Design and Analysis of NLG systems

Language generation by humans is innate, so not much thought goes into the conception of the language generation process that happens in nature.
As a consequence, this reflects the general simplistic view for most onlookers on using machines to generate language.
In this section, we discuss the inputs, components, processes and architecture to gain a better appreciation on the foundations for modern N LG systems.

Inputs to Natural Language Generation

In general, language generation is often viewed as a goal or intent-driven communication. Goals of communication that are set to affect the target: opinion, knowledge, future behaviour, emotional state and interpersonal relationship. The task of language generation depends upon the complexity of processing that required to go from the input to the desired output.
In general terms, we can characterize the input to a language generation system as a four-tuple (k,c,u,d) where

k is the KNOWLEDGE SOURCE to be used,
c is the COMMUNICATIVE GOAL to be achieved,
u is a USER MODEL,
d is a DISCOURSE HISTORY

It is difficult to formalize the characteristics of KNOWLEDGESOURCE other than to say it contains an encoding of information from one or more applications or databases. However, it can contain the precise information to be presented, or it will be the task of the NLGsystemto select the appropriate content from k.

COMMUNICATIVE GOAL of a task is to be distinguished with the overall purpose of the NLG system. c depends upon the k we are using.

USER MODEL identifies the indented audience for whom the text is generated. This also requires that the output is at the level of the audience, be it novice or expert.

DISCOURSE HISTORY keeps in track the history of communication with the audience. This is applicable in dialogue-based systems and situations where end-user experience is personalised.

Components of an NLG system

Existing NLG systems typically carry out seven basic kinds of tasks, from input to output. These tasks can also be executed in parallel.

1. Content Determination: The process of deciding what information should be communicated. This is influenced by communicative goals, target audience, available screen real estate, availability of underlying information for that instance to communicate.

2. Content Structuring: Imposition of ordering and structure over the set of messages to be conveyed to make it more discernible. This influences the choice of sentence and paragraph boundaries. Structuring can be achieved using templates, corpus and heuristics.

3. Lexicalization: This process determines the choice of words (nouns, verbs, adjectives and adverbs) to convey the domain concept or theme of the message relative to the user model. This also involves pragmatic consideration for the formal and informal circumstance. Addressing monotony by bringing variety in describing a similar circumstance.

4. Referring expression generation: Thetaskto select words or phrases to describe the domain-specific entity to the target. The initial or subsequent reference of an entity can be determined based on discourse history. This decides on how an entity should be used in a message.

5. Sentence Aggregation: Grouping messages into sentences to combine messages and avoid redundancy.

6. Linguistic realization: Applies the rules of grammar to abstract representations specific to the target human language. It governs the word and sentence formation along with punctuations, casing conventions are correct for the specific natural language.

7. Structure Realization: Converts abstract document structures like paragraphs and sections into mark-up symbols understood by the document presentation component.

Architecture of an NLG System

There are many ways of building a system that performs the tasks outlined above. The common architecture in modern applied NLG systems generally has a three-stage pipeline with the following system :

Document Planner: This stage combines the content determination and content structuring tasks described above. This reflects the fact that in many real applications, it can be challenging to separate these two activities. The Document plan, which is the output of document planner structures the messages in a structure that specifies how messages are to be grouped and related so as to satisfy the initial communicative goal.

Microplanner: This stage refines and combines sentence aggregation, lexicalization, and referring expression generation so as to produce a fully formed text specification.

Surface Réaliser: Concerned with linguistic realization, along with the post-processing such as rendering the text in a specific output format(HTML, SGML, speech etc.), i.e. involves syntactic, morphological, and orthographic processing.

A brief comparison between NLGeneration and NLUnderstanding

Natural Language Understanding (NLU) and Natural Language Generation together form the field of natural language processing (NLP). NLU maps natural language to internal computer representations; natural language generation maps internal computer representation of information into human language.

But there are stark differences in the internal operations of NLU vs N LG at a more fundamental level. NLU is best characterized as one of hypothesis management: Given some input, which of the multiple possible interpretations at any given stage of processing, is the appropriate one? On the other hand, NLG is best characterized as a process of CHOICE: Given the different means that are available to achieve some desired outcome, which should be used?

