Challenges and Opportunities in Low-Resource Natural Language Generation

Anonymous ACL submission

Abstract

Many domains and tasks in natural language generation (NLG) are inherently ‘low-resource’, where training data and linguistic analyses are scarce. This poses a particular challenge to researchers and system developers in the era of machine-learning-driven NLG. Here, we initially discuss the challenges researchers and developers often encounter when dealing with low-resource settings in NLG. We argue that it is unsustainable to collect large aligned datasets or build large language models from scratch for every possible domain due to cost, labour, and time constraints, so researching and developing approaches for low-resource settings is vital. We then present a survey of current approaches to low-resource NLG, followed by a list of proposed solutions and promising avenues for future work in NLG for low-resource settings.

1 Introduction

Natural Language Generation (NLG) is the process of generating text from structured or unstructured data and has recently renewed attention due to the emergence of large pre-trained language models (e.g. Brown et al., 2020) that promise to generate output that is more natural, variable, and adaptable to new domains as compared to rule-based approaches. However, the development of robust, controllable, and usable NLG systems depends heavily on the availability of large, high-quality, labelled datasets that are appropriate for the task at hand. Unfortunately, for most domains such data is unavailable, is very small in size, or does not offer wide coverage of the domain, probably with the exception of weather, restaurant, and sports domains. Even in the aforementioned domains, data is fairly small compared to low-resource tasks/languages in other areas of natural language processing (NLP), such as in machine translation.

In this work, we focus on NLG as ‘data-to-text generation’, where data can take the form of raw sensor data, tabular information, database records, slot-value pairs, dialogue acts, or syntactic or semantic structures. We focus on ways in which NLG is low-resource, that is, on the limitations to the creation of new NLG systems due to a lack of data, analyses, or tools for a target domain or language (cf. Table 1). In this sense, data are corpora which include paired input representations and texts, i.e. associating inputs with outputs. Analyses are linguistic analyses relevant to a domain of application, including but not limited to grammars of a target language or conversation analyses of a target domain. Tools can then be automated means of analysing linguistic data (e.g. parsers, part-of-speech taggers, etc.), secondary resources based on primary data (e.g. word embeddings), or software libraries for different NLG (sub)tasks.

Given that a lack of data is the most common obstacle for NLG, it is no surprise that most of work on low-resource NLG has focused on data augmentation (Chan et al., 2021; Oraby et al., 2019). Data augmentation approaches for NLG require careful design as the augmented text should be grammat-
2 Data: Corpora for NLG

Corpora are very important for the development of both data-driven and knowledge-driven NLG systems. We focus here on NLG corpora for generating full utterances of at least one complete sentence which include input meaning representations (MRs) (ranging from raw sensor data to morphologically specified syntax trees) and corresponding texts, in situations where existing resources are limited. Currently, the domains with the most data-to-text corpora available are restaurant descriptions, weather forecasts, and sports reporting. Corpora for the widely studied restaurant domain range in size from 400 utterances (BAGEL) to 50k utterances (E2E Challenge corpus), some being based on hand-crafted systems while the majority are crowdsourced 1. We now highlight prominent strategies for building NLG corpora; however, even with a simple MR format (e.g. CUED slot value pairs), collecting high-quality parallel MR-text corpora is expensive, meaning that most datasets built using these methods remain small.

1The ACL SIG on generation maintains a list: https://aclweb.org/aclwiki/Data_sets_for_NLG

‘Found’ NLG Corpora Sometimes data-to-text corpora can be easily adapted from existing sources of semantic and textual data. For instance, Belz and Kow (2010) assembled a parallel corpus of sets of facts (from hobbyists) and corresponding texts (from Wikipedia) about British hills. Similarly, the WikiBio dataset (Lebret et al., 2016) pairs the first sentence of each article in the WikiProject Biography dataset with the facts reported in that article’s ‘infobox’. GenWiki extends these approaches even further, aiming to provide an automatically aligned corpus of texts from Wikipedia paired with graphs from DBPedia identified based on overlapping entities for more than one million texts (Jin et al., 2020). Apart from Wikipedia, researchers have collected datasets from other online resources that contain both data (in metadata) and (somewhat) aligned text, e.g. (Liang et al., 2009; Barzilay and Lapata, 2005). Note that these approaches rely on the fact that others, such as domain experts, have already chosen to create a semantic or tabular representation of important details, therefore this method of building new datasets is not generalisable. However, when such datasets are available for a relevant domain, they have the potential to provide data at a scale not easily achieved otherwise.

