semi-structured data that deviates from the raw numeric signals initially used for D2T systems [266]. The statistics for prominent datasets among the ones discussed below are detailed in Table 2.

2.1 Meaning Representations
Mooney [217] defines a meaning representation language (MRL) as a formal unambiguous language that allows for automated inference and processing wherein natural language is mapped to its
respective meaning representation (MR) through semantic parsing [101]. Robocup [42], among pioneering MR-to-text datasets, offers data from 1539 pairs of temporally ordered simulated soccer games in the form of MRs (pass, kick, turnover) accompanied with their respective human commentary. In order to mitigate the cost of building large-scale MR datasets, Liang et. al. [183] use grounded language acquisition to construct WeatherGov - a weather forecasting dataset with 29528 MR-text pairs, each consisting of 36 different weather states. Abstract meaning representation (AMR) [13], similarly, is a linguistically grounded semantic formalism representing the meaning of a sentence as a directed graph, as depicted in Fig. 2a. The LDC repository\(^5\) hosts various AMR-based corpora. Following this, using simulated dialogues between their statistical dialogue manager [357] and an agenda-based user simulator [283], Mairesse et. al. [204] offer BAGEL - an MR-text collection of 202 Cambridge-based restaurant descriptions each accompanied with two inform and reject dialogue types. Wen et. al. [343], through crowdsourcing, offer an enriched dataset consisting of 5192 instances of 6 additional dialogue act types such as confirm and informonly (8 total) for hotels and restaurants in San Francisco. Novikova et. al. [232] show that crowdsourcing with the aid of pictorial stimuli yield better phrased references compared to textual MRs. Following this, they released the E2E dataset\(^6\) as a part of the E2E challenge [230]. With 50,602 instances of MR-text pairs of restaurant descriptions, its lexical richness and syntactic complexity provides new challenges for D2T systems. Table 1 showcases comparative snapshots of the aforementioned datasets.

Table 1. Comparative showcase of sample MRs (and their corresponding narratives) from the RoboCup, WeatherGov, BAGEL, SF Hotels and Restaurants, and E2E datasets.

| MR | Text |
|----|------|
| **RoboCup** [42] | badPass(arg1=pink11,...), ballstopped() ballstopped(), kick(arg1=pink11) turnover(arg1=pink11,...) | pink11 makes a bad pass and was picked off by purple3 |
| **WeatherGov** [183] | rainChance(time=26-30,...), temperature(time=17-30,...) windDir(time=17-30,...), windSpeed(time=17-30,...) precipPotential(time=17-30,...), rainChance(time=17-30,...) | Occasional rain after 3am. Low around 43. South wind between 11 and 14 mph. Chance of precipitation is 80%. New rainfall amounts between a quarter and half of an inch possible. |
| **BAGEL** [204] | inform( name(the Fountain) near(the Arts Picture House) area(centre), pricerange(cheap)) | There is an inexpensive restaurant called the Fountain in the centre of town near the Arts Picture House |
| **SF Hotels & Rest.** [343] | inform( name="red door cafe", goodformeal="breakfast", area="cathedral hill", kidsallowed="no") | red door cafe is a good restaurant for breakfast in the area of cathedral hill and does not allow children. |
| **E2E** [232] | name[Loch Fyne], eatType[restaurant], food[French], priceRange[less than £20], familyFriendly[yes] | Loch Fyne is a family-friendly restaurant providing wine and cheese at a low cost. |

2.2 Graph Representations

Graph-to-text translation is not only central to D2T as its application carries over to numerous NLG fields such as question answering [74, 130], summarization [86], and dialogue generation [190, 216]. Further, the D2T frameworks for graph-to-text borrow heavily from the theoretic formulations offered from the literature in the field of graph neural networks (GNNs), as will be discussed in §5.1.5. The domain-specific benchmark datasets, as discussed above (see §2.1) inherently train models to generate stereotypical domain-specific text. By crowdsourcing annotations for DBPedia

---

\(^5\)https://amr.isi.edu/

\(^6\)http://www.macs.hw.ac.uk/InteractionLab/E2E/
graphs spanning 15 domains, Gardent et al. [99] introduce the WebNLG dataset\(^7\). The data instances are encoded as Resource Description Format (RDF) triples of the form (subject, property, object) as depicted in Fig. 2b - (Apollo 12, operator, NASA). With 27,731 multi-domain graph-text pairs, WebNLG offers more semantic and linguistic diversity than previous datasets twice its size [342]. The abstract generation dataset (AGENDA) [168], built with knowledge graphs extracted from articles in the proceedings of AI conferences [6] using SciIE [199], offers 40,000 graph-text pairs of the article abstracts. To further promote generation challenges and cross-domain generalization, Nan et al. [224] merge the E2E and WebNLG dataset with large heterogeneous collections of diverse predicates from Wikipedia tables annotated with tree ontologies to generate the data-record-to-text (DART) corpus. With 82,191 samples, this resulting open-domain corpus is almost quadruple the size of WebNLG.

![Fig. 2. AMR [169] and knowledge graph [274] snapshots, representing variants of graph-based inputs to D2T systems.](image)

2.3 Tabular Representations

Information represented in large tables can be difficult to comprehend at a glance, thus, table-to-text (T2T) aims to generate narratives highlighting crucial elements of a tabular data instance through summarization and logical inference over the table - as showcased in Figure 3. Similar to graph-to-text, the underpinnings of tabular representation learning is also shared with other fields outside of NLG, such as the generation of synthetic network traffic [276, 353]. WikiBio [174], as an initial foray towards a large-scale T2T dataset, offers 700k table-text pairs of Wikipedia info-boxes with the first paragraph of its associated article as the narrative. With a vocabulary of 400k tokens and 700k instances, WikiBio offers a substantially larger benchmark compared to the pioneering WeatherGov and Robocup datasets that have less than 30k data-text pairs. For neural systems, as the length of output sequence increases, the generated summary diverges from the reference. As such, the RotoWire dataset [347] (Fig. 3), consisting of verbose descriptions of NBA game statistics, brings forth new challenges in long-form narrative generation as the average reference length of RotoWire is 337 words compared to 28.7 of WikiBio. Similarly, with the observation that only 60% of the content in RotoWire narratives can be traced back to the data records, Wang [335] introduce RotoWire-FG, a refined version of the original dataset aimed at tackling divergence (see §3.2), where narrative instances not grounded by their respective tables are removed from the dataset. TabFact [50] contains annotated sentences that are either supported or refuted by the tables extracted from Wikipedia. Similar to RotoWire-FG, Chen et al. [47] offer a filtered version of TabFact by retaining only those narratives that can be logically inferred from the table.

\(^7\)https://webnlg-challenge.loria.fr/
Fig. 3. Showcasing the intent of T2T, the statistics of a basketball match between the Atlanta Hawks and the Miami Heat (left) is to be translated into easily consumable narratives (right). Snapshot from the RotoWire dataset [347].

For controlled generation, Parikh et al. [239] propose ToTTo which generates a single sentence description of a table on the basis of a set of highlighted cells where annotators ensure that the target summary only contains the specified subset of information. With over 120k training samples, ToTTo establishes an open-domain challenge for D2T in controlled settings. Similarly, to evaluate narrative generation in open domain settings with sentences that can be logically inferred from mathematical operations over the input table, Chen et. al. [47] modify the reference narratives of the TabFact dataset to construct LogicNLG with 7392 tables. Following this, with tables and their corresponding descriptions extracted from scientific articles, Moosavi et. al. [218] introduce SciGen, where the narratives include arithmetic reasoning over the tabular numeric entries. Building upon the long form generation premise of RotoWire, Chen et. al. [45] construct WikiTableT, a multi-domain table-text dataset with 1.5 million instances pairing Wikipedia descriptions to their corresponding info-boxes along with additional hyperlinks, named-entities, and article metadata. The majority of these datasets are available in a unified framework through TabGenie8.

Table 2. Highlights from prominent D2T datasets: format, number of samples (size), the number of linguistic tokens across the dataset (tokens), and availability of non-anglo-centric variants.

| Benchmark      | Format | Size  | Tokens |
|----------------|--------|-------|--------|
| E2E            | MR     | 50,602| 65,710 |
| LDC2017T10     | AMR    | 39,260| -      |
| WebNLG (en, ru)| RDF    | 27,731| 8,886  |
| DART           | RDF    | 82,191| 33,200 |
| WikiBio        | Record | 728,321| 400,000|
| RotoWire       | Record | 4,853 | 11,300 |
| TabFact        | Record | 16,573| -      |
| ToTTo          | Record | 120,000| 136,777|
| LogicNLG       | Record | 37,000| 52,700 |
| WikiTableT     | Record | 1.5M  | 169M   |

https://pypi.org/project/tabgenie/
2.4 Data Collection & Enrichment

The majority of the prominent datasets discussed in §2.1 - §2.3 are either collected by merging aligned data-narrative pairs that occur naturally in the “wild” [174, 347] or are collected through dedicated crowd-sourcing approaches [99, 230]. However, there are notable works that employ a hybrid approach to data collection. CACAPO [324], an MR-style multi-domain dataset, follows a collection process inspired by Oraby et al. [236] wherein the naturally-occurring narratives are first scraped from the internet and are later manually annotated to generate attribute-value pairs. Similarly, Chart-to-Text [153] follows a similar mechanism of data collection wherein candidate narratives for each chart are first automatically generated via a heuristic-based approach and then are rated by crowd-sourced workers. In similar lines, the ToTTo dataset [239] discussed in §2.3 uses crowd-sourced annotators as data “cleaners” - iteratively improving upon the automatically scraped narratives, rather than annotating them from scratch - thus greatly reducing the cost of data acquisition. In addition to innovations in data collection, efforts from the D2T community has also focused on the enrichment of existing datasets. As such, Ferreira et al. [90] augment the WebNLG dataset with intermediate representation for discourse ordering and referring expression generation. By manually delexicalizing (see §4.1) the narratives, Ferreira et al. were able to automatically extract a collection of referring expressions by tokenizing the original and delexicalized texts and finding the non-overlapping tokens between them. Similarly, the authors also extracted the order of the arguments in the text by referring to the order of the general tags in the delexicalized texts. This work has also been extended to enrich the E2E dataset [92].

3 DATA-TO-TEXT GENERATION FUNDAMENTALS AND NOTATIONS

3.1 What to Say and How to Say It

The data instance, typically, contains more information than what we would intend for the resulting narrative to convey - verbose narratives that detail every attribute of the data instance contradicts the premise of consolidation. Thus, to figure out what to say, a subset of the original information content is filtered out based on the target audience through the process of content selection. Starting from data-driven approaches such as clustering [75] and the use of hidden Markov models (HMMs) [16], the attention of the research community has recently shifted to learning alignments between the data instance and its narrative [183]. Bisazza and Marcello [30] note that pre-reordering the source words to better resemble the target narrative yields significant improvements in NMT. Prior to neural explorations, learning this alignment has been explored with log-linear models [8] and tree representations [170, 171]. With what to say determined, the next step lies in figuring out how to say it, that is, the construction of words, phrases, and paragraphs - this realization of the narrative structure is known as surface realization. While traditionally, the processes of content selection and surface realization [148, 270] act as discrete parts of the generation pipeline, the neural sequence-to-sequence (seq2seq) paradigm jointly learns these aspects. For a peripheral view of the articles discussed in this section, Table 4 highlights prominent papers categorized based on their D2T tasks and the benchmark datasets used. Similarly, Fig. 5 outlines the organization of the remainder of this survey.

3.2 Hallucinations and Omissions

Apart from the importance of coherence and linguistic diversity in surface realization, data fidelity is a crucial aspect of D2T systems - the narrative should neither hallucinate contents absent from the data instance nor omit contents present in the data instance. Often, the divergence present in benchmark training datasets, wherein the narrative may contain data absent from the source or not cover the entirety of the data instance, is the culprit behind hallucination tendencies in
the model [280]. Often times, the need for both linguistic diversity and data fidelity turns into a balancing act between conflicting optimization objectives leading to novel challenges [128]. While almost all of the D2T approaches discussed below engage in balancing coherence and diversity with data-fidelity (besides §5.1.4 Stylistic Encoding), overarchingly, the approach to balancing these conflicting objectives can be thought to take place in two forms:

- **Architectural Interventions**: The sections §5.1.1 Entity Encoders, §5.1.2 Hierarchical Encoders, §5.1.3 Plan Encoders & Autoencoders, §5.1.5 Graph Encoders, §5.1.6 Reconstruction & Hierarchical Decoders, and §5.1.10 Supplemental Frameworks suggest modifications/augmentations to the seq2seq architecture such that it fosters data-fidelity tendencies.

- **Loss-function Interventions**: An alternative avenue to achieving a balance between conflicting optimization objectives is to directly model the objective functions to perform multi-task learning: as such sections §5.1.7 Regularization Techniques and §5.1.8 Reinforcement Learning suggest modifications/augmentations to the seq2seq loss functions.

### 3.3 Establishing Notation and Revisiting Seq2Seq

For the consistency and readability of this survey, the notation outlining the basic encoder-decoder seq2seq paradigm [11, 54, 314, 326] in D2T (Fig. 4), as defined below and respectively compiled in Table 3, will remain valid throughout unless stated otherwise. However, the namespace for additional variable definitions in the individual sections will be limited to their mentions. Let $S = \{x_j, y_j\}_{j=1}^N$ be a dataset of $N$ data instances $x$ accompanied with its natural language narrative $y$. Based on the construction of $S$, $x$ can be a set of $K$ data records $x = \{r_j\}_{j=1}^K$ with each entry $r$ comprised of its respective entity $r.e$ and value $r.m$ attributes or $x$ can be an instance of a directed graph $x = (V, E)$ with vertices $v \in V$ and edges $(u, v) \in E$. In the RotoWire instance (Figure 3), for $r_j = \text{Heat}$, $r_j.e = \text{WIN}$ attribute would have value $r_j.m = 11$. Given pairs $(x, y)$, the seq2seq model $f_\theta$ is trained end-to-end to maximize the conditional probability of generation $P(y|x) = \prod_{t=1}^{T} P(y_t|y_{<t}, x)$. The parameterization of $f_\theta$ is usually carried out through RNNs such as LSTMs [29, 133] and GRUs [54], or transformer$^9$ architectures [326]. For attention-based RNN architectures with hidden states $h_t$ and $s_t$ for the encoder and decoder respectively, the context vector $c_t = \sum_i \alpha_{t,i} h_i$ weighs the encoder hidden states with attention weights $\alpha_{t,i}$. While Bahdanau et. al. [11] use a multi-layer perceptron (MLP) to model $\alpha_{t,i}$, several alterations to modeling the attention weights have been proposed [201, 331].

For handling of out-of-vocabulary (OOV) tokens, Gu et. al. [119] attempt to model the rote memorization process of human learning where a language model conditioned on binary variable $z_t \in \{0, 1\}$ can either generate $p_{gen}$ the next token or copy it from the source $p_{copy}$ based on their respective probabilities. While Gu et. al. [119] and Yang et. al. [355] parameterize the joint distribution over $y_t$ and $z_t$ directly (1), Gulçehre et. al. [121] decompose the joint probability (2), using an MLP to model $p(z_t|y_{<t}, x)$.

\[
\begin{align*}
P(y_t, z_t|y_{<t}, x) &
\begin{cases}
p_{gen}(y_t, y_{<t}, x) & z_t = 0 \\
p_{copy}(y_t, y_{<t}, x) & z_t = 1, y_t \in x \\
0 & z_t = 1, y_t \notin x
\end{cases} \tag{1}
\end{align*}
\]

\[
\begin{align*}
\begin{cases}
p_{gen}(y_t|z_t, y_{<t}, x) & p(z_t|y_{<t}, x) z_t = 0 \\
p_{copy}(y_t|z_t, y_{<t}, x) & p(z_t|y_{<t}, x) z_t = 1
\end{cases} \tag{2}
\end{align*}
\]

$^9$Though the base transformer architecture is oblivious to input structures, we assume positionally encoded transformers to fall into the seq2seq paradigm.
Fig. 4. Attention-based seq2seq framework: The encoder consumes the sequential input translating it to a weighed hidden representation to be then consumed and decoded into linguistic tokens by the decoder.