They both have the same endpoints, and their difference lies in the fact that they are concerned with navigating between these endpoints in the opposite direction. Does this mean that a single system can encode form and decode to natural language? In spite of the intuitive appeal of the idea, in practice, it is difficult to build bidirectional systems—challenges like the need to comprehend grammatically incorrect input do not exist in NLG. On the other hand, challenges like ensuring generated text are discernible by humans is not a problem in NLU. The other general challenges include the incompatibility of internal representation and the fact NLU requires the variety of examples which express much of the same thing, whereas in NLG it is often good enough to have one way of saying something.

Machine translation systems often use bidirectional grammars where inputs texts are chunks of sentences, but such approaches are not widely used in NLG as it continues to depend on the knowledge source. Of course, some of this will change in the future as the general understanding of both NLU and NLG grows. However, it is unlikely that we will everview'NLG as NLU in reverse' in the foreseeable future.

Building Natural Language Generation Systems (Studies in ....
https://b-ok.org/book/1271553/c61ae9

Neural Methods for NLG

In this section, we introduce language modelling, and the latest use of Transfer Learning techniques for language generation Language modelling is defined as the task of predicting the next word given the word so far.

Systems which produce this conditional probability distribution that does this task are called language model. Language models make use of the rrig.rko.v-assumption, which states that the future is independent of the past given the present. Informally, a targeted word is predicted based on ImmeSTate previous word or past few words, including recent targets. Implementations of Language models have been around for a while in settings like web search and autocomplete in phones with techniques N-gram and HMM.

Conditional Language modelling is the task of predicting the next word given the word so far, and also conditioning on other input x .

Language modelling, in general, can be operated at the character level, word level, sentence level or even paragraph level. Probability of the next word given the history of previously generated words within the context of x.

For a long time, Vanilla RNN, which is a class of artificial neural network, has been successfully used for language modelling, but they had challenges beyond effectiveness in short-term dependencies. Variants of RNNs like LSTM and GRU are better at handling the long-term and bidirectional dependencies. RNN's with encoder-decoder architecture (Sequence2Sequence) achieve significant results in machine translation by preserving context between the input and output text using the "attention" mechanism.

Bidirectional-LSTMs with "attention" mechanism is increasingly common in chatbots to maintain context between chats and generate the next response.

Sequence2 Sequence models have proven to be useful for tasks like Named Entity Recognition, Automatic speech recognition and are of much research interest in machine translation for source-to-source programming language translation.

RNN based neural approaches involve sequential computation which inhibits extensive parallelization and require high-end hardware to handle large datasets.

Transfer Learning :

In classical Machine Learning approaches, training and testing happen in the same domain the model is intended for. Transfer learning is the use of models trained in one domain as the effective starting point for models in a related domain. Models pre-trained for general tasks are then fine-tuned for a specific task, thus saving the need for extensive data, compute and fine-tuning requirements. This makes transfer learning suitable for applications with less training data.

Advent of Transformers:

Transformer models are fundamentally changing how language modelling task is performed these days.

Transformer models rely on the encoding-decoding approach enhanced with " self-attention ". A simple intuition behind this approach is to generate similarity and context (i.e. embedding) for any word with all other words independent of its position in the text. Transformer achieves this by allowing encoder-decoder to see the entire input sequence all at once (Vasyyani et al., 2017). The use of self-attention provides shorter paths for information to travel, which is conjectured to be one of the main reasons that transformers achieve better results on common language modelling benchmarks compared to other models (Dai et al., 2019). Moreover, transformers trained on very large datasets can generalize to other NLP tasks, and generate realistic samples that are coherent over long timeframes (Radforde: al., 2019).

Transformer based architecture lays the foundation for Open AI GPT-3, Google BERT, Facebook XLM-RoBERTa , Microsoft Turing-NLG etc. While these models have subtle architecture differences, the general trend leans towards training in large up labelled data, self-supervised, larger models with billions of parameters. Evaluation of Transformer models has been based on benchmarks like GLUE and SuperGLUE, which are collections of diverse datasets which test for various NLU and NLG tasks in-depth.