Creating meaning annotations Some research has annotated existing data-to-text corpora with discourse structures, such as Balakrishnan et al. (2019) who semi-automatically added discourse structures to the E2E Challenge corpus and Stevens-Guillem et al. (2020) who leveraged the rule-based Methodiust system (Isard, 2016) to create a discourse-annotated corpus to test neural methods of generation. Based on the automatically derived Methodiust Corpus, Maskharashvili et al. (2021) observe that “discourse relations are enormously helpful when the dataset for the domain is limited”.

Other work has sought to address the issue of content selection for dataset creation, independent of the actual text to be associated with each meaning representation (see also (Gkatzia, 2016)). For example, Perez-Beltrachini et al. (2016) leverage DBPedia to construct trees of semantic triples based on their frequency and relationship to one another in a large ontology, with the goal of selecting content which forms a natural unit that can be later associated with a human-written text.

Eliciting texts for given meanings Early datasets typically relied on domain experts or NLG
researchers directly, but most recent work uses crowdsourcing to quickly collect texts from a variety of speakers (e.g. Mairesse et al., 2010; Wen et al., 2015, 2016). Where early work tended to use prompts similar to a set of slot-value pairs, later work observed that such prompts encouraged the use of particular words & phrases, reducing textual diversity, and found that using images (Novikova et al., 2016) or full sentences (Howcroft et al., 2017) as prompts resulted in better quality. Recent work has incorporated quality estimates and text suggestions to fill gaps in the existing dataset (Chang et al., 2020). While these methods can quickly provide varied data, it remains impractical to create large, targeted datasets for all domains.

3 Analyses & Tools

While the primary data for NLG are parallel corpora, a variety of other resources can facilitate development. Language models can approximate fluency measures and can be used to produce texts using sampling. Linguistic analyses provide insight into what makes a text well-formed and assist in the design of rule-based systems and architectures for ML-based approaches. Finally, tools & resources such as parsers, part-of-speech taggers, semantic role labellers, ontologies, morphological analysers, and word embeddings can help to decompose NLG subtasks and make the problem more approachable.

3.1 (Large) Language Models

Statistical language models have been used since the late 1990s to help rank potential NLG outputs (Langkilde and Knight, 1998; Knight and Hatzivasiloglou, 1995). With the rise of large pre-trained language models (e.g. BERT, GPT: Devlin, 2018; Brown et al., 2020), there has been renewed interest in sampling-based approaches to generation, where an LM trained exclusively on text (i.e., without parallel MRs) generates a continuation for an initial string. While the challenge of making sampling-based NLG more controlled is an active area of research, these tools continue to be helpful to rank texts with likelihood as a fluency approximation.

Unfortunately, good language models require large collections of text in the target language in order to perform well. Many languages have no general corpora or LMs available which can be repurposed for NLG. Indeed, two of the largest multilingual pretrained LMs cover only about 100 languages (i.e. mBERT (Devlin, 2018) & mT5 (Xue et al., 2021)). Even when such resources are available, they are often based on legal, journalistic, or governmental language rather than everyday language, resulting in a genre mismatch making them poor off-the-shelf models for many applications.

3.2 Analysis

With enough data, we can hope that a powerful ML architecture will detect the patterns necessary to produce good texts. However, when corpora are not large enough for this, descriptive and automated linguistic analyses can help. Researchers can use linguistic documentation to develop their system, consulting formal grammars & lexica to understand the kinds of constructions possible in a target language. Researchers & developers can also partner with speakers of their target language to ensure that the system serves community needs while working together to understand the language they are generating (Hirmer et al., 2021).