Fig. 5. Data-to-text Generation Taxonomy Corresponding to Sections in the Survey Design

Similar to the greater NLG paradigm, different strategies for modeling the conditional probability of generation $P(y|x)$, the attention mechanisms $\{a_t,i, c_t\}$, and the copy mechanisms $\{p_{gen}, p_{copy}\}$, as discussed below, often form the basis for D2T innovations. In addition to this, variations in training strategies such as teacher-forcing [346], reinforcement learning [315], and autoencoder-based reconstruction [41] open up further avenues for D2T innovation.

4 INNOVATIONS IN DATA PREPROCESSING

Contrary to the other facets of NLG, such as chatbots, for which large-scale data can be harvested [1, 198], D2T datasets are often smaller in scale and task-specific. Ferreira et. al. [88] note that phrase-based translation models [166] can outperform neural models in such data sparsity. As such, delexicalization, noise reduction, linearization, and data augmentation are preprocessing techniques often employed to tackle said sparsity of training data.

4.1 Delexicalization & Noise Reduction

Delexicalization, often referred to as anonymization, is a common practice in D2T [79, 204] wherein the slot-value pairs for the entities and their attributes in training utterances are replaced with a
Table 3. Notation descriptions

| Notation | Description |
|----------|-------------|
| $S$      | Dataset     |
| $r = (r.e, r.m)$ | Data record $r$ with its entity $r.e$ and value $r.m$ attributes |
| $G = (V, E)$ | Graph instance $G$ with vertices $V$ and edges $E$ |
| $f_0 \in \{f_1, ..., f_n\}$ | Model $f_0$ that may belong to an ensemble $\{f_1, ..., f_n\}$ |
| $P(y|x)$ | Conditional probability of sequence $y$ given $x$ |

placeholder token such that weights between similar utterances can be shared [227] - as illustrated in Figure 6a. These placeholder tokens are later replaced with tokens copied from the input data instance [174]. In comparison to copy-based methods for handling rare entities, delexicalization has shown to yield better results in constrained datasets [300].

From the notion that delexicalization of the data instance may cause the loss of vital information that can aid seq2seq models in sentence planning, where some data instance slots may even be deemed nondelexicalizable [343], Nayak et al. [227] explore different nondelexicalized input representations (mention representations) along with grouping representations as a form of sentence planning (plan representations). The authors note improvements over delexicalized seq2seq baselines when input mentions are concatenated with each slot-value pair representing a unique embedding. The efficacy of such concatenation is also corroborated by Freitag and Roy [95]. Further, the addition of positional tokens representing intended sentence position to the input sequence offers further improvements. Addressing this, in addition to delexicalizing categorical slots, Juraska et al. [150] employ hand-crafted tokens for values that require different treatment in their verbalization: for the slot `food`, the value `Italian` is replaced by `slot_vow_cuisine_food` indicating that the respective utterance should start with a vowel and the value represents a cuisine - an Italian restaurant. Perez-Beltrachini and Lapata [244] delexicalize numerical expressions, such as dates, using tokens created with the attribute name and position of the delexicalised token. Colin and Gardent [59] note performance improvements with an extensive anonymization scheme wherein all lemmatized content words (expect adverbs) are delexicalized as compared to restricting delexicalization to named entities.

The presence of narratives that fail to convey vital attributes of data instances leads to semantic noise in the dataset [84, 136]. Dušek et al. [78] employ slot matching [263] to clean the E2E corpus for semantic correctness and explore the impact of semantic noise on neural model performance.
Similarly, Obeid and Hoque [233] substitute data mentions in the narrative, identified through Named Entity Recognition (NER), with a predefined set of tokens which are later replaced through a look-up operation. Liu et al. [194] focus on generating faithful narratives and truncate the reference narratives by retaining only the first few sentences since the latter are prone to have been inferred from the table.

4.2 Linearization
Frameworks for MR (and graph) narration that shy from dedicated graph encoders rely on effective linearization techniques - the representation of graphs as linear sequences, as illustrated in Figure 6b. While Ferreira et al. [88] note improvements in neural models with the adoption of a 2-step classifier [177] that maps AMRs to the target text, Konstas et al. [169] showcase agnosticism to linearization orders by grouping and anonymizing graph entities for delexicalization with the Stanford NER [93]. The reduction in graph complexity and subsequent mitigation of the challenge brought forth by data sparsity lends any depth-first traversal of the graph as an effective linearization approach. Moryossef et al. [219] append text plans modeled as ordered trees [308] to the WebNLG training set and use an off-the-shelf NMT system [121] for plan-to-text generation. However, the authors note that the restriction of requiring single entity mentions in a sentence establishes their approach as dataset dependent.

For pretrained language models such as GPT-2 [257] and T5 [258], Zhang et al. [361] and Gong et al. [109] represent tables as a linear sequence of attribute-value pairs and use a special token as the separator between the table data and the reference text. It should be noted that T5, while performing the best on automated metrics, fails to generate good summaries when numerical calculations are involved. Chen et al. [47] traverse the table horizontally, each row at a time, where each element is represented by its corresponding field and cell value separated by the keyword is. For scientific tables, Suadaa et al. [312] view a table \( T_D \) as a set of cells with their corresponding row and column headers \( h = [rh : ch] \) with \( th \) for overlapping tokens, numerical value \( val \), and metric-type \( m \). The cells are marked with target flag \( tgt \) which is set to 1 for targeted cells and 0 otherwise respective to the content plan. The linearization of the resulting tables is done with templates that consist of concatenation \( T_D = [h : th : val : m : tgt] \), filtration based on \( tgt \), pre-computed mathematical operations, and their respective combinations.

4.3 Data Augmentation
Often, appending contextual examples from outer sources to the training set, or permuting the training samples themselves to append variation, helps mitigate data sparsity. This is known as data augmentation. Nayak et al. [227] propose the creation of pseudo-samples by permuting the slot orderings of the MRs while keeping the utterances intact. Juraska et al. [150], however, take an utterance-oriented approach where pseudo-samples are built by breaking training MRs into single-sentence utterances. For the shared surface realization task [213], Elder and Hokamp [83] augment the training set with sentences from the WikiText corpus [212] parsed using UDPipe [309]. Following this premise, Kedzie and Mckeown [159] curate a collection of utterances from novel MRs using a vanilla seq2seq model with noise injection sampling [53]. The validity of the MRs associated with these utterances are computed through a CNN-based parser [162] and the valid entries are augmented to the training set. However, it is worth noting, that performance gains from augmenting the training set with out-of-domain (OOD) instances, tend to saturate after a certain point [95]. Also, practitioners of data augmentation should note that caution is advised when augmenting with synthetic data, as the inclusion of such data may reinforce the mistakes of the model [126].
Chen et al. [46] append knowledge-graphs representing external context to the table-text pairs and quantify its efficacy through their metric KBGain - the ratio of tokens unique to the external context to the total number of tokens in the narrative. Similarly, Ma et al. [202] augment the limited training data for table-text pairs by assigning part-of-speech (POS) tags for each word in the reference and further increase the robustness of their model with adversarial examples created by randomly adding and removing words from the input. In contrast, Chen et al. [47] create adversarial examples by randomly swapping entities in the narrative with ones that appear in the table. Following this, Liu et al. [194] use an augmented plan consisting of table records and entities recognized from the reference narrative which eliminates the inclusion of information not present in the table. For few-shot learning, Liu et al. [189] observed that the performance of a GPT-3 model [37] improved upon providing in-context examples computed based on their k-nearest neighbor (k = 2) embeddings.

5 INNOVATIONS IN THE SEQ2SEQ FRAMEWORK

Seq2Seq models (see §3.3), serve as the basis for neural NLG [54, 314, 326]. As such, to compare the efficacy of neural architectures for long-form D2T, Wiseman et al. [347] compare the performance of various seq2seq models to their templated counterparts on the RotoWire dataset. Based on their observations, the conditional copy model [121] performs the best on both word-overlap and extractive metrics (see §6.2.1) compared to the standard attention-based seq2seq model [11] and its joint copy variant [119]. Similarly, in an evaluation of 62 seq2seq, data-driven, and templated systems for the E2E shared task, Dušek et al. [81] note that seq2seq systems dominate in terms of both automated word-based metrics and naturalness in human judgement. Wiseman et al. [347], however, note that the traditional templated generation models outperform seq2seq models on extractive metrics although they score poorly on word-overlap metrics. Thus, the adaptation of seq2seq models to D2T for richer narratives with less omissions and hallucinations still remains an active focus of the research community.

It is worth noting that all seq2seq models discussed below operate at the word level. Models operating at the character level [3, 112, 277] have shown reasonable efficacy with the added computational savings from forgoing the preprocessing steps of delexicalization and tokenization. However, the attention garnered by them from the research community is slim. From their comparative analysis, Jagfeld et al. [140] note that as character-based models perform better on the E2E dataset while word-based models perform better on the more linguistically challenging WebNLG dataset, it is hard to draw conclusions on the framework most suited for generic D2T. In the sections that follow, we detail notable innovations over the last half-decade in seq2seq modeling, branched on the basis of their training strategies - supervised and unsupervised learning.

5.1 Supervised Learning

5.1.1 Entity Encoders. Centering theory [118], as well as many other noted linguistic frameworks [40, 57, 125, 156, 172, 251], highlight the critical importance of entity mentions to the coherence of the generated narrative. The ordering of these entities (r.e in §3.3) is crucial for such narratives to be considered as entity coherent [154]. Unlike typical language models which are conditioned solely based on previously generated tokens $c_t$, Lebret et al. [174] provide additional context $\{z_{c_t}, g_f, g_w\}$ to the generation where $z_{c_t}$ represents table entity $c_t$ as a triplet of its corresponding field name, start, and end positions, and $\{g_f, g_w\}$ are one-hot encoded vectors where each element indicates the presence of table entities from the fixed field and word vocabularies - illustrated in Fig. 7. Similarly, Bao et al. [14] encode the table cell $c$ and attributes $a$ as the concatenation $[e^c_i : e^a_i]$ where the decoder uses this vector to compute the attention weights. Liu et al. [193]
| Dataset | Publication Highlights | Framework & Human Evaluation |
|---------|-----------------------|------------------------------|
| MR-to-Text |
| Robocup & WeatherGov | Coarse-to-fine aligner & penalty based on learned priors | LSTM → LSTM + regularization → N |
| Recipe & SF H&R | Neural agenda-checklist modeling | GRU → GRU + agenda encoders → Y |
| BAGEL | Reranking beam outputs w/ RNN-based reranker | LSTM → LSTM + reranker → N |
| Restaurant Ratings | Nondelexicalized inputs w/ data augmentation | LSTM → LSTM Y |
| WikiData | Complementary text-to-data translation | GRU → GRU Y |

| E2E |
|----------------|-----------------|
| Comparative evaluation of 62 systems | Seq2Seq + Data-driven + Templated Y |
| Two-level hierarchical encoder | CAEncoder → GRU N |
| Ensemble w/ heuristic reranking | Ensemble w/ LSTM + CNN N |
| Hierarchical decoding with POS tags | GRU → GRU N |
| Unsupervised DTG w/ DAEs | LSTM + LSTM + DAEs Y |
| Comparative evaluations w/ ensembling & penalties | Ensemble w/ LSTM + T N |
| Syntactic controls w/ SC-LSTM | SC-LSTM Y |
| Computational pragmatics based DTG | GRU → GRU N |
| Extensive anonymization | LSTM → LSTM Y |
| Semantic correctness in neural DTG | LSTM → LSTM Y |
| Self-training w/ noise injection sampling | GRU → GRU Y |
| Char-level GRU w/ input reconstruction | GRU → GRU N |
| CRFs w/ Gumbel categorical sampling | CRF N |

| AGENDA |
|----------------|-----------------|
| Graph-centric Transformer & AGENDA dataset | T → LSTM + LSTM encoding Y |
| Phrase vs Neural MR-text w/ preprocessing analysis | LSTM → LSTM + Phrase-based N |
| Unlabeled pre-training & linearization agnosticism | LSTM → LSTM N |
| Dual encoding for hybrid traversal | CNN → LSTM Y |
| Graph reconstruction w/ node & edge projection | T → T + reconstruction loss Y |
| Fine-tuning OPT-2 on AMR-text joint distribution | OPT-2 Y |
| Graph encoding with GCNs | GCN → LSTM N |
| LSTM based triple encoder | LSTM → LSTM Y |
| Discrete neural pipelines & comparisons to end-to-end | GRU → GRU + T Y |
| Sentence planning with ordered trees | LSTM → LSTM Y |
| Complementary graph contextualization | GAT → T Y |
| Detachable multi-view reconstruction | T Y |
| Dual encoder for structure and planning | GCN → LSTM Y |
| Task-adaptive pretraining for PLMs | BART + T5 Y |
| Knowledge enhanced language models & KeLM dataset | T5 Y |
| Graph-text joint representations & pretraining strategies | BART + T5 Y |

| WikiBio |
|----------------|-----------------|
| Tabular positional embeddings & WikiBio dataset | LSTM → LSTM + Kneser-Ney N |
| Encoding tabular attributes & WikiTableText dataset | GRU → GRU N |
| Field information through modified LSTM gating | LSTM → LSTM N |
| Link-based and content-based attention | LSTM → LSTM N |
| Multi-instance learning w/ alignment-based rewards | LSTM → LSTM Y |
| Key fact identification and data augment for few shot | LSTM + T Y |
| Hierarchical encoding w/ supervised auxiliary learning | LSTM → LSTM Y |
| Forced attention for omission control | LSTM → LSTM Y |
| External contextual information w/ knowledge graphs | GRU → GRU Y |
| Confidence priors for hallucination control | BERT + Pointer Networks Y |
| Soft copy switching policy for few-shot learning | OPT-2 Y |
| Variational autorecoders for template induction | VAE modified to VTM Y |
| Autoregressive modeling with iterative text-editing | Pointer networks + text-editing Y |
| Reinforcement learning with adversarial networks | GAN N |
| Linearly combined multi-reward policy | Pointer networks N |
| Source-target disagreement auxiliary loss | T N |
| BERT-based IR system for contextual examples | T5 + BERT Y |
| Classification-based metrics & Rotowire dataset | LSTM → LSTM + Templated N |
| Numeric operations and operation-result encoding | GRU → GRU + operation encoders Y |
| Dynamic hierarchical entity-modeling & MLB dataset | LSTM → LSTM Y |
| Content selection & planning w/ gating and IE | LSTM → LSTM Y |
| Contextualized numeric representations | LSTM → LSTM Y |
| Dynamic salient record tracking w/ styled generation | GRU → GRU N |
| Two-tier hierarchical input encoding | T → LSTM Y |
| Auxiliary supervision w/ reasoning over entity graphs | LSTM + GAT Y |
| Neural agenda-checklist modeling | LSTM → LSTM Y |
| Interweaved plan and generation w/ variational models | LSTM → LSTM Y |
### Record-to-text (Continued)