The general sequence of training Transformer models like BERT is as below :
Use a default pre-trained model which is trained in BookCorpusand Wikipedia English
Fine-tune, the pre-trained model with a task-specific dataset
Use the fine-tuned model for the respective task.

arXiv:1904.08378v1 [cs.LG] 17 Apr 2019. https://arxiv.org/pdf/1904.08378.pdf

Some of these challenges are common to any machine learning and neural network setting but have a particular impact on NLG.

1. How much training data is actually required? While one of the drivers behind transfer learning is to use it when there is a lack of training data. Additional data is still required for fine-tuning the pre-trained model for a specific task. Enterprise application of NLG is likely to be domain-specific, resulting in the lack of enough labelled data for the desired quality of the result.
2. Concept drift: Concept drift is when data properties change over time. While web size datasets have been instrumental for Transfer Learning approaches in NLP, the implications of concept drift and updating the model are less understood. This invites further research in this space.
3. Content Fidelity: Content generated from Transformer based models can be found to lose coherence over a long sentence, semantic repetition and

contradict themselves. Neural methods of NLG suffer from Hallucination, where the generated content is not true or generate tendencies absent in the input data. This could potentially be restrictive for use in high fidelity environments.

4. Societal impact: Pre-trained language models bringdown the complexity for low-ski lied actors to generate harmful indistinguishable content at speed, especially in English. This concern has resulted in Open AI choosing a staggered approach for releasing the model and Microsoft not releasing technical artefacts.

5. Bias & Fairness: Web crawled data absorbs the biases, prejudices, entrenched stereotypes in the real world; this is eventually perpetuated in some form to the generated content. This invites caution and governance in the use of pre-trained language models.

The traditional division of labour in NLG can be split into two questions, "what to say" and "how to say it" separately, and leads to systems with explicit content selection, macro and micro-planning, and surface realization components. Figure 1 illustrates the trade-off between these two types.

Neural methods for language generation contrasts this by skipping the structured steps and predicting. Rule-based approaches have the benefit of being interpretable with better levers for control and predictability but require manual effort to scale. On the other hand, neural methods for text generation despite empirical success offer less control over the outcomes and are even less understood.

Evaluation of NLG Systems :

Evaluation components are almost obligatory for real-world systems: Determining how well the system meets the goals that it was indented to satisfy. One of the challenges in NLGis that everybody seems to agree on what the output of an NLG system should be, but what the exact input is can vary substantially (McDonald,1993). The knowledge source can be the usual structured data sources or the new sources like audio, images or video. The language generation system has to be adaptable for the type of source and domain. The range of possible outputs is only open-ended and left to the imagination. Typically, systems can be compared only if the input is similar. For NLG systems, the complexity in evaluation extends to the outputs as well. In general, assessment has been complicated as the different components of NLG systems have been researched separately. Hence there is less consensus on evaluation standards.

In this section, we will focus on the evaluation approaches for data-to-text generation, which could be adopted to standard NLG use-cases for enterprises with regards to report and narrative generation. SjjarckJones and GaJJiers (1996) frame an approach based on intrinsic and extrinsic methods.

Intrinsic Evaluation methods

An intrinsic evaluation measures the performance of a system without reference to other aspects of the setup, such as the system's effectiveness in relation to its users. Intrinsicevaluations are dominated by two methodologies, evaluation based on human ratings and judgements, the other using corpora.

- Subjective evaluations based on human ratings and judgements: Evaluation of the generated texts on an n-point rating scale. The human experts rate the generated text on a number of dimensions like overall quality and coherence, content, organization, writing style and correctness. Similar rating scale exercises can be performed by showing a different version of the same text for the users to vote on the preferred version. Human evaluations are generally subjective, exhibit high variance, costly in terms of time and resources. These days human evaluation can be sourced from avenues like Amazon MecfinLca)Turk and CrowdFlower for widely spoken languages I like English and German.