Generally speaking, it is easiest to leverage these linguistic resources when developing a rule-based NLG system, where observations can be encoded explicitly. For example, three grammars of Rapanui have been published in English in the past 110 years (Churchill, 1912; Du Feu, 2012; Kieviet, 2017), giving insight into word order, morphology, and agreement phenomena. Such features can then be input in grammar engineering tools like the Lingo matrix grammar construction toolkit (Bender et al., 2002, 2010), to provide a starting point for building a rule-based NLG system.

Both rule-based systems and end-to-end ML approaches benefit from tools for automated linguistic analysis, such as lemmatisers, part-of-speech taggers, parsers, and semantic role labellers. To learn to map input MRs to output texts, a system or developer must recognise useful generalisations and abstractions. For example, part-of-speech tags and constituency parses can provide information about the order in which words of a given class should appear, while dependency parses and semantic role labels can relate individual words to each other. Similarly, normalization diminishes the impact of noise, lemmatization helps associate word ‘stems’ with meanings, morphological analysers/realisers help with word forms, parsers help get words in the right order and relation to one another, semantic role labellers/NLU systems help you associate

\[\text{https://ctrlgenworkshop.github.io/}\]

\[\text{https://matrix.ling.washington.edu}\]
chunks of form and meaning directly.

All of these individual analyses together help decompose the task. This is not a new idea in NLG, since data-to-text systems have used modular architectures for decades (Reiter, 2007). Since this is an important promising direction, we discuss pipelines and problem structuring in detail in Section 4.5.

3.3 NLG Libraries

Similarly to general NLP tools, NLG-specific tooling shows variable availability. While SimpleNLG has been adapted to 7 other languages4 (English: Gatt and Reiter, 2009), this library still requires significant effort to develop supporting components. Other available tools for developing NLG systems include FUF/SURGE5 (Elhadad and Robin, 1997, 1996) for surface realisation and OpenCCG6 (White, 2006) for generation from hybrid logic dependency semantics, which can be used to represent meaning at deeper and shallower levels alike, with combinatory categorial grammar.

A number of neural NLG models are also publicly available, such as Wen et al.’s (2015) SC-LSTM7 and Dusek & Jurčíček’s (2016) TGen8. While TGen has been widely used as a baseline for end-to-end NLG tasks, these systems generally represent the outcome of a particular research project rather than being intended to be used as a platform for developing future NLG systems.

4 Mitigation strategies

So far we have discussed the data, analyses, & tools that are often missing in low-resource settings. We now turn to mitigation strategies for dealing with a lack of resources, namely data augmentation, machine learning methods, and rule-based methods.

4.1 Data Augmentation

Data augmentation (DA) describes a family of approaches that aim to increase the number of training examples automatically, without manual data collection. Despite their popularity and demonstrated efficiency in other areas such as computer vision (Shorten and Khosgoftaar, 2019) and NLP9, this area is still relatively under-explored in NLG, partly due to unique challenges. This section reviews existing approaches to DA for text generation; for DA approaches to NLG in general we refer the reader to the survey by Feng et al. (2021).

DA methods promise to enrich current datasets, potentially increasing the linguistic diversity of the dataset (e.g., by enhancing stylistic traits as proposed by Oraby et al. (2018)). In order to train NLG models, we require text that is not only grammatically correct but also coherent and appropriate for the task at hand. Therefore, straightforward approaches explored in computer vision, such as cropping and rotation are not appropriate. More elaborate approaches such as back-translation & paraphrasing have been used in other areas of NLP, such as MT, which are promising also for NLG. Here, however, we only review methods specifically developed and applied in NLG settings.

There are two predominant DA methods used in NLG: (1) generation of new examples with pre-trained LMs; and (2) generation of new examples with statistical or rule-based NLG systems.

Chang et al. (2021a) propose a simple approach to generating new samples for NLG using a pre-trained LM. First, they create an unannotated dataset with unlabeled MR instances by randomly selecting MRs from a pre-existing expert-annotated dataset and populating them with existing values. This dataset is then automatically annotated with noisy text labels generated by a pre-trained model, fine-tuned on joint MR and text conditioned on samples from the augmented MR set.