| Dataset                        | Publication Highlights                                                                 | Framework & Human Evaluation |
|--------------------------------|---------------------------------------------------------------------------------------|------------------------------|
| TabFact                        | [47] Coarse-to-fine two-stage generation                                               | LSTM + T + GPT-2 + BERT      |
| WikiPerson                     | [339] Disagreement loss w/ optimal-transport matching loss                             | T                            |
| Humans, Books & Songs          | [109] Attribute prediction-based reconstruction loss                                    | GPT-2                        |
| ToTTo                          | [189] Contextual examples through k nearest neighbors                                  | GPT-3                        |
| LogicNLG                       | [312] Targeted table cell representation                                              | GPT-2                        |
| NumericNLG                     | [49] Semantic confounders w/ Pearl’s do-calculus                                       | DCVED + GPT                  |
|                               | [187] Table-to-logic pretraining for logic text generation                             | T5 + BART                    |
|                               | [225] Faithful generation with unlikelihood & replacement detection                    | T5                           |
|                               | [44] Table serialization and structural encoding                                       | T → GPT-2                    |
|                               | [7] T5 infused with tabular embeddings                                                | T5                           |
| **Cross-domain**               |                                                                                        |                              |
| E2E                            | [140] Char-based vs word-based seq2seq                                                | GRU → GRU                    |
| WebNLG                         | [138] Template induction w/ neural HSMM decoder                                       | HSMM                         |
| DART                           | [98] Training w/ partially aligned dataset                                            | T → T + supportiveness       |
| Wikilibio                      | [157] Iteratively editing templated text                                              | GPT-2 + LaserTagger          |
| RotoWire                       | [127] RoBERTa-based semantic fidelity classifier                                       | GPT-2 + RoBERTa              |
| WITA                           | [48] Knowledge-grounded pre-training & KGTEXT dataset                                  | T → T + GAT                  |
|                               | [351] Disambiguation and stitching with PLMs                                          | GPT3 + T5                    |
|                               | [7] Unified learning of D2T and T2D                                                   | T5 + VAE                     |
|                               | [144] Search and learn in a few-shot setting                                          | T5 + Search & Learn          |
| **Timeseries-to-text**         |                                                                                        |                              |
| WebNLG & DART                  | [294] Open-domain transfer learning for time-series narration                          | BART + T5 + Timeseries analysis |
|                               |                                                                                        |                              |
| **Chart-to-text**              |                                                                                        |                              |
| Chart2Text                     | [233] Preprocessing w/ variable substitution                                          | T → T                        |
|                               |                                                                                        |                              |
|                               | [153] Neural baselines for Chart-to-text dataset                                       | LSTM + T + BART + T5         |

modify the LSTM unit with a field gate to update the cell memory indicating the amount of entity field information to be retained in the cell memory. Following [174], Ma et al. [202] use a Bi-LSTM to encode the concatenation of word, attribute and position embeddings. However, to indicate whether an entity is a key fact, a multi-layer perceptron (MLP) classifier is used on said representation for binary classification. Inspired from Liu and Lapata [195], Gong et al. [108] construct a historical timeline by sorting each table record with respect to its date field. Three encoders encode a table entity separately in row $r_{i,j}^r$, column $r_{i,j}^c$, and time $r_{i,j}^t$ dimensions. The concatenation of these representations are fed to a MLP to obtain a general representation $r_{i,j}^{gen}$ over which specialized attention weights are computed to obtain the final record representation as $\hat{r}_{i,j} = \alpha_r r_{i,j}^r + \alpha_c r_{i,j}^c + \alpha_t r_{i,j}^t$. Exploiting the attributes of the E2E dataset - the set number of unique MR attributes and the limited diversity in lexical instantiations of their values, Puzikov and Gurevych [256] employ a simple approach wherein the recurrent encoder is replaced with one dense layer that takes in MR representations through embedding lookup. Similarly, to keep track of entity mentions in the SF dataset for long-form text generation, Kiddon et al. [160] introduce a checklist vector $a_t$ that aids two additional encoders to track used (mentioned in the resulting narrative) and new (not mentioned as of time step t) items on the defined agenda. The output hidden state is modeled as a linear interpolation between the three encoder states - $c_{t}^{gru}$ of the base GRU and $\{c_{t}^{new}, c_{t}^{used}\}$ from the agenda models, weighted by a probabilistic classifier. Extending this concept of entity encoding to transformer-based architectures, Chen et al. [44] adapt the multi-headed attention layer architecture [326] to encode serialized table attributes that is then fed to a GPT-based decoder. Similarly, as a unified text-to-text alternative approach to [44], Andrejczuk et al. [7] include the row and column embeddings $\hat{r}_{i,j}$ of the input table on top of the token embeddings for table-structure learning in a T5 model.

Certain D2T tasks, such as sport commentaries [15, 279, 316], require reasoning over numeric entities present in the input data instance. Although numeracy in language modeling is a prominent niche of its own [295, 317, 334], notable D2T-specific approaches include that of Nie et al. [229] - precomputing the results of numeric operations $op_e \in \{\text{minus}, \text{argmax}\}$ on the RotoWire dataset,
the authors propose the combination of dedicated operation and operation-result encoders, the latter utilizing a quantization layer for mapping lexical choices to data values, in addition to a record encoder. In similar fashion to [355], the concatenated embeddings \( \{r.\text{id}_x, r.e, r.m\} \) fed to a bi-directional GRU generate record representations while the concatenated embeddings of \( o.p_i \) attributes fed to a non-linear layer yields operation representations. To address the difficulty in establishing lexical choices on sparse numeric values [271, 303], the authors add quantization to the operation-results encoder that maps results of scalar operations \( e \) (minus) to \( l \in \mathcal{L} \) possible bins through a weighed representation \( h_i = \sum \mu_{i,l} e \) using softmax scores of each individual result \( \mu_{i,l} \). Following this body of work, to contextualize numeric representations and thus understand their logical relationships, Gong et al. [107] feed raw numeric embeddings for all numericals corresponding to the same table attributes to a transformer-based encoder to obtain their contextualized representations. Through a ranking scheme based on a fully connected layer, these contextualized representations are further trained to favor larger numbers.

### Hierarchical Encoders

The intuition behind the use of hierarchical encoders, in the context of D2T, is to model input representations at different granularities, either through dedicated modules [191, 261, 360] or attention schemes [193, 253, 355]. As such, Zhang et al. [360] leverage their CAEncoder [359] to incorporate precomputed future representations \( h_i+1 \) into current representation \( h_i \) through a two-level hierarchy. Similarly, Rebuffel et al. [261] propose a two-tier encoder to preserve the data structure hierarchy - the first tier encodes each entity \( e_i \) based on its associated record embeddings \( r_{i,j} \), while the second tier encodes the data structure based on its entity representation \( h_i \) obtained through the individual embeddings \( r_{i,j} \). On the other hand, Liu et al. [191], as illustrated in Fig. 8b, propose a word-level \( h_{r.e}^{w} \) and an attribute-level \( H_{r.e} \) two-encoder setup to capture the attribute-value hierarchical structure in tables. The attribute-level encoder takes in the last hidden state \( h_{r.e}^{w}_{\text{last}} \) for attribute \( r.e \) from the word level LSTM as its input. Using these hierarchical representations, fine-grained attention \( \beta_{r.e}^{m} \propto g(h_{r.e}^{w}, s_t) \) and coarse-grained attention \( \gamma^{r.e} \propto g(H_{r.e}, s_t) \) are used for decoding where \( g \) represents a softmax function. Similarly, based on hierarchical attention [355], Liu et al. [193] employ an attention scheme that attends to both word level and field level tokens. Following this, Puduppully et al. [253] propose language modeling conditioned on both the data instance and a dynamically updated entity representation. At each time-step \( t \), a gate \( y_t \) is used to decide whether an update is necessary for the entity memory representation \( u_k \) and a parameter \( \delta_{t,k} \) decides the impact of said update (3).

\[
y_t = \sigma(W_1s_t + b_1) \quad \& \quad \delta_{t,k} = y_t \odot \sigma(W_2s_t + W_3u_{t-1,k} + b_3)
\]

### Plan Encoders & Autoencoders

Traditionally, the what to say aspect of D2T (see §3.1) used to be its own module in a set of pipelines [100, 270], thus offering flexibility in planning the narrative structure. However, the end-to-end learning paradigm often models content selection and surface realization as a shared task [174, 210, 347]. Although convenient, without explicitly...
modeling the planning of the narrative (Fig. 8a), language models struggle to keep coherence in long-form generation tasks. As such, Puduppully et al. [252] model the generation probability \( P(y|r) \) as the joint probability of narrative \( y \) and content plan \( z \) given a record \( r \) such that \( P(y|r) = \sum_z P(z|r)P(y|r,z) \). Similar to their prior work [253], a content selection gate operates over the record representation \( r_i \) giving an information controlled representation \( r_i^c \). The elements of \( z \) are extracted using an information extraction system [347] and correspond to entities in \( y \) while pointer networks [331] are used to align elements in \( z \) to \( r \) during training. Iso et al. [139], on the other hand, avoid precomputing content plans \( z \) by dynamically choosing data records during decoding - an additional memory state \( h_{ent} \) remembers mentioned entities and updates the language model state \( h_{lm} \) accordingly. The authors propose two representations for an entity - static embedding \( e_r \) based on row \( r \) and aggregated embedding \( \bar{e} \) based on all rows where the entity appears. In the context of the RotoWire dataset, the aggregate embedding \( \bar{e} \) is supposed to represent how entity \( e \) played in the game. For \( h_t = \{h_t^{lm}, h_t^{ent}\} \), \( P(z_t = 1|h_{t-1}) \) (4) models the transition probability, and based on whether \( e \) belongs to the set of entities \( e_s \) that have already appeared at time step \( t \), \( P(e_t = e|h_{t-1}) \) (5) computes the next probable entity \( e \) to mention. The authors note that such discrete tracking dramatically suppresses the generation of redundant relations in the narrative.

\[
\begin{align*}
P(z_t = 1|h_{t-1}) &= \sigma(W_t(h_{t-1}^{lm} \oplus h_{t-1}^{ent})) \quad (4) \\
P(e_t = e|h_{t-1}) &\propto \begin{cases} 
 e^{(W_t h_{t-1}^{ent})} & e \in e_{t-1} \\
 e^{(W_t h_{t-1}^{lm})} & \text{otherwise} \end{cases} \quad (5)
\end{align*}
\]

With the premise that paragraphs are the smallest sub-categorization where coherence and topic are defined [358], Puduppully and Lapata [255] propose a paragraph-based macro planning framework specific to the design of MLB [253] and RotoWire [347] datasets where the input to the seq2seq framework are predicted macro-plans (sequence of paragraphs). Building upon this, in contrast to precomputing global macro plans, Puduppully et al. [254] interweave the macro planning process with narrative generation where latent plans are sequentially inferred through a structured variational model as the narrative is generated conditioned on the plans so far and the previously generated paragraphs. Similarly, to establish order in the generation process, Sha et al. [292] incorporate link-based attention [114] in addition to content-based attention [11] into their framework. Similar to transitions in Markov chains [155], a link matrix \( L \in \mathbb{R}^{n_f \times n_f} \) for \( n_f \) tabular attributes defines the likelihood of transitioning from the mention of attribute \( i \) to \( j \) as \( L(f_i, f_j) \). Wang et al. [337] propose combining autoregressive modeling [287] to generate skeletal plans and using an iterative text-editing based non-autoregressive decoder [120] to generate narratives constrained on
said skeletal plans. The authors note that this approach reduces hallucination tendencies of the model. Similarly, motivated by the strong correlation observed between entity-centric metrics for record coverage and hallucinations, Liu et al. [194] adopt a two-stage generation process where a plan generator first transforms the input table records into serialized plans $R \rightarrow R + P$ based on the separator token $SEP$ and then translates the plans into narratives with the help of appended auxiliary entity information extracted through NER.

**Handcrafted templates** traditionally served as pre-defined structures where entities computed through content selection would be plugged-in. However, even in the neural D2T paradigm, inducing underlying templates helps capture the narrator voicing and stylistic representations present in the training set. As such, Ye et al. [356] extend the use of the variational autoencoders (VAEs) [163] for template induction with their variational template machine (VMT) that disentangles the latent representation of the template $z$ and the content $c$. In essence, the model can be trained to follow specific templates by sampling from $z$. Inspired from stylistic encoders [137], the authors further promote template learning by anonymizing entities in the input table thus effectively masking the content selection process. Similarly, to mitigate the strong model biases in the standard conditional VAEs [319], Chen et al. [49] estimate semantic confounders $z_c$ - linguistically similar entities to the target tokens that confound the logic of the narrative. Compared to the standard formulation $p(y|x)$, the authors employ Pearl’s do-calculus [242] to learn the objective $p(y|\text{do}(x))$ that asserts that confounder $z_c$ is no longer determined by instance $x$, thus ensuring logical consistency in the narrative. To ensure that the estimated confounders are meaningful, they are grounded through proxy variables $c$ such that confounding generation $p(c|z_m)$ can be minimized. Recently, modeling D2T and T2D as complementary tasks, Doung et al. [77] leverage the VAE formulation with the underlying architecture of a pre-trained T5 model to offer a unified multi-domain framework for the dual task. To combat the lack of parallel-corpora for the back-translation (T2D) training, the authors introduce latent variables to model the marginal probabilities of back-translation through an iterative learning process.

Likewise, for approaches beyond the use of autoencoders, Chen et al. [47] take inspiration from practices in semantic parsing [71] and propose a coarse-to-fine two-stage generation scheme. In the first stage, a template $Y_T$ containing placeholder tokens $ENT$ is generated, representing the global logical structure of the narrative. The entities are then copied over from the input data instance to replace tokens $ENT$ in the second step to generate the final narrative $Y$. Suadaa et al. [312], similarly, follow template-guided generation [151] (see §4.2) where the precomputed results of numeric operations are copied over to the template and replace the placeholder tokens. For Pre-trained Language Models (PLMs), the authors incorporate copying into the fine-tuning stage for this action.

### 5.1.4 Stylistic Encoders

In addition to the traits of coherence, fluency, and fidelity, stylistic variation is crucial to NLG [307]. It is interesting to note that the n-gram entropy of generated texts in seq2seq based NLG systems are significantly lower than that in its training data - leading to the conclusion that these systems adhere to only a handful of dominant patterns observed in the training set [237]. Thus, introducing control measures to text generation has recently garnered significant attention from the NLG community [137, 344]. As such, the semantically conditioned LSTM (SC-LSTM) proposed by Wen et al. [343] extends the LSTM cell to incorporate a one-hot encoded MR vector $d$ that takes the form of a sentence planner. Following this, Deriu and Cieliebak [62] append additional syntactic control measures to the MR vector $d$ (such as the first token to appear in the utterances and expressions for different entity-value pairs) by simply appending one-hot vectors representation of these control mechanisms to $d$. Similarly, Lin et al. [186] tackle the lack of a template-based parallel dataset with style imitation - as illustrated in Fig. 9, for each
instance \((x, y)\), an exemplar narrative \(y_e\) is retrieved from the training set based on field-overlap distance \(D(x, x_e)\) and an additional encoder is used to encode \(y_e\). The model is trained with competing objectives for content determination \(P(y|x, y_e)\) and style embodiment \(P(y_e|x_e, y_e)\) with an additional content coverage constraint for better generation fidelity.