- Objective Human likeness measures using corpora: This approach is based on automatic evaluation methods inherited from other closely related linguistic fields I like Machine Translation and Automatic Summarization. Text generated from NLG systems is compared with Human -written reference tests. These techniques are predominantly based on word overlap (e.g. BLEU, ROGUE), string distance (e.g. Levenshtein distance, TER) and content overlap (e.g. Jaccard index, MAS I). The focus of these metrics is on the output text, rather than its fidelity to the input. The appeal behind this approach is that it is relatively cheap, quick and repeatable.

A more intrinsic approach is to measure perplexity, which is a way to capture the degree of uncertainty a model has in predicting unseen text. Low perplexity is always preferable.

- Learned automatic evaluation metrics based on Transfer learning: This approach banks on the recent advancements in Transformerbased models, specifically BERT. Ratings are generated based on contextual similarity (BERTScore) and extend the same with novel pre-training(BLEURT) which use synthetic and public human rating data.

Despite the ease of these approaches, it should be understood that they do not necessarily measure the performance of the NLG system as a whole.

- Evaluating Genre Compatibility and Stylistics Effectiveness: This is a particularly exciting aspect involving the evaluation of language generation involving personas like a weatherman, economist, sports commentator etc. including a specific "tone of voice". In such situations, it is desirable that the generated text has stylistic variations. The intention behind these variations is to increase the effectiveness of the communicative goal.

Extrinsic Evaluation methods

Task-based evaluations involve directly measuring the impact of generated texton end users. This type of evaluation consists of a group of users or domain experts performing objective evaluations to measuring if the communicative goal or desired behaviour was achieved. In modern systems, this type of evaluation is made by capturing logs, clickstreams and signups etc. Requirements for this type of evaluation has to be factored early in the design process of real-world NLG systems.

Evaluation: Concluding remarks

In order to have a holistic understanding of the effectiveness of the language generation system under evaluation, it is best to have multiple evaluation methods. The reporting of the results should also include the correlation between them. Unfavourable correlations need not imply that the results of particular methods are invalid. Instead, they may indicate that measures focus on different aspects of a system or its output.

In general. Natural language Generation refers to any setting in where text is generated,
NLG is a subcomponent of many some of these tasks below:

text-to-text: Machine translation, Summarization, Dialogue, question answering, creative writing
data-to-text : Intelligent report generation image-to-text: image captioning
text-to-speech: Spoken Dialog Systems

Smart Report Generation: NLG is used to create a textual summary to structured data from databases as a complement or replacement to standard visualizations, Extended use of these summaries is to orient the user before interacting with the visualization. This can range from a listed company earnings article to regular management dashboards for sales, operations etc.

Hyper-personalization: NLG can perfect the digital experience by generating personalized and engaging content based on various heuristics which are unique to the individual user from landing page to emails.

Publishing industry: Automatic and dynamic content creation often co-authoring with human contributors for rich and engaging content.

IoT: NLG systems are integrated into the ever-increasing IoT devices to send meaningful and personalized alerts and interactions from home automation systems.

Gaming and chatbots: représenta real-time interactive domain for NLG. It helps bring out personalities and emotions of characters that interact with each other and also with human players. NLG systems reduce the burden of authoring dialogue for believable characters in games. The same can be used to enable chatbot conversations to be more persona and contextual driven.

Robotic Process Automation: NLG systems combined with RPA agents can automate the process of data access from multiple systems, combine the output and trigger custom text outputs

Voice-based systems: Text-to-speech synthesis (TTS) takes text created from NLG and converts it to speech.

Software Engineering: NLG systems can be used to automatically generate documentation on data schemas which can be particularly useful for supporting legacy systems.

Levels of NLG

There are different levels of sophistication in generating texts. These levels are still, of course, a bit fuzzy and tend to overlap when it manifests in tools[].

Level 1: Simple Fill-In-The-Blanksystems

Level 1 systems can be classified as the fill-in-the-blank template systems. As cited previously, windows mail-merge fills predetermined slots in word document by retrieving data from spreadsheet or database etc. It also allows for some level of flexibility and the use of if-then-else type conditional logics. Using this type of approach is acceptable for elementary applications, but it gets difficult to create and maintain anything considered more than basic.