Rule-based systems have long been used to generate new examples for developing or evaluating other NLG methods. Belz (2008) created an automatically generated version of SUMTIME meteo (Sripada et al., 2008), an expert annotated data-to-text dataset in the weather domain. This automatically generated version has been used to explore statistical NLG (Angeli et al., 2010, inter alia). Similarly, Oraby et al. (2018) utilised a statistical natural language generator to create a corpus of stylistic texts used to train seq2seq neural NLG models.

Distant supervision can also be used to create new NLG corpora. For example, Agarwal et al. (2021) use distant supervision to verbalize knowledge graph subgraphs centered on entities, producing ‘silver standard’ data. The authors were able to create a large, not cleaned dataset for training and only cleaned the subset used for fine-tuning.

4

---

4https://github.com/Simplenlg/simplenlg

5https://www.cs.bgu.ac.il/~elhadad/install-fuf.html

6https://github.com/OpenCCG/openccg

7https://github.com/shawnwun/RNNLG

8https://github.com/UFAL-DSG/tgen

9https://github.com/GEM-benchmark/NL-Augmenter
<table>
<thead>
<tr>
<th>Mitigation Strategy</th>
<th>Requirements</th>
<th>Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Augmentation</td>
<td>labelled data, rules</td>
<td>additional labelled data</td>
</tr>
<tr>
<td>Pre-training/Fine-tuning</td>
<td>existing LM or high-resource data, low-resource data</td>
<td>knowledge transfer to new domain</td>
</tr>
<tr>
<td>Zero-shot</td>
<td>existing LM</td>
<td>adapting to new scenarios with no labelled data</td>
</tr>
<tr>
<td>Few-shot</td>
<td>existing LM, few training examples</td>
<td>adapting to new scenarios with few labelled data</td>
</tr>
<tr>
<td>Prompt-based</td>
<td>existing LM, prompts</td>
<td>weight initialisation for better training</td>
</tr>
<tr>
<td>Meta-learning</td>
<td>high-resource data, low-resource dataset</td>
<td>better learning through interaction</td>
</tr>
<tr>
<td>Feedback-based</td>
<td>small domain dataset</td>
<td>learning more general representations, multiple tasks</td>
</tr>
<tr>
<td>Multi-task</td>
<td>multiple auxiliary tasks</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: The table summarises the approaches used to mitigate the lack of training data for NLG systems.

### 4.2 Transfer Learning

Transfer learning uses the knowledge gained from a previous task (in a high-resource setting) to improve model performance in a lower-resource setting related task (Torrey and Shavlik, 2009). Typically, a model is trained with data from one or more high-resource domains/languages and then the model’s weights are used to initialise the model to be trained in the low-resource setting through a process called fine-tuning. Data availability for the target language has a substantial impact on the approaches available to researchers, as only very limited digitized data is available for most languages. For example, Joshi et al. (2020) found that only seven languages (of the world’s approximately 7000 languages) qualify as truly high-resource.

One major factor in data availability is the cost of technology relative to the basic income of countries where a language is spoken (Ahia et al., 2021). Given the high cost of collecting & annotating data, this observation is not surprising. Recent approaches to text generation have emphasised the use of large-pretrained LMs (see Section 3.1) and task-specific fine-tuning in order to transfer general language statistics to a particular task. Consider for instance GPT3 (Brown et al., 2020), trained on 45TB of text data in English. Training such a model for any language requires an amount of data and compute power unavailable in most regions.

Similarly, ‘few-shot’ and ‘zero-shot’ approaches aim to develop general language models which can be applied to new tasks with limited intervention (Zhao and Eskenazi, 2018). There are different approaches to transfer learning proposed for NLG based on different characteristics. Below is an attempt to group them (see also Table 2).