### Table

| Name     | Food | Area     | Price   | Near       |
|----------|------|----------|---------|------------|
| Loch Fyne | Italian | Riverside | €20-25  | Strada     |

**Exemplar \(y_e\)**

*Zizzi is a pub providing fine French dining but with an expensive price, located near Coom in the city center.*

**Generation \(y\)**

*Loch Fyne provides fine Italian dining with a £20-25 price, located near Strada at the riverside.*

![Fig. 9. Style imitation with exemplar narratives - Lin et. al. [186]](image)

### 5.1.5 Graph Encoders

The use of explicit graph encoders in D2T stems from the intuition that neural graph encoders such as Graph Convolutional Networks (GCNs) [164] have strong relational inductive biases that produce better representations of input graphs [18] as an effective alternative to linearization. This entails generating representations for the nodes \(v \in V\) and edges \((u, v) \in E\) in the input graph.

#### GCNs & Graph-RNNs:

Marcheggiani and Titov [208] compute node representations \(h_v' (6)\) through explicit modeling of edge labels \(lab(u, v)\) and directions \(dir(u, v) \in \{in, out, loop\}\) for each neighboring node \(u \in N(v)\) in their GCN parameterization where learned scalar gates \(g_{u,v}\) weigh the importance of each edge. With residual \((h_v' = h_v' + h_v) [129]\) and dense \((h_\mu' = [h_v'; h_\mu]) [138]\) skip connections, Marcheggiani and Perez-Beltrachini [207] adopt the above-mentioned encoder with an LSTM decoder [201] for graph-to-text generation. Differing from previous iterations of Graph LSTMs [184], Distiawan et. al. [68] compute the hidden states of graph entities with consideration of the edges pointing to the entity from the previous entities, allowing their GTR-LSTM framework to handle non-predefined relationships. The ordering of the vertices fed into the LSTM is based on a combination of topological sort and breadth-first traversal. Inspired by hybrid traversal techniques [226, 298], Ribeiro et. al. [273] propose a dual graph encoder - first operating on a top-down traversal of the input graph where the predicate \(p\) between two nodes is used to transform labelled edges \((u_i, p, u_j)\) to two unlabelled edges \((u_i, p)\) and \((p, u_j)\) while the second operates on a bottom-up traversal where directions of edges are reversed \((u_i, u_j) \rightarrow (u_j, u_i)\).

\[
h_v' = \text{ReLU} \left( \sum_{u \in N(v)} g_{u,v} (W_{dir(u,v)} h_u + b_{lab(u,v)}) \right) \quad (6)
\]

Damonte and Cohen [61] note that GCNs can assist LSTMs in capturing re-entrant structures and long term dependencies. As such, to bridge the gap between the GCN [19, 285] and the linearized LSTM encoders in graph-to-text translation, Zhao et. al. [364] propose DualEnc which uses both to capture their complementary effects. The first GCN models the graph, retaining its structural integrity, while the second GCN serializes and re-orders the graph nodes resembling a planning stage and feeds it to the LSTM decoder.

#### GATs & Graph Transformers:

To address the shortcomings of RNN-based sequential computing, Koncel-Kedziorski et. al. [168] extend the transformer architecture [326] to graph-structured inputs with the GraphWriter. The distinction of GraphWriter from graph attention networks (GAT) [327] is
made through the contextualization of each node representation $v_i$ (7) with respect to its neighbours $u_j \in N(v_i)$ through attention mechanism $a_n$ for the $N$ attention heads. In contrast, Ribeiro et. al. [275] focus on capturing complementary graph contexts through distinct global $h_{o}^{\text{global}}$ and local $h_{o}^{\text{local}}$ message passing using GATs. Their approach to graph modeling also differs in its token-level approach for node representations with positional embeddings injected to preserve sequential order of the tokens.

$$\hat{v}_i = v_i + \|_{n=1}^{N} \sum_{N(v_i)} \alpha^n_{ij} W^n u_j \quad \& \quad \alpha^n_{ij} = a^n(v_i, u_j)$$

Song et. al. [306] enrich the training signal to the relation-aware transformer model [367] through additional multi-view autoencoding losses [264]. This detachable multi-view framework deconstructs the input graph into triple sets for the first view, reconstructed with a deep biaffine model [73], and linearizes the graph through a depth-first traversal for the second view. In contrast, Ke et. al. [158] obtain the entity and relation embeddings through contextual semantic representations with their structure-aware semantic aggregation module added to each transformer layer - the module consists of a mean pooling layer for entity and relation representations, a structure-aware self-attention layer [296], and finally a residual layer that fuses the semantic and structural representations of entities.

5.1.6 Reconstruction & Hierarchical Decoders. Input Reconstruction: Conceptualized from autoencoders [34, 304, 330], reconstruction-based models quantify the faithfulness of an encoded representation by correlating the decoded representation to the original input. As such, Wiseman et. al. [347] adopt decoder reconstruction [321] to the D2T paradigm by segmenting the decoder hidden states $h_t$ into $\frac{T}{B}$ continuous blocks $b_i$ of size at most $B$. The prediction of record $r$ from such a block $b_i$, $p(r.e, r.m|b_i)$, is modeled as $\text{softmax}(f(b_i))$ where $f$ is a convolutional layer followed by a multi-layer perceptron (MLP). To replicate the actions of an autoencoder, Chisholm et. al. [52] train a seq2seq based reverse re-encoding text-to-data model along with a forward seq2seq data-to-text model. Similarly, Roberti et. al. [277] propose a character-level GRU implementation where the recurrent module is passed as a parameter to either the encoder or the decoder depending on the forward $\hat{y} = f(x)$ or reverse $\hat{x} = g(y)$ direction. Following the mechanics of back-translation (text-to-data) [290, 321], Bai et. al. [12] extend the standard transformer decoder [326] to reconstruct the input graph by jointly predicting the node and edge labels while predicting the next token. The standard training objective of minimizing the negative log-likelihood of the conditional word probabilities $l_{std}$ is appended by a node prediction loss $l_{node}$ that minimizes the word-to-node attention distance and a edge prediction loss $l_{edge}$ that minimizes the negative log-likelihood over the projected edges. For table-structure reconstruction, Gong et. al. [109] define the reconstruction loss based on attribute prediction and content matching similar to the optimal transport distance [339]. It should be noted that these auxiliary tasks improve the model performance in few-shot settings.

Hierarchical Decoding: Similar to hierarchical encoding (see §5.1.2), hierarchical decoding intends to designate granular roles to each decoder in the hierarchy. Serban et. al. [291] show that injecting variations at the conditional output distribution does not capture high-level variations. As such, to model both high and low level variations, Shao et. al. [293] propose their planning-based hierarchical variational model (PHVM) based on the conditional variational auto-encoder [305]. PHVM follows a hierarchical multi-step encoder-decoder setup where a plan decoder first generates a subset $g$ of the input $\{d_1, ..., d_n\} \in x$. Then, in the hierarchical generation process, a sentence decoder and a word decoder generate the narrative conditioned on plan $g$. To dissipate the decoder responsibilities
in the seq2seq paradigm, Su et al. [310] propose a 4-layer hierarchical decoder where each layer is responsible for learning different parts of the output speech. The training instances are appended with part-of-speech (POS) tags such that each layer in the decoder hierarchy is responsible for decoding words associated with a specific set of POS patterns.

Hierarchical Attention-based Decoding: To alleviate omissions in narrative generation, Liu et al. [192] propose forced attention - with word-level coverage $\theta_i^t$ and attribute-level coverage $\gamma_i^t$, a new context vector $\hat{c}_t = \pi c_t + (1 - \pi) v_t$ is defined with a learnable vector $\pi$ and a compensation vector $v_t = f(\theta_i^t, \gamma_i^t)$ for low-coverage attributes $e$. To enforce this at a global scale, similar to Xu et al. [352], a loss function $L_{FA}$ based on $y_i^e$ is appended to the seq2seq loss function.

5.1.7 Regularization Techniques. Similar to regularization in the greater deep learning landscape [110], regularization practices in D2T append additional constraints to the loss function to enhance generation fidelity. As such, Mei et al. [210] introduce a coarse-to-fine aligner to the seq-to-seq framework that uses a pre-selector and refiner to modulate the standard aligner [11]. The pre-selector assigns each record a probability $p_i$ of being selected based on which the refiner re-weights the standard aligner’s likelihood $w_{ti}$ to $\alpha_{ti}$. The weighted average $z_t = \sum_i \alpha_{ti} m_i$ is used as a soft approximation to maintain the architecture differentiability. Further, the authors regularize the model with a summation of the learned priors $\sum_{i=1}^N p_i$ as an approximation of the number of selected records. Similarly, Perez-Beltrachini and Lapata [244] precompute binary alignment labels for each token in the output sequence indicating its alignment with some attribute in the input record. The prediction of this binary variable is used as an auxiliary training objective for the D2T model. For tabular datasets, Liu et al. [191] propose a two-level hierarchical encoder that breaks the learning of semantic tabular representation into three auxiliary tasks incorporated into the loss function of the model. The auxiliary sequence labeling task $L_{SL}$, learnt in unison with seq2seq learning, predicts the attribute name for each table cell. Similarly, the auto-encoder supervision $L_{AE}$ penalizes the distance between the table $z_t$ and the narrative $z_b$ representations, while the multi-label supervision task $L_{ML}$ predicts all the attributes in the given table. The individual losses, along with the language modeling loss, defines the loss function of the framework. To mitigate information hallucination and avoid the high variance exhibited by the use of policy gradients in the reinforcement-learning paradigm, Wang et al. [339] compute two losses in addition to the language modeling loss - the first checks the disagreement between the source table and the corresponding narrative through the L2 loss between their embeddings, similar to Yang et al. [354], while the second uses optimal-transport [43] based maximum flow between the narrative and input distributions $\mu$ and $v$. Tian et al. [318] propose the use of confidence priors to mitigate hallucination tendencies in table-to-text generation through learned confidence scores. At each decoding step $y_t$, instead of concatenating all the previous attention weights, only the antecedent attention weight $a_{t-1}$ is fed back to the RNN, such that an attention score $A_t$ can be used to compute how much $a_t$ affects the context vector $c_t$ - as all the source information in $c_t$ comes from $a_t$. The confidence score $C_t(y_t)$ is then used to sample target sub-sequences faithful to the source using a variational Bayes scheme [167]. Similarly, inspired by Liu et al. [191], Li et al. [180] propose two auxiliary supervision tasks incorporated into the training loss - number ranking and importance ranking, both crucial to sport summaries, modeled with pointer networks on the outputs of the row and column encoders respectively.

5.1.8 Reinforcement Learning. In the D2T premise, language-conditional reinforcement learning (RL) [200] often aids in model optimization through its role as auxiliary loss functions. While traditionally, the BLEU (see §6.1) and TF-IDF [260] scores of generated texts were used as the basis for reinforcement learning [192], Perez-Beltrachini and Lapata [244] use alignment scores
of the generated text with the target text. Similarly, Gong et al. [107] use four entity-centric metrics that center around entity importance and mention. Rebuffel et al. [262] propose a model agnostic RL framework, PARENTing which uses a combination of language model loss and RL loss computed based on PARENT F-score [65] to alleviate hallucinations and omissions in table-to-text generation. To avoid model overfitting on weaker training samples and to ensure the rewards reflect improvement made over pretraining, self-critical training protocol [272] is applied using the REINFORCE algorithm [345]. The improvement in PARENT score over a randomly sampled candidate \( y_c \) and a baseline sequence generated using greedy decoding \( y_b \) is used as the reward policy. In contrast, Zhao et al. [365] use generative adversarial networks (GANs) [111] where the generator is modeled as a policy with the current state being the generated tokens and the action defined as the next token to select. The reward for the policy is a combination of two values - the discriminator probability of the sentence being real and the correspondence between generated narrative and the input table based on the BLEU score. As RL frameworks based on singular metrics makes it difficult to simultaneously tackle the multiple facets of generation, Ghosh et al. [105] linearly combine metrics for recall, repetition, and reconstruction, along with the BLEU score, to form a composite reward function. The policy is adapted from Wang et al. [338] and trained using Maximum Entropy Inverse Reinforcement Learning (MaxEnt IRL) [368].

### 5.1.9 Fine-tuning Pretrained Language Models

Pretrained language models (PLMs) [64, 257] have been successful in numerous text generation tasks [288, 363]. The extensive pretraining grants these models certain worldly knowledge [247] such that, at times, the models refuse to generate nonfactual narratives even when fed deliberately corrupted inputs [274]. As such, Mager et al. [203] propose an alternate approach to fine-tuning GPT-2 for AMR-to-text generation where the fine-tuning is done on the joint distribution of the AMR \( x_j \) and the text \( y_i \) as \( \prod_j^N p(y_i|y_{<i}, x_{1:M}) \cdot \prod_j^M p(x_j|x_{<j}) \). On the other hand, inspired by task-adaptive pretraining strategies for text classification [122], Ribeiro et al. [274] introduce supervised and unsupervised task-adaptive pretraining stages as intermediaries between the original pretraining and the fine-tuning for graph-to-text translation. Interestingly, the authors note good performance of the task-adapted PLMs even when trained on shuffled graph representations. Chen et al. [51] note the few-shot learning capabilities of the task-adapted PLMs even when trained on shuffled graph representations. Similarly, as a light-weight alternative to fine-tuning the entire model, Li and Liang [182] take inspiration from prompting [37], and propose prefix-tuning which freezes the model parameters to only optimize the prefix, a task-specific vector prepended to the input. The authors note significant improvements in low-data settings when the prefix is initialized with embeddings from task specific words such as table-to-text. For avenues that allow leveraging the worldly knowledge of PLMS even without fine-tuning, Xiang et al. [351] leverage a combination of prompting GPT-3 for disambiguation and T5 for sentence-fusion leading to a domain-agnostic framework for data-to-text generation.

Inspired from practices in unlikelihood learning [323, 341], Nan et al. [223] model T5 as both a generator and a faithfulness discriminator with two additional learning objectives for unlikelihood and replacement detection. To train the model with said objectives, \( n \) contradictory sentences \( Y^{(i)}_{\text{True}} \) are generated for each entailed sentence \( Y^{(i)}_{\text{False}} \) wherein the discrimination probability is computed at every step of token generation. Similarly, to address omissions in D2T (§3.2), Jolly et al. [144] adapt the search-and-learn formulation [178] to a few-shot setting through a two-step finetuning process wherein a T5 model fine-tuned on the D2T task is further fine-tuned again with omitted attributes \( r.e/r.m \) reinserted to the narratives as pseudo-grouthtruths.

### 5.1.10 Supplemental Frameworks

#### Supplementary Modules

Fu et al. [98] propose the adaptation of the seq2seq framework for their partially-aligned dataset WITA using a supportiveness adaptor
and a rebalanced beam search. The pre-trained adaptor calculates supportiveness scores for each word in the generated text with respect to the input. This score is incorporated into the loss function of the seq2seq module and used to rebalance the probability distributions in the beam search. Framing the generation of narratives as a sentence fusion task, Kasner and Dušek use the pre-trained LasterTagger text editor to iteratively improve a templated narrative. Su et al. adopt a BM25 based information retrieval (IR) system to their prototype-to-generate (P2G) framework, which, aided with their BERT-based prototype selector, retrieves contextual samples for the input data instance from Wikipedia, allowing for successful few shot learning in T5. For reasoning over tabulated sport summaries, Li et al. propose a variation on GATs named as GatedGAT that operates over an entity graph modeled after the source table to aid the generation model in entity-based reasoning.