Level 2: Rule-based approaches for producing text

Level 2 systems are essentially Level 1 systems with enablement from general-purpose programming languages. These scripting or programming languages provide the ability to support complex conditional logics, loops and better text processing capabilities in the form of pre-built libraries. Templates which reflect the business requirements are created and
Beyond the speed of text generation, they can also be embedded to fully functional web applications. But there is a lack of any linguistic capability that will generate complex, high-quality texts.

Level 3: Word-Level Grammatical Functions

Level 3 systems add word-level grammatical functions to Level 2 systems. They have the ability to deal with things like morphology(e.g., plural of a child is children, not childs), morphophonology (e.g., choosing between a or an), and orthography (e.g., one instead of two".." at the end of I like Washington D.C.). This makes it significantly easier to generate grammatically correct texts, and reduce the complexity which would otherwise be handled by template systems.

Beyond grammatical correctness, Level 3 systems do not necessarily cater to other aspects of text generation.

Level 4: Dynamically creating sentences

Level 4 systems dynamically create sentences and paragraphs based on an input about the target meaning to be conveyed. Dynamically creating sentences in this fashion means the system can do sensible things in unusual (edge) cases, without needing the developer to explicitly write code for every edge (boundary) case. It also allows the system to I linguistically "optimise" sentences in a number of ways, induce ng reference, aggregation, ordering, and connectives. For example, producing John was hungry, so he ate an apple. He was also cold ; instead of John was hungry. John was cold. John ate an apple.

Level 4 systems do an excellent job of producing high-quality sentences and paragraphs that is 'micro-level1. But for full-fledged meaningful content, better 'macro-level' capabilities are required.

Level 5: Dynamically Creating Documents

Level 5 systems add intelligence for "macro-level" task in the text generation, i.e. The task of producing a full-fledged, well structured and relevant document. How this is done depends on the goal of the text. For exam pie, a text that is intended to be persuasive may be based on models of argumentation and behaviour change; while a text that summarises data for decision support may be based on an analysis of key factors that influence the decision, plus models of narrative and human decision-making.
Level 5 systems do an excellent job of producing good quality narratives. Here, Level 1 can be considered as "templates" and Level 5 as "IMLG". Enterprise solution evaluations should consider the many levels of product sophistication. Use cases merit the tool choice; however, tools at Level2 and3 can also claim to be "real" NLG.

Considerations for NLG

Before exploring the technical content of work in NLG, it is worth discussing some practical considerations before fielding an NLG system to work. The challenges of implementing an NLG system is similar to that of any new application in an enterprise. The first point of entry into the requirements analysis is with the question if NLGis the most appropriate technique for the specific problem. There are more straightforward ways to generate text in the form of mail-merge, which is part of most Windows Office packages. It involves the insertion of input into predefined slots in the template. Mail-merge based solutions are easy to create for the first time, but as the scale and sophistication increase, can become challenging to maintain when more enhancements are required.

- Ownership and Access: One of the vital challenges of most NLG projects is to have access to the data source and be engaged in the continuous changes related to that data source
- Data Format: Generally, structured data formats(databases, excel sheets, etc.) are best suited for language generation. Access to unstructured data is not a limiting factor, but the ability to transform it to workable format needs to be considered,
- Data Quality: Data source with missing data can be challenging for traditional reporting or business intelligence systems. It will be far more difficult for effective language generation if data quality is not satisfactory for standard reporting. Most NLG systems will not have the required data transformation modules to compensate at its level.
- Data completeness: In spite of its sophistication NLG system scan not generate text if the necessary data for the intended task is either missing or not available at the moment of generation. Human experts are better rare compensating for the lack of data.
- Data integrity: Accuracy, consistency and reliability of the data source should be mastered upstream for the NLG system to focus on its core tasks

- Tolerance for computer-generated mistakes Organisations can be far more intolerant towards computer-generated errors when compared to their human counterparts.