#### Pre-training & Fine-tuning

Transfer learning via fine-tuning typically requires the adaptation of a large pre-trained language model by updating and storing all of the parameters, resulting in a new language model for every task. One of the earliest works in this setting involved training a model from scratch in a related domain and then fine-tuning it in a new domain (Dethlefs, 2017). Kale and Roy (2020) propose transferring knowledge from a NMT model trained on English-Czech, which is then fine tuned for a data-to-text task in Czech. Pasricha et al. (2020) proposed a transfer learning approach which actually utilises one of the large language models (see Section 3.1) which is fine-tuned in the target task. In this setting, the data used for fine tuning is pre-pended with tags describing the part of speech as well as the type of the entity and are included in the vocabulary. Ribeiro et al. (2021) also show that pre-trained language models perform well in graph-based MR to text generation, even when the input is reduced to bags of nodes and edge labels. Most works in this area generate text in English. However, fine-tuning large models also works in other languages. For instance, Naeous et al. (2021) propose fine-tuning AraBERT (Antoun et al., 2020) for empathetic NLG in Arabic.

Although fine-tuning requires significantly less computational power and time as compared to training models from scratch, it can still pose a considerable deployment challenge as the magnitude of pre-trained models continues to increase from millions to billions of parameters. As such, other data- and computationally-efficient transfer learning approaches have been explored that try to minimize the number of parameters that are fine-tuned. Such methods include prompt-based, few- and zero-shot learning approaches which are discussed next.

#### Prompt-based, Few- and Zero-shot learning

To alleviate the need to update all parameters of a pre-trained model, researchers have explored prompt-based methods that keep the parameters of a model frozen and instead use prompts as part...
of the input (Liu et al., 2021) to perform downstream tasks without further training. Although no training is required, prompt-based methods require: 1) a prompting function that converts the input into some specific form; 2) template prompts, which can be created manually or automatically; 3) corresponding filled & answer prompts; and 4) answers. In many cases, prompt methods do not require any further training and providing the aforementioned prompting elements is enough for a model to perform zero-shot learning (as in (Dou and Peng, 2022)). In other settings, though, prompting can be used for further training/fine-tuning a model, when at least a small amount of data is available. For instance, Li and Liang (2021) proposed an approach to NLG that keeps pre-trained LM parameters fixed, while employing a task-specific prefix vector, which is tuned for the task at hand. Clive et al. (2021) extend this approach for controlled text generation. Prompt-learning is a very recent paradigm, so we expect to see more work in this promising area.

Similarly to prompting, zero- and few-shot learning aim to achieve learning with minimal training/new data instances. Transfer learning aims to "learn" transferable features that can be used in downstream tasks. In few-shot learning, there might be only a few examples to learn from (or zero in zero-shot). Ma et al. (2019) proposed decomposing table-to-text generation into content selection and surface realisation, so that each subtask can be trained with a smaller dataset than it would be needed for the end-to-end task. Chen et al. (2020) also proposed pre-training a model from scratch, although their paradigm employed distant supervision before fine-tuning the model to specific tasks. Finally, Chang et al. (2021b) focused on improving few-shot NLG by prioritising informative training instances to fine-tune the model.

4.3 Meta-learning

Meta learning can be thought of as the machine learning process of "learning how to learn" (Mi et al., 2019a). Meta-learning approaches are split into metric-learning (Koch, 2015), model-based (Andrychowicz et al., 2016), and optimisation based approaches (Finn et al., 2017). Most of recent meta-learning approaches to NLG follow an optimisation approach, the Model-Agnostic Meta-Learning (MAML) algorithm, originally proposed by Finn et al. (2017). MAML aims to make models achieve good generalisation performance by adapting quickly to a new task during training in low-resource settings, despite a low quantity of training data, by learning a better initialization of model parameters that facilitates fast adaptation to new low-resource scenarios. Mi et al. (2019b) explored different adaptation settings based on domain similarity and showed that "nearer" domains can adapt better through a meta-learning setting, outperforming other optimisation methods such as multi-task learning. Meta-learning has also shown promising results in MT. Gu et al. (2018) compared MAML to transfer learning (Zoph et al., 2016) and showed that meta-learning leads to further improvements, despite training data for the low-resource language being limited to a significantly smaller dataset. As the corpus size of the low-resource language decreases, transfer learning approaches suffer significantly more than meta-learning approaches, which demonstrating the effectiveness of MAML for low-resource languages. However, as corpus size increases, the differences between the two approaches are much less significant. Exploring the trade-off between data size and learning paradigm (meta-learning vs transfer learning) is a promising direction for this line of work.