**Re-ranking & Pruning:** Dušek and Jurcicek append the seq2seq paradigm with an RNN-based re-ranker to penalize narratives with missing and/or irrelevant attributes from the beam search output. Based on the Hamming distance between two 1-hot vectors representing the presence of slot-value pairs, the classifier employs a logistic layer for a binary classification decision. Their framework, TGen, was the baseline for the E2E challenge. Following this, Juraska et al. first compute slot-alignment scores with a heuristic-based slot aligner with is used to augment the probability score from the seq2seq model. The aligner consists of a gazetteer that searches for overlapping content between the MR and its respective utterance, WordNet to account for semantic relationships, and hand-crafted rules to cover the outliers. Noting that even copy-based seq2seq models tend to omit values from the input data instance, Gehrmann et al. incorporate coverage and length penalties of Wu et al. With tunable parameters and , increases when too many generated words attend to the same input and increases with the length of the generated text. In contrast to Tu et al., however, the penalties are only used during inference to re-rank the beams. Similar to Paulus et al., authors prune beams that start with the same bi-gram to promote syntactic variations in the generated text. Similar to NLI based approaches, Harkous et al. append a RoBERTa based semantic fidelity classifier (SFC) that reranks the beam output from a fine-tuned GPT-2 model.

\[
\begin{align*}
    cp(x, y) &= \beta \cdot \sum_{i=1}^{\mid x \mid} \log(\min(\sum_{t=1}^{\mid y \mid} a_i^t, 1)) \\
    lp(y) &= \frac{(5 + \mid y \mid)^{\alpha}}{(5 + 1)^{\alpha}}
\end{align*}
\]

5.1.11 **Ensemble Learning:** Juraska et al. propose SLUG, an ensemble of three neural encoders - two LSTMs and one CNN, individually trained, for MR-text generation. The authors note that selecting tokens with the maximum log-probability by averaging over different encoders at each time steps results in incoherent narratives, thus the output candidate is selected based on the rankings among the top 10 candidates from each model. SLUG’s ensemble along with its data preprocessing and ranking schemes, as detailed in the sections above, was crowned winner of the 2017 E2E challenge. Similarly, to prompt the models in the ensemble to learn distinct sentence templates, Gehrmann et al. adopt diverse ensembling where an unobserved random variable assigns a weight to each model for each input. Constraining to trains each model on a subset of the training set thus leading to each model learning distinct templates. The final narrative is generated with a single model with the best perplexity on the validation set.
5.2 Unsupervised Learning

5.2.1 D2T Specific Pretraining. Following the successful applications of knowledge-grounded language models [4, 197], Konstas et al. [169] propose a domain-specific pretraining strategy inspired by Sennrich et al. [290] to combat the challenges in data sparsity, wherein self-training is used to bootstrap an AMR parser from the large unlabeled Gigaword corpus [225] which is in turn used to pretrain an AMR generator. Both the generator and parser adopt the stacked-LSTM architecture [350] with a global attention decoder [201]. Similarly, following success brought forth by the suite of pre-trained language models (PLMs) [37, 64, 257, 259], Chen et al. [48] propose a knowledge-grounded pretraining framework (KGPT) trained on 1.8 million graph-text pairs of their knowledge-grounded dataset KGTEXT built with Wikipedia hyperlinks matched to WikiData [332]. The framework consists of a graph attention network [327] based encoder and transformer [326] based encoders and decoders. Ke et al. [158] propose three graph-specific pretraining strategies based on the KGTEXT dataset - reconstructing masked narratives based on the input graph, conversely, reconstructing masked graph entities based on the narrative, and matching the graph and narrative embeddings with optimal transport. Similarly, Agarwal et al. [2] verbalize the entirety of the Wikidata Corpus [332] with two-step fine-tuning for T5 [259] to construct their KeLM corpus. The authors utilize this corpus to train knowledge-enhanced language models for downstream NLG tasks with significant improvements shown in the performance of REALM [123] and LAMA [247] for both retrieval as well as question answering tasks. Similarly, specifically geared for logical inference from tables, Liu et al. [187] propose PLoG wherein a PLM is first pre-trained on table-to-logic conversion intended to aid logical table-to-text generation for downstream datasets to the likes of LogicNLG.

5.2.2 Autoencoders. For a D2T framework solely based on unlabeled text, Freitag and Roy [95] adapt the training procedure of a denoising auto-encoder (DAE) [329] to the seq2seq framework with the notion of reconstructing each training example from a partially destroyed input. For each training instance $x_i$, a percentage $p$ (sampled from a Gaussian distribution) of words are removed at random to get a partially destroyed version $\hat{x}_i$. However, the authors note carry over of this unsupervised approach across further D2T tasks such as the WebNLG challenge [99] could be limited by the fact that the slot names in WebNLG contributes to the meaning representation.

5.3 Innovations outside of the Seq2Seq Framework

5.3.1 Template Induction. For enhanced interpretability and control in D2T, Wiseman et al. [348] propose a neural parameterization of the hidden semi-markov model (HSMM) [221] that jointly learns latent templates with generation. With discrete latent states $z_t$, length variable $l_t$, and a deterministic binary variable $f_t$ that indicates whether a segment ends at time $t$, the HSMM is modeled as a joint-likelihood (10). Further, associated phrases from $x$ can be mapped to latent states $z_t$ such that common templates can be extracted from the training dataset as a sequence of latent states $z' = \{z_1',...,z_S'\}$. Thus, the model can then be conditioned on $z'$ to generate text set to the template. Following this, Fu et al. [97] propose template induction by combining the expressive capacity of probabilistic models [222] with graphical models in an end-to-end fashion using a conditional random field (CRF) model with Gumbel-softmax used to relax the categorical sampling process [141]. The authors note performance gains on HSMM-based baselines while also noting that neural seq2seq fare better than both.

$$P(y, z, l, f|x) = \prod_{t=0}^{T-1} P(z_{t+1}, I_{t+1}|z_t, I_t, x)^{f_t} \times \prod_{t=1}^{T} P(y_{t-l_t+1:t}|z_t, I_t, x)^{f_t} \quad (10)$$
5.3.2 Discrete Neural Pipelines. While the traditional D2T pipeline observes discrete modeling of the content planning and linguistic realization stages [270], neural methods consolidate these discrete steps into end-to-end learning. With their neural referring expressions generator NeuralREG [89] appended to discrete neural pipelines, and the GRU [54] and transformer [326] as base models for both, Ferreira et. al. [91] compare neural implementations of these discrete pipelines to end-to-end learning. In their findings, authors note that the neural pipeline methods generalize better to unseen domains than end-to-end methods, thus alleviating hallucination tendencies. This is also corroborated by findings from Elder et. al. [82].

5.3.3 Computational Pragmatics. Pragmatic approaches to linguistics naturally correct under-informativeness problems [117, 134] and are often employed in grounded language learning [206, 215]. Shen et. al. [297] adopt the reconstructor-based [96] and distractor-based [58] models of pragmatics to MR-to-text generation. These models extend the base speaker models $S_0$ using reconstructor $R$ and distractor $D$ based listener models $L(y|x) \in \{L^R, L^D\}$ to derive pragmatic speakers $S_1(y|x) \in \{S^R_1, S^D_1\}$ (11,12) where $\lambda$ and $\alpha$ are rationality parameters controlling how much the model optimizes for discriminative outputs.

$$S^R_1(y|x) = L^R(x|y)^\lambda \cdot S_0(y|x)^{1-\lambda} \quad (11)$$

$$S^D_1(y|x) \propto L^D(x|y)^\alpha \cdot S_0(y|x) \quad (12)$$

6 EVALUATION OF DATA-TO-TEXT SYSTEMS

In this section, we take a deeper look into the specifics of evaluation for D2T systems. Traditionally, the evaluation of D2T systems is compartmentalized into either *intrinsic* or *extrinsic* measures [26]. The former either uses automated metrics to compare the generated narrative to a reference text or employs for human judgement [146] - both evaluating the properties of the system output. The latter focuses on the ability of the D2T system to fulfill its intended purpose of imparting information - to what degree does the system achieve its overarching task for which it was developed. From the analysis of 79 papers spanning 2005-2014, Gkatzia and Mahamood [106] note the overwhelming prevalence of intrinsic evaluation with 75.7% of articles reporting it compared to 15.1% that report an extrinsic measure. This is unsurprising, as intrinsic evaluation can be automated and is often convenient, not requiring additional crowd-sourced human labor and collection of feedback from deployed systems. As such, Reiter [267] notes the importance of extrinsic (pragmatic) evaluation and its absence in the field. Thus, the absence of literature in extrinsic evaluation measures leads us to focus on the innovations in improving the quality of intrinsic evaluation metrics (§6.2). For a broader view on the evaluation of text generation systems in the greater NLG landscape, we refer the readers to a recent survey of evaluation practices for text generation systems by Celikyilmaz et. al. [39].

6.1 BLEU: the false prophet for D2T

With the abundance of paired datasets where each data instance is accompanied by a human generated reference text, often referred to as the *gold standard*, the NLG community has sought after quick, cheap, and effective metrics for evaluation of D2T systems. The adoption of automated metrics such as BLEU, NIST, and ROUGE, by the machine translation (MT) community, by the virtue of their correlation with human judgement [69, 185, 238], similarly carried over to the D2T community. Among them, Belz and Gatt [22] note that NIST best correlates with human judgements on D2T texts, when compared to 9 human domain-experts and 21 non-experts. However, they note that these n-grams based metrics perform poorer in D2T as compared to MT due to the
domain-specific nature of D2T systems wherein the generated texts are judged better by humans than human-written texts. From a review of 284 correlations reported in 34 papers, Reiter [268] notes that the correlations between BLEU and human evaluations are inconsistent - even in similar tasks. While automated metrics can aid in the diagnostic evaluation of MT systems, the author showcases the weakness of BLEU in the evaluation of D2T systems. This notion has been resonated several times [269, 286]. On top of this, undisclosed parameterization of these metrics and the variability in the tokenization and normalization schemes applied to the references can alter this score by up to 1.8 BLEU points for the same framework [250]. Similarly, it has also been shown that ROUGE tends to favor systems that produce longer summaries [313]. Further complicating the evaluation of D2T is the fact modern frameworks are neural - comparing score distributions, even with the aid of statistical significance tests, are not as meaningful due to the non-deterministic nature of neural approaches and accompanying randomized training procedures [265].

6.2 Innovations in Intrinsic Evaluation

Noting the shortcomings of prevalent word-overlap metrics (§6.1), alternative automated metrics for intrinsic evaluation have been proposed (§6.2.1 & §6.2.2). To account for divergence in reference texts, Dhingra et. al. [65] propose PARENT - a metric that computes precision and recall of the generated narrative \( \hat{y} \) with both the gold narrative \( y \) and its entailment to the semi-structured tabular input \( x \).

6.2.1 Extractive Metrics. With dialogue generation models adopting classification-backed automated metrics [152, 179], Wiseman et. al. [347] propose a relation extraction system to the likes of [60, 72] wherein the record type \( r.t \) is predicted using its corresponding entity \( r.e \) and value \( r.m \) as \( p(r.t|e, m; \theta) \). With such a relation extraction system, the authors propose three metrics for automated evaluation:

- **Content Selection (CS)** is represented by the precision and recall of unique relations extracted from \( \hat{y}_{1:t} \) that are also extracted from \( y_{1:t} \).
- **Relation Generation (RG)** is represented by the precision and number of unique relations extracted from \( \hat{y}_{1:t} \) that can be traced to \( x \).
- **Content Ordering (CO)**, similarly, by the normalized Damerau-Levenshtein distance [36] between the sequence of records extracted from \( y_{1:t} \) and \( \hat{y}_{1:t} \).

The authors note that, given the two facets of D2T, CS pertains to what to say and CO to how to say it while RG pertains to both (factual correctness).

6.2.2 Contextualized Metrics. Böhm et. al. [32] note that while modern frameworks for text generation compete with higher scores on automated word-overlap metrics, the quality of the generation leaves a lot to be desired. As such, the adaptation of continuous representations based metrics shifts the focus from surface-form matching to semantic matching. Zhang et. al. [362] introduce BERTScore which computes similarity scores for tokens in the system and reference text based on their BERT [64] embeddings while Mathur et. al. [209] devise supervised and unsupervised metrics for NMT based on the same BERT embeddings - both having substantially higher correlation to human judgement compared to standard word-overlap metrics (see §6.1). Following this, Clark et. al. [56] extend the word mover’s distance [173] to multi-sentence evaluation using ELMo representations [245]. Zhao et. al. [366] propose the MoverScore which uses contextualized embeddings from BERT where the aggregated representations are computed based on power means [281]. Dušek and Kasner [80] employ RoBERTa [196] for natural language inference (NLI), where, for a given hypothesis and premise, the model computes scores for entailment between the two. While lower
scores for forward entailment can point to omissions, backward entailment scores correspondingly point to hallucinations. Similarly, Chen et al. [47] propose parsing-based and adversarial metrics to the evaluate model correctness in logical reasoning.

6.2.3 Human Judgement. While human judgment is often considered to be the ultimate D2T evaluation measure, they are subject to a high degree of inconsistency (even with the same utterance), which may be attributed to the judge’s individual preferences [63, 333] - an issue that could be circumvented through a larger sample size, however such an endeavor is accompanied with an equal increase in cost for data acquisition. As such, there have been several recommendations on the proper usage of ratings and Likert scales [143, 146, 165]. From the analysis of 135 papers from specialized NLG conferences, Amidei et al. [5] note that several studies employ the Likert scale on an item-by-item basis in contrast to its design as an aggregate scale, and the analysis performed on these scales with parametric statistics do not disclose the assumptions about the distribution of the population probability. Howcroft et al. [135], on the other hand, note that the definitions of fluency and accuracy for which these scales are employed lack consistency among the papers investigated. To mitigate these inconsistencies through good experimental design, Novikova et al. [231] propose RankME, a relative-ranking based magnitude estimation method that combines the use of continuous scales [23, 113], magnitude estimation [301], and relative assessment [38]. Further, to address the quadratic growth of data required for cross-system comparisons, the authors adopt TrueSkill [132], a bayesian data-efficient ranking algorithm used in MT evaluation [33], to RankME. The HumEval workshop [21, 24, 25] has been an invaluable resource in investigating the shortcomings and building maps to better practices in human evaluations.

6.3 Emphasis on Reproducibility
The last half-decade has seen the ML community place significant emphasis on the reproducibility of academic results [248, 302]. However, the focus of these reproducibility efforts are placed on automated metrics (§6.1, §6.2.1, §6.2.2) with the reproducibility of human evaluation results receiving far less attention. As human evaluation is often considered the ultimate measure of D2T, Belz et al. [27] initiate ReproGen, a shared task focused on reproducing the results of human evaluations - intended to shed better light on the reproducibility of human evaluations and the possible interventions in design and execution of human evaluations to make them more reproducible. The authors note the inconsistency (expected) of the evaluators across different studies and hence point towards the use of metadata standardization through data-sheets such as HEDS [299]. Providing a complementary view, van der Lee et al. [325] review practices in human evaluation from 304 publications in the International Conference in Natural Language Generation (INLG) and the Annual Meeting of the Association of Computation Linguistics (ACL) from 2018 to 2019 and outline the severe discrepancy in the spectrum of evaluator demographics and sample sizes, design practices, evaluation criteria and put forth some common ground through a set of best practices for conducting human evaluations.