Reasons for considering NLGsystems :

- Consistency: NLGsystems can bring inconsistency in writing a narrative provided the input data is as expected. Human authors are far more creative but can be limited by relevant domain experience, bias and general conditions relating to the vagaries of the human mind. Mistakes can be detected but will require layers of quality assurance personnel and process.
- Conformance to standards: NLG system can be tuned to stick to a style, format and set vocabulary. This particularly becomes important in situation's involving regulators.
- Speed of document production: NLG systems when adequately integrated with technology can generate narratives as and when the relevant data is available. This can free up capacity to analyse the data rather than create the report
- Multilanguage support: NLG systems can produce text in several languages. Finding multilingual human experts can mainly get difficult.
- Variability of the output: NLG systems can be built to avoid canned responses by introducing a level of predetermined variations.

NL4XAI

"How did this happen?" is a natural question for anyone dealing with AI systems. This is also a limiting factor for broader adoption of AI as there is uncertainty around how "intelligent" machines arrive at decisions. Aiding humans to verify the flow of AI decision structures and to revise or ratify their anecdotal and ground-truth knowledge can help build better trust in AI systems. Explainable AI (XAI) is the subfield of AI which focus on making complex AI models more understandable to both humans and machines.

Role of NLG in XAI

Natural language, as an intuitive medium has the potential to increase the intelligibility of AI systems. If explanations are generated in a natural language, it needs to imbibe the language-specific narrative structure, explicitly communicate uncertainty and communicate at the level of the audience.

Here are some crucial arguments driven by real-world situations in favour of XAI

- System verification and validation: Verification and Validation are a critical part of quality assurance practices in general. This challenges any default trust in black box algorithms in spite of high accuracy.
- Improvement of the system: Gaining an understanding of the choices models make to arrive at predictions enable better model comparison beyond accuracy and error metrics. This also makes it easier for engineers to debug the system.
- Learning from the system: XAI systems can extract distilled knowledge from systems that have learnt from data at scale. These novel insights can facilitate knowledge transfer from machine to humans.
- Regulatory compliance: Assignment of responsibility on rouge or unsatisfactory outcome has become increasingly crucial for the government and regulatory bodies. Thus the capability to understand point-in-time decisions, reproducing and explaining the "thought" behind a prediction is vital for regulatory scrutability.

There are different post-modelling methods forexgJainabiMty and interpretability, which vary based on the use-case and domain. Some conventional approaches include visualisation of variable importance, What-if analysis, LIME, S HAP etc. In general, these methods attempt at explaining predictions using different strategies at the local or global level, for neural network-based algorithms it is common to see backpropagation based

methods explanation like layer-wise Relevance Propagation (LRP), DeegjJFT, Guided Backprop (GB), SmoothGrad, and Integrated Gradients (IG).
With just this quick survey, it makes it clear the approaches to achieve XAI are equally sophisticated. This is can now be seen as a practical problem of increasing interest, especially with the ever-widening disséminât ion of AI systems; this is a visible area where NLG can contribute.

This goes back to the core tenets of NLG as a communicative goal-driven process to drive the desired effect in the audience. There are four different stakeholder communities in XAI: Developers, Theorist, Ethicistand the User[], They have specific interest in the outcome of the AI system, any narration generated for them will require different levels of abstraction vs granularity trade-offs. For example, a metrics-driven approach for explain ing a smoking prediction is to simply list all the model features and their proportional influence on the result.
"John is a white male. John has been smoking a pack a day for 50 years. John is 67 years old.
John does not have a family history of lung cancer. John is a high school graduate.
John has a 6% chance of developing lung cancer within the next 6 years."

While there might be some appeal for the level of detail in a few areas, the usability of the prediction is limited if humans are unable to justify their trust in the prediction. This requires generating narrations for the non-expert to increase their satisfaction in the justification^.
"John has been smoking a pack a day for 50 years, so he may develop lung cancer even though he does not have a family history of lung cancer."
It is to be noted that this is not an accurate description of how the model works, but it might be a better explanation for the target audience.
Natural language for XAI is in its early days of research, and it is likely to gain traction. Some challenges have been identified, which are essential for generating good explanation:

- Evaluation: Develop "cheap but reliable" ways of estimating scrutability, trust, etc.

- Vague Language: Develop good models for the use of vague language in explanations.
- Narrative: Develop algorithms for creating narrative explanations.
- Data Quality: Develop techniques to let users know how results are influenced by data issues.

All these are generic N LG challenges that are important in NLG, but these.