4.4 Feedback-based Learning

Low-resource settings were always a bottleneck for NLG. Earlier work in this area employed reinforcement learning dialogue systems NLG (Rieser and Lemon, 2011; Dethlefs and Cuayáhuitl, 2011) to overcome the issue of limited data. Low-resource settings can manifest themselves also in domains where large collections of unlabelled data are available without parallel inputs. For instance, in MT one might have access to large monolingual datasets in both source and target languages but not an aligned one. In this case, feedback-based methods can help, such as the dual-learning setup from Kim et al. (2019), presented as a two-agent communication game. In this setting, the first agent only understands language A, and the second agent only understand language B. The two agents communicate with through translation models and provide feedback on whether the translated message they received is a natural sentence in their own language. They then use this feedback to improve their individual models. Similarly, Shen et al. (2019) propose treating language production as a game between speakers and listeners, where listeners must
be able to reconstruct the intended meaning. Their models are trained to predict and avoid confusing outputs based on either reconstruction or distraction pragmatics (this is however not low-resource). Tran and Nguyen (2018) propose an adversarial training procedure for domain adaptation with two critics, which guide the generator to generate outputs similar to the sentences in the target domain, when limited amount of target domain data exist.

4.5 Task structure & rule-based approaches

While ‘fully end-to-end’ machine learning models are always enticing, careful thought about how to structure the task(s) can lead to significant improvements in outcomes. For example, in multitask learning a single model is trained on multiple tasks, allowing feedback from learning to perform well on one task to influence the others. Even without jointly learning to solve multiple tasks, decomposing generation into a sequence of stages in a pipeline can improve performance by simplifying what the model needs to learn.

**Multi-task learning** To our knowledge, there is no related work in multi-task learning for low-resource data-to-text generation. The closest work in this area jointly learns a semantically conditioned and unconditioned LMs for generation across multiple datasets (Zhu et al., 2019). For text-to-text generation, Magooda et al. (2021) showed that abstractive summarisation can benefit from being learnt in a multi-task framework, especially when combined with paraphrase learning. In addition to their target task, they train their model to perform extractive summarisation, concept & paraphrase detection, and language modelling. In a larger resource setting, Agarwal et al. (2020) adapt T5 (Raffel et al., 2020) to data-to-text generation for English and Russian while jointly learning text-to-data semantic parsing for both languages.

**Pipelines & Problem Structuring** Where in multitask learning the focus is on training a single neural network to perform multiple tasks, pipeline models use separate stages for each subproblem and do not require a single unified neural network (cf. Figure 1). This allows for specialisation, where solutions to subproblems can be refined independently. For example, a surface realisation module could be trained on multiple domains regardless of the handling required at the level of document or sentence planning. Of course, modularity comes with the risk of error propagation, as errors in an early component can break later ones.

Recent work has explored various decompositions of data-to-text generation based on the conventional NLG pipeline. For example, Howcroft et al. (2018) proposed a model to learn sentence planning rules for use with an off-the-shelf surface realiser while Moryossef et al. (2019) reversed this focus, using rule-based planning to handle content ordering & chunking and training a surface realiser.

Figure 1 sketches two fully neural NLG pipelines as examples. Castro Ferreira et al. (2019) decomposed generation into content ordering, content chunking, lexicalisation, referring expression generation (REG), & textual realisation and trained a neural model for each subtask, finding that pipelines improve text quality. Both Perez-Beltrachini and Lapata (2018) and Puduppully and Lapata (2021) created higher level pipelines which perform content selection (with the latter also performing content ordering and sentence chunking) before generating a text. Other work has used latent variable models to learn ordering and chunking constraints instead of focusing on explicit pipelines (Shen et al., 2020; Xu et al., 2021).