7 CONCLUSION AND FUTURE DIRECTIONS
As delineated in §3 & §4, innovations in D2T take inspiration from several facets of NLG and ML. From alterations to the seq2seq, pretraining-finetuning, autoencoding, ensemble-learning, and reinforcement-learning paradigms, to domain-specific data preprocessing and data encoding strategies, the prospects for innovations in D2T appear as grand as that for the NLG landscape itself. Alongside, progresses made in non-anglocentric datasets, datacards that reinforce accountability, and metrics that offer heuristic evaluation, aid in elevating D2T standards. As NLG research evolves, so will D2T, and vice-versa. In the following sections, we impart our thoughts for future directions
for each facet of D2T - the desiderata for D2T dataset design (§7.1), a forward look at the possibilities for approaches and architectures for D2T (§7.2) and finally, closing thoughts on the future of D2T evaluation (§7.3).

7.1 Desiderata for D2T Datasets

In §2, we outlined the development of parallel corpora with data-narrative pairs alongside dominant benchmark datasets in each task category. In addition to these benchmarks, it is as crucial to acknowledge niche datasets - Obeid and Hoque [233] compile a collection of 8,305 charts with their respective narratives, followed by Chart-to-text [153], that encompasses 44,096 multi-domain charts. The shared gains and pitfalls in dataset design across the D2T task categories, as discussed in §2, offer insights that can aid the construction of future datasets with the potential to challenge the current paradigm:

• **Domain Agnosticism**: Although domain-specific datasets allow the models to learn and leverage domain-specific conventions for performance gains in niche tasks, the resulting models are less malleable to unseen domains. To be adaptable and deployable for unseen niche tasks that may vary based on user requirements, it is crucial that the datasets used to train D2T models are not restricted to a single domain to avoid over-fitting on domain-specific keywords.

• **Dataset Consistency**: Often, the greatest challenges for D2T systems, namely hallucination and omission (see §3.2), can be traced back to the datasets. Datasets facing divergence (as outlined in §2.3), wherein the narratives are not consistent with the data instances or vice-versa, often lead to models that hallucinate or omit important aspects of the data [280].

• **Human-crafted References**: Often, to replicate human linguistics, datasets in NLP/NLG contain human annotations (narratives), considered as gold references. Reiter [267] notes that D2T datasets, unintentionally, may contain machine-generated annotations, such as those for WeatherGov, and urges the community to focus on human-centered narratives.

• **Linguistic Diversity**: It is vital to acknowledge that the majority of the D2T benchmark datasets are anglo-centric. Joshi et. al [147] note that models built on non-anglo-centric datasets, which are fewer and far between, have the potential to impact many more people than models built on highly resourced languages. The WebNLG 2020 challenge[10], for instance, encourages submissions for both English and Russian parsing.

7.2 Approaches to Data-to-text Generation: Looking Forward

In the above sections §4 and §5, we have extensively outlined the recent innovations in D2T both inside and outside of seq2seq modeling. However, looking forward, with the emergence of highly capable large language models (LLMs) such as ChatGPT [235], below we discuss the reconciliation of these emergent technologies with the current D2T paradigm:

• **Few-shot Learning**: Data-to-text generation is a task that requires extrapolation beyond general linguistic understanding and commonsense reasoning, thus general LLM prompting strategies [340] may not be suited for this endeavor. Adding to that the data sparsity prevalent in D2T, extensions of prefix-tuning [182] and in-context sample search [189] may be especially favorable for building strong subset of samples for few-shot learning in LLMs.

• **Deviation from Task-specific Architectures**: The paradigm for data-to-text generation, as it stands now, prefers custom architectures, and rightfully so - they allow focused modeling of entities (§5.1.1) and dedicated attention mechanisms (§5.1.2 & §5.1.6) to combat data infidelity that occurs as a consequence of RNN’s lack of long-form coherence.

---

[10]https://webnlg-challenge.loria.fr/challenge_2020/
Transformer-based LLMs, however, may inadvertently model these dependencies and attention mechanisms as a function of their self-attention modules, thus allowing the convergence towards a universal architecture.

- **Effective Linearization:** In line with the point above, the preference for plan-based approaches to D2T (§5.1.3) stems from issues in coherence. While there has been extensive work in linearizing graphs and tables (§4.2) showcasing that linearization in LLMs can be as effective, if not more, compared to dedicated encoders designed to capture inductive biases (§5.1.1, §5.1.2, §5.1.5), recent work suggests that LLMs can handle unexpected tasks simply through their linear transcription into linguistic sentences (the LIFT framework) [67]. This line of work has extensive potential for modeling plans through their transcriptions into sentences.

- **Numeracy for Data-to-text Generation:** The NLP niche of building LLMs capable of quantitative reasoning (often referred to as Math-AI or Math-NLP) has garnered significant interest from the research community [295, 317, 334]. Although there are works in D2T that incorporate this aspect [107], the two research niches are often disparate. D2T, a field that aims to combat hallucination of data points, has a lot to gain from the advances in Math-NLP that enable models to better reason about said data points.

- **Interfacing with External APIs:** In line with the above point, interfacing LLMs with computational APIs (Wolfram Alpha [349], ToolFormer [284]) has showcased significant enhancement potential for these already capable models. This paves a path in D2T where we deviate from viewing the generation of narratives as a sequential input-to-output mapping but rather a more involved loop comprising of numerical and logical reasoners, computational engines, pattern matchers, and validators that combine to form the greater NLG pipeline.

### 7.3 The Future of D2T Evaluation

While NLP systems generally have benchmark datasets that closely resemble the target tasks that these systems are intended to be deployed for (summarization, sentiment analysis, and language translation), data-to-text systems are highly specialized to the incoming data stream which differs from user to user. Thus, a one-size-fits-all approach to benchmarking, especially with automated metrics on benchmark datasets, can not showcase the utility of these systems in the real world, leading to an urgency for practical tools for extrinsic evaluation. Additionally, besides the need for fluidity and fidelity, systems placed in the real world require accountability [214]. The current leaderboard system poses the risk of blind metric optimization with disregard to model size and fairness [85]. For a holistic approach to evaluation, Gehrmann et al. [102] propose a living benchmark, GEM, similar to likes to Dynabench [161], providing challenge sets (a set of curated test sets intended to be challenging) and benchmark datasets accompanied with their D2T-specific data cards [28].

Further, as unified D2T frameworks become more decentralized with growing user-bases, the designers of these systems can utilize the user-interaction logs as measures for extrinsic evaluation, similar to the likes of ChatGPT [235]. While the D2T community places greater emphasis on measures to evaluate the quality of the generated narrative, the utility of these narratives can be evaluated with task effectiveness with and without the presence of narratives [55, 336].

### 8 ACKNOWLEDGEMENTS

This work is supported in part by US National Science Foundation grant DGE1545362.
REFERENCES

[1] Rob Abbott, Brian Ecker, Pranav Anand, and Marilyn Walker. 2016. Internet argument corpus 2.0: An sql schema for dialogic social media and the corpora to go with it. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16). 4445–4452.

[2] Oshin Agarwal, Heming Ge, Smak Shakerei, and Rami Al-Rfou. 2021. Knowledge Graph Based Synthetic Corpus Generation for Knowledge-Enhanced Language Model Pre-training. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 3554–3565.

[3] Shubham Agarwal and Marc Dymetman. 2017. A surprisingly effective out-of-the-box char2char model on the E2E NLG Challenge dataset. In Proceedings of the 18th Annual SKGidial Meeting on Discourse and Dialogue. 158–163.

[4] Sungjin Ahn, Heeyoul Choi, Tanel Parnamaa, and Yoshua Bengio. 2016. A neural knowledge language model. arXiv preprint arXiv:1608.00318 (2016).

[5] Jacopo Amidei, Paul Piwek, and Alistair Willis. 2019. The use of rating and Likert scales in Natural Language Generation human evaluation tasks: A review and some recommendations. In Proceedings of the 12th International Conference on Natural Language Generation. 397–402.

[6] Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elghohary, Sergey Feldman, Vu Ha, et al. 2018. Construction of the Literature Graph in Semantic Scholar. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers), 84–91.

[7] Ewa Andrezjuk, Julian Eisenschlos, Francesco Piccinno, Syrine Krichene, and Yasemin Aytun. 2022. Table-To-Text generation and pre-training with TabT5. In Findings of the Association for Computational Linguistics: EMNLP 2022. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 6758–6766. https://aclanthology.org/2022.findings-emnlp.503

[8] Gabor Angeli, Percy Liang, and Dan Klein. 2010. A simple domain-independent probabilistic approach to generation. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing. 502–512.

[9] Luca Anselma and Alessandro Mazzei. 2018. Designing and testing the messages produced by a virtual dietitian. In 11th International Conference on Natural Language Generation (INLG 2018). Association for Computational Linguistics, 244–253.

[10] Tatsuya Aoki, Akira Miyazawa, Tatsuya Ishigaki, Keiichi Goshima, Kasumi Aoki, Ichiro Kobayashi, Hiroya Takamura, and Yusuke Miyao. 2018. Generating Market Comments Referring to External Resources. In Proceedings of the 11th International Conference on Natural Language Generation. Association for Computational Linguistics, 135–139. https://doi.org/10.18653/v1/W18-6515

[11] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In ICLR.

[12] Xuefeng Bai, Linfeng Song, and Yue Zhang. 2020. Online Back-Parsing for AMR-to-Text Generation. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). 1206–1219.

[13] Laura Baneresu, Claire Bonial, Smak Shakerei, Rami Al-Rfou, and Ulf Hermjakob. 2021. Knowledge Graph Based Synthetic Corpus Generation for Knowledge-Enhanced Language Model Pre-training. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 3554–3565.

[14] Regina Barzilay and Lillian Lee. 2004. Catching the Drift: Probabilistic Content Models, with Applications to Generation and Summarization. In Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004. Association for Computational Linguistics, 113–120.

[15] Regina Barzilay and Mirella Lapata. 2005. Collective Content Selection for Concept-to-Text Generation. In Proceedings of the 12th International Conference on Natural Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 331–338.

[16] Regina Barzilay and Lillian Lee. 2004. Catching the Drift: Probabilistic Content Models, with Applications to Generation and Summarization. In Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004. Association for Computational Linguistics, 113–120.

[17] Regina Barzilay and Kathleen R McKeown. 2005. Sentence fusion for multidocument news summarization. Computational Linguistics 31, 3 (2005), 297–328.

[18] Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Flores Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Yashar Ganjehlou, Jong Hoon Hwang, Russell A. Adams, William M. Dai, Rebeccah S. Schrock, Christopher D. Buehler, Aravind Srinivasan, Guillaume Desjardins, Matthew Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. 2018. Relational inductive biases, deep learning, and graph networks. CoRR abs/1806.01261 (2018). arXiv:1806.01261

[19] Daniel Beck, Gholamreza Haffari, and Trevor Cohn. 2018. Graph-to-Sequence Learning using Gated Graph Neural Networks. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 273–283.
Innovations in Neural Data-to-text Generation: A Survey

Jan Milan Deriu and Mark Cieliebak. 2018. Syntactic manipulation for generating more diverse and interesting texts.

Marco Damonte and Shay B Cohen. 2019. Structural Neural Encoders for AMR-to-text Generation.

Emilie Colin and Claire Gardent. 2019. Generating text from anonymised structures.

Reuben Cohn-Gordon, Noah Goodman, and Christopher Potts. 2018. Pragmatically Informative Image Captioning.

Zhiyu Chen, Harini Eavani, Wenhu Chen, Yinyin Liu, and William Yang Wang. 2020. Few-Shot NLG with Pre-Trained.

Wenhu Chen, Yu Su, Xifeng Yan, and William Yang Wang. 2020. KGPT: Knowledge-Grounded Pre-Training for Data-to-Text Generation.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2019. Tabfact: A large-scale dataset for table-based fact verification.

Zhiyu Chen, Harini Eavani, Wenhu Chen, Yinyin Liu, and William Yang Wang. 2020. Few-Shot NLG with Pre-Trained Language Model.

Andrew Chisholm, Will Radford, and Ben Hachey. 2017. Learning to generate one-sentence biographies from Wikidata.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2019. Tabfact: A large-scale dataset for table-based fact verification.

Kyunghyun Cho. 2014. On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. In Proceedings of SSST-8, Eighth Workshop on Syntax; Semantics and Structure in Statistical Translation. 103–111.

Arjun Choudhry, Mandar Sharma, Pramod Chundury, Thomas Kapler, Derek WS Gray, Naren Ramakrishnan, and Niklas Elmqvist. 2020. Once upon a time in visualization: Understanding the use of textual narratives for causality. IEEE Transactions on Visualization and Computer Graphics 27, 2 (2020), 1332–1342.

Elizabeth Clark, Asli Celikyilmaz, and Noah A Smith. 2019. Sentence mover’s similarity: Automatic evaluation for multi-sentence texts. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2748–2760.

Herbert H Clark. 1977. Comprehension and the given-new contract. Discourse production and comprehension (1977), 1–40.

Reuben Cohn-Gordon, Noah Goodman, and Christopher Potts. 2018. Pragmatically Informative Image Captioning with Character-Level Inference. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). 439–443.

Emilie Colin and Claire Gardent. 2019. Generating text from anonymised structures. In Proceedings of the 12th International Conference on Natural Language Generation. 112–117.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. Journal of machine learning research 12, ARTICLE (2011), 2493–2537.

Marco Damonte and Shay B Cohen. 2019. Structural Neural Encoders for AMR-to-text Generation. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). 3649–3658.

Jan Milan Deriu and Mark Cieliebak. 2018. Syntactic manipulation for generating more diverse and interesting texts. In 11th International Conference on Natural Language Generation (INLG 2018), Tilburg, The Netherlands, 5-8 November 2018. Association for Computational Linguistics, 22–34.
[63] Nina Dethlefs, Heriberto Cuayahuitl, Helen Hastie, Verena Rieser, and Oliver Lemon. 2014. Cluster-based prediction of user ratings for stylistic surface realisation. In Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics. 702–711.

[64] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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). 4171–4186.

[65] Bhuwan Dhingra, Manaal Faruqui, Ankur Parikh, Ming-Wei Chang, Dipanjan Das, and William Cohen. 2019. Handling Divergent Reference Texts when Evaluating Table-to-Text Generation. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 4884–4895.

[66] Nicholas Diakopoulos. 2019. Automating the news. In Automating the News. Harvard University Press.

[67] Tuan Dinh, Yuchen Zeng, Ruisu Zhang, Ziqian Lin, Michael Gira, Shashank Rajput, Jy-yong Sohn, Dimitris Papailopoulos, and Kangwook Lee. 2022. LiFT: Language-Interfaced Fine-Tuning for Non-language Machine Learning Tasks. In Advances in Neural Information Processing Systems.

[68] Bayu Distiawan, Jianzhong Qi, Rui Zhang, and Wei Wang. 2018. GTR-LSTM: A triple encoder for sentence generation from RDF data. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 1627–1637.

[69] George Doddington. 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. 138–145.

[70] Chenhe Dong, Yinhui Li, Haifan Gong, Miaoxin Chen, Junxin Li, Ying Shen, and Min Yang. 2021. A Survey of Natural Language Generation. arXiv preprint arXiv:2112.11739 (2021).

[71] Li Dong and Mirella Lapata. 2018. Coarse-to-Fine Decoding for Neural Semantic Parsing. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 731–742.

[72] Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2015. Classifying Relations by Ranking with Convolutional Neural Networks. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). 626–634.

[73] Timothy Dozat and Christopher D Manning. 2017. Deep Biaffine Attention for Neural Dependency Parsing. (2017).

[74] Nan Duan, Duyu Tang, Peng Chen, and Ming Zhou. 2017. Question Generation for Question Answering. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 866–874. https://doi.org/10.18653/v1/D17-1090

[75] Pablo Ariel Duboue and Kathleen R. McKeeown. 2003. Statistical Acquisition of Content Selection Rules for Natural Language Generation. In Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing. 121–128.

[76] Sebastian Duerr and Peter A Gloor. 2021. Persuasive Natural Language Generation—A Literature Review. arXiv preprint arXiv:2101.05786 (2021).

[77] Song Duong, Alberto Lumbiras, Mike Gartrell, and Patrick Gallinari. 2023. Learning from Multiple Sources for Data-to-Text and Text-to-Data. In International Conference on Artificial Intelligence and Statistics (AISTATS).

[78] Ondřej Dušek, David M Howcroft, and Verena Rieser. 2019. Semantic Noise Matters for Neural Natural Language Generation. In Proceedings of the 12th International Conference on Natural Language Generation. 421–426.

[79] Ondřej Dušek and Filip Jurcicek. 2016. Sequence-to-Sequence Generation for Spoken Dialogue via Deep Syntax Trees and Strings. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). 45–51.

[80] Ondřej Dušek and Zdeněk Kasner. 2020. Evaluating Semantic Accuracy of Data-to-Text Generation with Natural Language Inference. In Proceedings of the 13th International Conference on Natural Language Generation. 131–137.

[81] Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2018. Findings of the E2E NLG Challenge. In Proceedings of the 11th International Conference on Natural Language Generation. 322–328.

[82] Henry Elder, Jennifer Foster, James Barry, and Alexander O’Connor. 2019. Designing a Symbolic Intermediate Representation for Neural Surface Realization. In Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation. 65–73.

[83] Henry Elder and Chris Hokamp. 2018. Generating High-Quality Surface Realizations Using Data Augmentation and Factored Sequence Models. In Proceedings of the First Workshop on Multilingual Surface Realisation. Association for Computational Linguistics, 49–53. https://doi.org/10.18653/v1/W18-3606

[84] Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Amuj Goyal, Peter Ku, and Dilek Hakkani-Tür. 2020. MultiWOZ 2.1: A Consolidated Multi-Domain Dialogue Dataset with State Corrections and State Tracking Baselines. In Proceedings of the 12th Language Resources and Evaluation Conference. 422–428.
Innovations in Neural Data-to-text Generation: A Survey

[85] Kawin Ethayarajh and Dan Jurafsky. 2020. Utility is in the Eye of the User: A Critique of NLP Leaderboards. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). 4846–4853.

[86] Angela Fan, Claire Gardent, Clloë Braud, and Antoine Bordes. 2019. Using Local Knowledge Graph Construction to Scale Seq2Seq Models to Multi-Document Inputs. In 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing.

[87] Christiane Fellbaum. 1998. A semantic network of English: the mother of all WordNets. In EuroWordNet: A multilingual database with lexical semantic networks. Springer, 137–148.

[88] Thiago Castro Ferreira, Iacer Calixto, Sander Wubben, and Emil Krahmer. 2017. Linguistic realisation as machine translation: Comparing different MT models for AMR-to-text generation. In Proceedings of the 10th International Conference on Natural Language Generation. 1–10.

[89] Thiago Castro Ferreira, Diego Moussallem, Ákos Kádár, Sander Wubben, and Emil Krahmer. 2018. NeuralREG: An end-to-end approach to referring expression generation. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 1959–1969.

[90] Thiago Castro Ferreira, Diego Moussallem, Emil Krahmer, and Sander Wubben. 2018. Enriching the WebNLG corpus. In Proceedings of the 11th International Conference on Natural Language Generation. 171–176.

[91] Thiago Castro Ferreira, Chris van der Lee, Emil Van Miltenburg, and Emil Krahmer. 2019. Neural data-to-text generation: A comparison between pipeline and end-to-end architectures. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). 552–562.

[92] Thiago Castro Ferreira, Helena Vaz, Brian Davis, and Adriana Pagano. 2021. Enriching the E2E dataset. In Proceedings of the 14th International Conference on Natural Language Generation. 177–183.

[93] Jenny Rose Finkel, Trond Grenanger, and Christopher D Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In Proceedings of the 43rd annual meeting of the association for computational linguistics (ACL’05), 363–370.

[94] Mary Ellen Foster. 2019. Natural language generation for social robotics: opportunities and challenges. Philosophical Transactions of the Royal Society B 374, 1771 (2019), 20180027.

[95] Markus Freitag and Scott Roy. 2018. Unsupervised Natural Language Generation with Denoising Autoencoders. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. 3922–3929.

[96] Daniel Fried, Jacob Andreas, and Dan Klein. 2018. Unified Pragmatic Models for Generating and Following Instructions. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). 1951–1963.

[97] Yao Fu, Chuanqi Tan, Bin Bi, Mosha Chen, Yansong Feng, and Alexander Rush. 2020. Latent Template Induction with Gumbel-CRFs. Advances in Neural Information Processing Systems 33 (2020), 20259–20271.

[98] Zihao Fu, Wei Shi, Wai Lam, Lidong Bing, and Zhiyuan Liu. 2020. Partially-Aligned Data-to-Text Generation with Distinct Supervision. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). 9183–9193.

[99] Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating Training Corpora for NLG Micro-Planners. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, 179–188. https://doi.org/10.18653/v1/P17-1017

[100] Albert Gatt and Emil Krahmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. Journal of Artificial Intelligence Research 61 (2018), 65–170.

[101] Ruifang Ge and Raymond Mooney. 2005. A statistical semantic parser that integrates syntax and semantics. In Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005). 9–16.

[102] Sebastian Gehrmann, Tosin Adewumi, Karmany Aggarwal, Pawan Sasanka Ammanamanchi, Anuoluwapo Aremu, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna-Adriana Clinciu, Dipanan Das, Kaustubh Dhole, et al. 2021. The GEM Benchmark: Natural Language Generation, its Evaluation and Metrics. In Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021). 96–120.

[103] Sebastian Gehrmann, Falcon Dai, Henry Elder, and Alexander M Rush. 2018. End-to-End Content and Plan Selection for Data-to-Text Generation. In Proceedings of the 11th International Conference on Natural Language Generation. 46–56.

[104] Nahum Gershon and Ward Page. 2001. What storytelling can do for information visualization. Commun. ACM 44, 8 (2001), 31–37.

[105] Sayan Ghosh, Zheng Qi, Snigdha Chaturvedi, and Shashank Srivastava. 2021. How Helpful is Inverse Reinforcement Learning for Table-to-Text Generation?. In Proceedings of the 9th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers). 71–79.

[106] Dimitra Gkatzia and Saad Mahamood. 2015. A snapshot of NLG evaluation practices 2005-2014. In Proceedings of the 15th European Workshop on Natural Language Generation (ENLG). 57–60.
[107] Heng Gong, Wei Bi, Xiaocheng Feng, Bing Qin, Xiaojian Liu, and Ting Liu. 2020. Enhancing content planning for table-to-text generation with data understanding and verification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. 2905–2914.

[108] Heng Gong, Xiaocheng Feng, Bing Qin, and Ting Liu. 2019. Table-To-Text Generation with Effective Hierarchical Encoder on Three Dimensions (Row, Column and Time). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 3143–3152.

[109] Heng Gong, Yawei Sun, Xiaocheng Feng, Wei Bi, Xiaojian Liu, and Ting Liu. 2020. Tablegpt: Few-shot table-to-text generation with table structure reconstruction and content matching. In *Proceedings of the 28th International Conference on Computational Linguistics*. 1978–1988.

[110] Ian Goodfellow, Y Bengio, and A Courville. 2016. Regularization for deep learning. *Deep learning* (2016), 216–261.

[111] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems* 27 (2014).

[112] Raghav Goyal, Marc Dymetman, and Eric Gaussier. 2016. Natural Language Generation through Character-based RNNs with Finite-state Prior Knowledge. In *COLING*. 1083–1092.

[113] Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2013. Continuous measurement scales in human evaluation of machine translation. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*. 33–41.

[114] Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, et al. 2016. Hybrid computing using a neural network with dynamic external memory. *Nature* 538, 7626 (2016), 471–476.

[115] Thomas RG Green and Marian Petre. 1992. When visual programs are harder to read than textual programs. In *Human-Computer Interaction: Tasks and Organisation, Proceedings ECCE-6 (6th European Conference Cognitive Ergonomics)*, Vol. 57. Citeseer.

[116] Thomas RG Green, Marian Petre, and Rachel KE Bellamy. 1991. *Comprehensibility of visual and textual programs: a test of superlativism against the “match-mismatch” conjecture*. Open University, Computer Assisted Learning Research Group.

[117] Herbert P Grice. 1975. Logic and conversation. In *Speech acts*. Brill, 41–58.

[118] Barbara J Grosz, Scott Weinstein, and Aravind K Joshi. 1995. Centering: a framework for modeling the local coherence of discourse. *Computational Linguistics* 21, 2 (1995), 203–225.

[119] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating Copying Mechanism in Sequence-to-Sequence Learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1631–1640.

[120] Jiatao Gu, Changhan Wang, and Junbo Zhao. 2019. Levenshtein transformer. *Advances in Neural Information Processing Systems* 32 (2019).

[121] Çağlar Gülçehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the Unknown Words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 140–149.

[122] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Bellagy, Doug Downey, and Noah A Smith. 2020. Don’t Stop Pretraining: Adapt Language Models to Domains and Tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 8342–8360.

[123] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval Augmented Language Model Pre-Training. In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119)*, Hal Daumé III and Aarti Singh (Eds.). PMLR, 3929–3938.

[124] Abner Guzman-Rivera, Dhruv Batra, and Pushmeet Kohli. 2012. Multiple choice learning: Learning to produce multiple structured outputs. *Advances in neural information processing systems* 25 (2012).

[125] Michael Alexander Kirkwood Halliday and Ruqaiya Hasan. 2014. *Cohesion in english*. Routledge.

[126] Shuang Hao, Wenfeng Han, Tao Jiang, Yiping Li, Haonan Wu, Chunlin Zhong, Zhangjun Zhou, and He Tang. 2024. Synthetic Data in AI: Challenges, Applications, and Ethical Implications. *arXiv preprint arXiv:2401.01629* (2024).

[127] Hamza Harkous, Isabel Groves, and Amir Saffari. 2020. Have Your Text and Use It Too! End-to-End Neural Dataset-to-Text Generation with Semantic Fidelity. In *Proceedings of the 28th International Conference on Computational Linguistics*. 2410–2424.

[128] Tatsunori B Hashimoto, Hugo Zhang, and Percy Liang. 2019. Unifying Human and Statistical Evaluation for Natural Language Generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 1689–1701.

[129] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.
[130] Shizhu He, Cao Liu, Kang Liu, and Jun Zhao. 2017. Generating natural answers by incorporating copying and retrieving mechanisms in sequence-to-sequence learning. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 199–208.

[131] Xiaodong He and Li Deng. 2017. Deep Learning for Image-to-Text Generation: A Technical Overview. IEEE Signal Processing Magazine 34, 6 (2017), 109–116. https://doi.org/10.1109/MSP.2017.2741510

[132] Ralf Herbrich, Tom Minka, and Thore Graepel. 2006. TrueSkill™: a Bayesian skill rating system. Advances in neural information processing systems 19 (2006).

[133] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. Neural computation 9, 8 (1997), 1735–1780.

[134] Laurence Horn. 1984. Toward a new taxonomy for pragmatic inference: Q-based and R-based implicature. Meaning, form, and use in context: Linguistic applications 11 (1984), 42.

[135] David M Howcroft, Anja Belz, Miruna-Adriana Clinciu, Dimitra Gkatzia, Sadid A Hasan, Saah Mahamood, Simon Mille, Emiel Van Miltenburg, Sashank Santhanam, and Verena Rieser. 2020. Twenty years of confusion in human evaluation: NLG needs evaluation sheets and standardised definitions. In Proceedings of the 13th International Conference on Natural Language Generation. 169–182.

[136] David M Howcroft, Dietrich Klakow, and Vera Demberg. 2017. The Extended SPaRKy Restaurant Corpus: Designing a Corpus with Variable Information Density. In INTERSPEECH. 3757–3761.

[137] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In International conference on machine learning. PMLR, 1587–1596.

[138] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition. 4700–4708.

[139] Gloriana Jagfeld, Sabrina Jenne, and Ngoc Thang Vu. 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. 221–232.

[140] Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumble-softmax. In International Conference on Learning Representations (ICLR 2017).

[141] Robert L Johnson and Grant B Morgan. 2016. Survey scales: A guide to development, analysis, and reporting. Guilford Publications.

[142] Shailza Jolly, Zi Xuan Zhang, Andreas Dengel, and Lili Mou. 2022. Search and learn: improving semantic coverage for data-to-text generation. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36. 10858–10866.

[143] Anjuli Kannan and Oriol Vinyals. 2016. Adversarial Evaluation of Dialogue Models. (2016).

[144] Nikiforos Karamanis, Massimo Poesio, Chris Mellish, and Jon Oberlander. 2004. Evaluating centering-based metrics of coherence. In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04). 391–398.

[145] Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The State and Fate of Linguistic Diversity and Inclusion in the NLP World. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 6282–6293.
[243] Jaclyn Peiser. 2019. The Rise of the Robot Reporter. The New York Times (Feb 2019).
[244] Laura Perez-Beltrachini and Mirella Lapata. 2018. Bootstrapping Generators from Noisy Data. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). 1516–1527.
[245] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). Association for Computational Linguistics, 2227–2237. https://doi.org/10.18653/v1/N18-1202
[246] Marian Petre. 1995. Why looking isn’t always seeing: readership skills and graphical programming. Commun. ACM 38, 6 (1995), 33–44.
[247] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language Models as Knowledge Bases?. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). 2463–2473.
[248] Joëlle Pineau, Philippe Vincent-Lamarre, Koustuv Sinha, Vincent Larivière, Alina Beygelzimer, Florence d’Alché Buc, Emily Fox, and Hugo Larochelle. 2021. Improving reproducibility in machine learning research (a report from the neurips 2019 reproducibility program). The Journal of Machine Learning Research 22, 1 (2021), 7459–7478.
[249] François Portet, Ehud Reiter, Albert Gatt, Jim Hunter, Somayajulu Sripada, Yvonne Freer, and Cindy Sykes. 2009. Automatic generation of textual summaries from neonatal intensive care data. Artificial Intelligence 173, 7-8 (2009), 789–816.
[250] Matt Post. 2018. A Call for Clarity in Reporting BLEU Scores. In Proceedings of the Third Conference on Machine Translation: Research Papers. 186–191.
[251] Ellen F Prince. 1981. Towards a taxonomy of given-new information. Radical pragmatics (1981).
[252] Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-text generation with content selection and planning. In Proceedings of the AAAI conference on artificial intelligence, Vol. 33. 6908–6915.
[253] Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-text Generation with Entity Modeling. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2023–2035.
[254] Ratish Puduppully, Yao Fu, and Mirella Lapata. 2022. Data-to-text Generation with Variational Sequential Planning. arXiv preprint arXiv:2202.13756 (2022).
[255] Ratish Puduppully and Mirella Lapata. 2021. Data-to-text Generation with Macro Planning. Transactions of the Association for Computational Linguistics 9 (2021), 510–527.
[256] Yevgeniy Puzikov and Iryna Gurevych. 2018. E2e nlg challenge: Neural models vs. templates. In Proceedings of the 11th International Conference on Natural Language Generation. 463–471.
[257] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. OpenAI blog 1, 8 (2019), 9.
[258] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv preprint arXiv:1910.10683 (2019).
[259] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-To-Text Transformer. Journal of Machine Learning Research 21 (2020), 1–67.
[260] Juan Ramos et al. 2003. Using tf-idf to determine word relevance in document queries. In Proceedings of the first instructional conference on machine learning, Vol. 242. Citeseer, 29–48.
[261] Clément Rebuffel, Laure Soulier, Geoffrey Scoutheeten, and Patrick Gallinari. 2020. A hierarchical model for data-to-text generation. In European Conference on Information Retrieval. Springer, 65–80.
[262] Clément Rebuffel, Laure Soulier, Geoffrey Scoutheeten, and Patrick Gallinari. 2020. Parenting via model-agnostic reinforcement learning to correct pathological behaviors in data-to-text generation. arXiv preprint arXiv:2010.10866 (2020).
[263] Lena Reed, Shereen Oraby, and Marilyn Walker. 2018. Can Neural Generators for Dialogue Learn Sentence Planning and Discourse Structuring?. In Proceedings of the 11th International Conference on Natural Language Generation. 284–295.
[264] Marek Rei. 2017. Semi-supervised Multitask Learning for Sequence Labeling. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2121–2130.
[265] Nils Reimers and Iryna Gurevych. 2017. Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. 338–348.
[266] Ehud Reiter. 2007. An architecture for data-to-text systems. In proceedings of the eleventh European workshop on natural language generation (ENLG 07). 97–104.