These models generally preserve the ‘fluency’ expected of modern neural network LMs and improve semantic fidelity. This is especially encour-
aging for low resource NLG where the relative pay-off for adding intermediate annotations to train pipeline models is likely to be higher. Such approaches are further enhanced by ‘delexicalisation’, e.g. replacing certain values with placeholders (Shimorina and Gardent, 2019) to limit spurious variation and facilitate training.

For low-resource domains in better resourced languages, useful resources for AMR-to-text generation\(^1\) and UD-to-text\(^2\) generation (Mille et al., 2018, 2019, 2020) could be used for the final stage of generation, simplifying system development and allowing researchers to focus on developing components for document and sentence planning.

Complementary to these efforts are a number of new corpora (van der Lee et al., 2020) and extensions of existing corpora (Castro Ferreira et al., 2020, 2021) with annotations for intermediate representations in an NLG pipeline.

### 5 Promising directions

It is not sustainable to collect large aligned datasets for every domain, so researching and developing approaches for low-resource settings is vital. Here we outline some promising directions in this area.

**Data Augmentation** Although it is clear that NLG could benefit from data augmentation, this area is still very under-explored. Approaches developed for other NLP tasks could be explored here whereas others won’t be applicable for structured prediction problems like NLG. The potential future directions can be split into three main categories: (a) Data augmentation through paraphrasing: The first family of DA approaches focuses on simple data augmentation, a slot is replaced with another slot span (the length range from one token to more) randomly: e.g. “cheapest” is replaced with “most expensive”; (b) using pre-trained models to distill knowledge; (c) back-translation to create resources in different languages.

**Prompt-based Learning** Prompt-based learning is a fairly novel direction for NLP (including NLG) in general and our understanding is quite limited to a few experiments. Work on this area can highlight limitations of current pre-trained models and lead to potential improvements as well as can reveal situations where large models are safe to use.

**Feedback-based approaches** Although there has been a lot of discussion in AI in general about machines learning the same way as humans, research that focuses on increasing a language model’s capabilities have only been limited to game-based settings or simulation (as in the case of Reinforcement Learning). More research in this area, might be useful to endow models with new capabilities as in the frameworks proposed by Gkatzia and Belvedere (2021) and Wang et al. (2016).

**Multitask learning & pipelines** Very few of the combinatorial possibilities for neural NLG pipelines have been explored to date, so it remains unclear which tasks are best solved in sequence versus jointly. One especially promising direction is to approach such pipelines in a multitask setting: when annotated data exists for subtasks, the model can receive feedback for individual tasks during training while passing along the penultimate layers of the network to each subsequent task, thus allowing later tasks in the pipeline to influence the hidden representations learnt in earlier tasks.

**When to use meta-learning and when to use transfer-learning?** There is some evidence from MT that transferring models between related languages increases performance Zoph et al. (2016). On the other hand, data size matters (Kocmi and Bojar, 2018). Previous work has also shown that once you start increasing the amount of data in the target domain, transfer learning achieves better results. However, it is unclear where the ‘sweet spot’ lies, and more research in this area is required.

### 6 Conclusion

All high-resource settings are alike; each low-resource setting is low-resource in its own way.\(^3\) In this paper, we have highlighted the ways in which natural language generation is influenced by the availability of data, analyses, and tools at both the language and the domain level and described a number of machine learning approaches which aim to learn an NLG system given different kinds of resource limitations. We look forward to seeing what new data-to-text resources are developed for various languages and domains and encourage the research community to identify supplemental information sources and leverage ML architectures and problem structures to learn from less data.

---

\(^1\)https://nert-nlp.github.io/AMR-Bibliography/
\(^2\)Universal Dependencies (de Marneffe et al., 2021)

\(^3\)Inspired by Leo Tolstoy’s Anna Karenina.
References


Jacob Devlin. 2018. Multilingual BERT README.


Young Jin Kim, Marcin Junczys-Dowmunt, Hany Hassan, Alham Fikri Aji, Kenneth Heafield, Roman Grundkiewicz, and Nikolay Bogoychev. 2019. From