[267] Ehud Reiter. 2017. You need to understand your corpora! The Weathergov example. https://ehudreiter.com/2017/05/09/weathergov/

[268] Ehud Reiter. 2018. A structured review of the validity of BLEU. Computational Linguistics 44, 3 (2018), 393–401.

[269] Ehud Reiter and Anja Belz. 2009. An investigation into the validity of some metrics for automatically evaluating natural language generation systems. Computational Linguistics 35, 4 (2009), 529–558.

[270] Ehud Reiter and Robert Dale. 1997. Building applied natural language generation systems. Natural Language Engineering 3, 1 (1997), 57–87.

[271] Ehud Reiter, Somayajulu Sripada, Jim Hunter, Jin Yu, and Ian Davy. 2005. Choosing words in computer-generated weather forecasts. In Proceedings of the IEEE conference on computer vision and pattern recognition. 7008–7024.

[272] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In Proceedings of the IEEE conference on computer vision and pattern recognition. 211–227.

[273] Leonardo FR Ribeiro, Claire Gardent, and Iryna Gurevych. 2019. Enhancing AMR-To-Text Generation with Dual Graph Representations. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). 3183–3194.

[274] Leonardo FR Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. 2021. Investigating Pretrained Language Models for Graph-To-Text Generation. In Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI. 211–227.

[275] Leonardo FR Ribeiro, Yue Zhang, Claire Gardent, and Iryna Gurevych. 2020. Modeling Global and Local Node Contexts for Text Generation from Knowledge Graphs. Transactions of the Association for Computational Linguistics 8 (2020), 589–604.

[276] Markus Ring, Daniel Schlör, Dieter Landes, and Andreas Hotho. 2019. Flow-based network traffic generation using generative adversarial networks. Computers & Security 82 (2019), 156–172.

[277] Marco Roberti, Giovanni Bonetta, Rossella Cancelliere, and Patrick Gallinari. 2019. Copy mechanism and tailored training for character-based data-to-text generation. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, 648–664.

[278] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. Foundations and Trends in Information Retrieval 3, 4 (2009), 333–389.

[279] Jacques Pierre Robin. 1995. Revision-based generation of natural language summaries providing historical background: corpus-based analysis, design, implementation and evaluation. Ph. D. Dissertation. Columbia University.

[280] Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. 2018. Object Hallucination in Image Captioning. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. 4035–4045.

[281] Andreas Rücklé, Steffen Eger, Maxime Peyrard, and Iryna Gurevych. 2018. Concatenated power mean word embeddings as universal cross-lingual sentence representations. arXiv preprint arXiv:1803.01400 (2018).

[282] Sashank Santhanam and Samira Shaikh. 2019. A survey of natural language generation techniques with a focus on dialogue systems: present, past and future directions. arXiv preprint arXiv:1906.00500 (2019).

[283] Jost Schatzmann, Blaise Thomson, Karl Weillhammer, Hui Ye, and Steve Young. 2009. Agenda-based user simulation. In Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers. 149–152.

[284] Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language Models Can Teach Themselves to Use Tools. ArXiv abs/2302.04761 (2023).

[285] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In European semantic web conference. Springer, 593–607.

[286] Donia Scott and Johanna Moore. 2007. An NLG evaluation competition? eight reasons to be cautious.

[287] Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, 1073–1083. https://doi.org/10.18653/v1/P17-1099.

[288] Abigail See, Aneesh Pappu, Rohun Saxena, Akhila Yerukola, and Christopher D Manning. 2019. DoMassively Pretrained Language Models Make Better Storytellers?. In Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL). 843–861.

[289] Edward Segel and Jeffrey Heer. 2010. Narrative visualization: Telling stories with data. IEEE transactions on visualization and computer graphics 16, 6 (2010), 1139–1148.
[290] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving Neural Machine Translation Models with Monolingual Data. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 86–96.

[291] Iulian Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 31.

[292] Lei Sha, Lili Mou, Tianyu Liu, Pascal Poupart, Sujuan Li, Baobao Chang, and Zhifang Sui. 2018. Order-planning neural text generation from structured data. In Thirty-Second AAAI Conference on Artificial Intelligence.

[293] Zihong Shao, Minlie Huang, Jiangtao Wen, Wenfei Xu, and Xiaoyan Zhu. 2019. Long and Diverse Text Generation with Planning-based Hierarchical Variational Model. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). 3257–3268.

[294] Mandar Sharma, John S Brownstein, and Naren Ramakrishnan. 2021. T 3: Domain-Agnostic Neural Time-series Narration. In 2021 IEEE International Conference on Data Mining (ICDM). IEEE. 1324–1329.

[295] Mandar Sharma, Nikhil Muralidhar, and Naren Ramakrishnan. 2022. Overcoming Barriers to Skill Injection in Language Modeling: Case Study in Arithmetic. arXiv preprint arXiv:2211.02098 (2022).

[296] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-Attention with Relative Position Representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). 464–468.

[297] Sheng Shen, Daniel Fried, Jacob Andreas, and Dan Klein. 2019. Pragmatically Informative Text Generation. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). 4060–4067.

[298] Stuart Shieber, Gertjan Van Noord, Fernando CN Pereira, and Robert C Moore. 1990. Semantic-head-driven generation. Computational Linguistics (1990).

[299] Anastasia Shimorina and Claire Gardent. 2018. Handling rare items in data-to-text generation. In Proceedings of the First Workshop on Human Evaluation of NLP Systems (HumEval). 54–75.

[300] Anastasia Shimorina and Anja Belz. 2022. The Human Evaluation Datasheet: A Template for Recording Details of Human Evaluation Experiments in NLP. In Proceedings of the 2nd Workshop on Human Evaluation of NLP Systems (HumEval). 54–75.

[301] Advaith Siddharthan and Napoleon Katsos. 2012. Offline sentence processing measures for testing readability with users. In Proceedings of the First Workshop on Predicting and Improving Text Readability for target reader populations. 17–24.

[302] Koustuv Sinha, Joelle Pineau, Jessica Forde, Rosemary Nan Ke, and Hugo Larochelle. 2020. NeurIPS 2019 reproducibility challenge. ReScience C 6, 2 (2020), 11.

[303] Charese Smiley, Vassilis Plachouras, Frank Schilder, Hiroko Bretz, Jochen L Leidner, and Dezhao Song. 2016. When to plummet and when to soar: Corpus based verb selection for natural language generation. In Proceedings of the 9th International Natural Language Generation conference. 36–39.

[304] Richard Socher, Jeffrey Pennington, Eric H Huang, Andrew Y Ng, and Christopher D Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In Proceedings of the 2011 conference on empirical methods in natural language processing. 151–161.

[305] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. 2015. Learning structured output representation using deep conditional generative models. Advances in neural information processing systems 28 (2015).

[306] Linfeng Song, Ante Wang, Jinsong Su, Yue Zhang, Kun Xu, Yubin Ge, and Dong Yu. 2020. Structural Information Preserving for Graph-to-Text Generation. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 7987–7998.

[307] Amanda Stent, Matthew Marge, and Mohit Singhai. 2005. Evaluating evaluation methods for generation in the presence of variation. In international conference on intelligent text processing and computational linguistics. Springer, 341–351.

[308] Amanda Stent, Rashmi Prasad, and Marilyn Walker. 2004. Trainable sentence planning for complex information presentations in spoken dialog systems. In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04). 79–86.

[309] Milan Straka and Jana Straková. 2017. Tokenizing, pos tagging, lemmatizing and parsing ud 2.0 with udpipe. In Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. 88–99.

[310] Shang-Yu Su, Kai-Ling Lo, Yi-Ting Yeh, and Yun-Nung Chen. 2018. Natural Language Generation by Hierarchical Decoding with Linguistic Patterns. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). 61–66.
[334] Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. 2019. Do NLP Models Know Numbers? Probing Numeracy in Embeddings. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). 5307–5315.

[335] Hongmin Wang. 2019. Revisiting Challenges in Data-to-Text Generation with Fact Grounding. In Proceedings of the 12th International Conference on Natural Language Generation. 311–322.

[336] Jun Wang and Klaus Mueller. 2015. The visual causality analyst: An interactive interface for causal reasoning. IEEE transactions on visualization and computer graphics 22, 1 (2015), 230–239.

[337] Peng Wang, Junyang Lin, An Yang, Chang Zhou, Yichang Zhang, Jingren Zhou, and Hongxia Yang. 2021. Sketch and Refine: Towards Faithful and Informative Table-to-Text Generation. In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. 4831–4843.

[338] Qingyun Wang, Xiaoman Pan, Lifu Huang, Boliang Zhang, Zhiying Jiang, Heng Ji, and Kevin Knight. 2018. Describing a Knowledge Base. In Proceedings of the 11th International Conference on Natural Language Generation. 10–21.

[339] Zhenyi Wang, Xiaoyang Wang, Bang An, Dong Yu, and Changyou Chen. 2020. Towards Faithful Neural Table-to-Text Generation with Content-Matching Constraints. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 1072–1086.

[340] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed H Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-Thought Prompting Elicits Elicits Reasoning in Large Language Models. In Advances in Neural Information Processing Systems.

[341] Sean Welleck, Illia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2019. Neural Text Generation With Unlikelihood Training. In International Conference on Learning Representations.

[342] Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Lina Maria Rojas-Barahona, Pei-Hao Su, David Vandyke, and Steve Young. 2016. Multi-domain Neural Network Language Generation for Spoken Dialogue Systems. In HLT-NAACL.

[343] Tsung-Hsien Wen, Milica Gasic, Nikola Mrksič, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically Conditioned LSTM-based Natural Language Generation for Spoken Dialogue Systems. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. 1711–1721.

[344] Lilian Weng. 2021. Controllable Neural Text Generation. lilianweng.github.io (2021). https://lilianweng.github.io/posts/2021-01-02-controllable-text-generation/

[345] Ronald J Williams and Jing Peng. 1991. Function optimization using connectionist reinforcement learning algorithms. Connection Science 3, 3 (1991), 241–268.

[346] Ronald J Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. Neural computation 1, 2 (1989), 270–280.

[347] Sam Wiseman, Stuart M Shieber, and Alexander M Rush. 2017. Challenges in Data-to-Document Generation. In EMNLP.

[348] Sam Wiseman, Stuart M Shieber, and Alexander M Rush. 2018. Learning Neural Templates for Text Generation. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. 3174–3187.

[349] Stephen Wolfram. 2023. Wolfram/Alpha as the Way to Bring Computational Knowledge Superpowers to ChatGPT. https://writings.stephenwolfram.com/2023/01/wolframalpha-as-the-way-to-bring-computational-knowledge-superpowers-to-chatgpt/

[350] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. (2016).

[351] Jiannan Xiang, Zhengzhong Liu, Yucheng Zhou, Eric Xing, and Zhiting Hu. 2022. ASDOT: Any-Shot Data-to-Text Generation with Pretrained Language Models. In Findings of the Association for Computational Linguistics: EMNLP 2022. Association for Computational Linguistics, 1886–1899.

[352] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In International conference on machine learning, PMLR, 2048–2057.

[353] Shengzhe Xu, Manish Marwah, Martin Arlitt, and Naren Ramakrishnan. 2021. Stan: Synthetic network traffic generation with generative neural models. In Deployable Machine Learning for Security Defense: Second International Workshop, MLHat 2021, Virtual Event, August 15, 2021, Proceedings 2. Springer, 3–29.

[354] Yang Yang, Juan Cao, Yujun Wen, and Pengzhou Zhang. 2021. Table to text generation with accurate content copying. Scientific reports 11, 1 (2021), 1–12.

[355] Zichao Yang, Phil Blunsom, Chris Dyer, and Wang Ling. 2017. Reference-Aware Language Models. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. 1850–1859.

[356] Rong Ye, Wenzhian Shi, Hao Zhou, Zhongyu Wei, and Lei Li. 2020. Variational Template Machine for Data-to-Text Generation. (2020).
[357] Steve Young, Milica Gašić, Simon Keizer, François Mairette, Jost Schatzmann, Blaise Thomson, and Kai Yu. 2010. The hidden information state model: A practical framework for POMDP-based spoken dialogue management. *Computer Speech & Language* 24, 2 (2010), 150–174.

[358] Włodek Zadrozny and Karen Jensen. 1991. Semantics of paragraphs. *Computational Linguistics* 17, 2 (1991), 171–210.

[359] Biao Zhang, Deyi Xiong, Jinsong Su, and Hong Duan. 2017. A context-aware recurrent encoder for neural machine translation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 25, 12 (2017), 2424–2432.

[360] Biao Zhang, Jing Yang, Qian Lin, and Jinsong Su. 2018. Attention regularized sequence-to-sequence learning for e2e nlg challenge. (2018).

[361] Shuo Zhang, Zhuyun Dai, Krisztian Balog, and Jamie Callan. 2020. Summarizing and exploring tabular data in conversational search. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1537–1540.

[362] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. BERTScore: Evaluating Text Generation with BERT. In *International Conference on Learning Representations*.

[363] Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and William B Dolan. 2020. DIALOGPT: Large-Scale Generative Pre-training for Conversational Response Generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. 270–278.

[364] Chao Zhao, Marilyn Walker, and Snigdha Chaturvedi. 2020. Bridging the structural gap between encoding and decoding for data-to-text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 2481–2491.

[365] Jianyu Zhao, Zhiqiang Zhan, Tong Li, Rang Li, Changjian Hu, Siyun Wang, and Yang Zhang. 2021. Generative adversarial network for Table-to-Text generation. *Neurocomputing* 452 (2021), 28–36.

[366] Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M Meyer, and Steffen Eger. 2019. MoverScore: Text Generation Evaluating with Contextualized Embeddings and Earth Mover Distance. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 563–578.

[367] Jie Zhu, Junhui Li, Muhua Zhu, Longhua Qian, Min Zhang, and Guodong Zhou. 2019. Modeling Graph Structure in Transformer for Better AMR-to-Text Generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 5459–5468.

[368] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. 2008. Maximum entropy inverse reinforcement learning. In *Aaai*, Vol. 8. Chicago, IL, USA. 1433–1438.