UNIT IV: CHAPTER 1
Natural Language Generation

Topics:

1. Introduction to NLG

2. NLG Architectures

3. Generation Task and Representations

4. Applications of NLG

Introduction to NLG

Natural Language Generation (NLG) is a branch of artificial intelligence (AI) that focuses on automatically generating human-like text from structured data or raw information. It is a subfield of Natural Language Processing (NLP) and is often used in applications such as chatbots, content automation, and report generation.

The process of generating text can be as simple as keeping a list of readymade text that is copied and pasted. Consequences can either be satisfactory in simple applications such as horoscope machines or generators of personalized business letters. However, a sophisticated NLG system is required to include stages of planning and merging of information to generate text that looks natural and does not become repetitive.

NLG systems typically follow these steps:

1. **Content Determination** – Deciding what information needs to be included.
2. **Text Structuring** – Organizing the content in a logical sequence.
3. **Sentence Aggregation** – Combining related information into coherent sentences.
4. **Lexical Selection** – Choosing appropriate words and phrases.
5. **Refinement & Formatting** – Adjusting grammar, punctuation, and presentation.

- Data-to-Text Applications (e.g., weather forecasts, stock market updates)

Stages of Natural Language Generation

- **Content determination**: Deciding the main content to be represented in a sentence or the information to mention in the text.

- **Document structuring:** Deciding the structure or organization of the conveyed information.
- **Aggregation:** Putting of similar sentences together to improve understanding and readability.
- **Lexical choice:** Using appropriate words that convey the meaning clearly.
- **Referring expression generation:** Creating such referral expressions that help in identification of a particular object and region. This task also includes making decisions about pronouns and other types of anaphora.
- **Realisation:** Creating and optimizing the text that should be correct as per the rules of grammar. For example, using will be for the future tense of to be.

Techniques for Evaluating NLG systems

1. **Task-based evaluation:** It includes human-based evaluation, who assess how well it helps him perform a task. For example, a system which generates summaries of medical data can be evaluated by giving these summaries to doctors and assessing whether the summaries help doctors make better decisions.
2. **Human ratings:** It assess the generated text on the basis of ratings given by a person on the quality and usefulness of the text.
3. **Metrics:** It compares generated texts to texts written by professionals.

Architectures of NLG Systems

1. **Traditional Rule Based Architecture**
2. **Template Based Architecture(Pipeline Architecture)**
3. **Statistical & Machine Learning Based Architecture**
4. **Deep Learning Based Architecture**
5. **Hybrid Architecture**

1. Template-Based NLG

Template-based NLG is one of the simplest methods for text generation. It follows pre-defined templates with placeholders that are filled with structured data.

Working Mode :

- A human creates fixed sentence structures (templates).
- The system fills these templates with relevant data.
- The output follows a consistent format.

Example:

Template:

"The weather in [City] is currently [Temperature] degrees with [Condition]."

Data Input:

{City: "New York", Temperature: "25°C", Condition: "Sunny"}

Generated Output:

"The weather in New York is currently 25°C with sunny conditions."

Advantages:

✓ Easy to Implement – Simple rule-based text generation.

✓ High Control – Ensures grammatically correct sentences.

✓ Useful for Structured Data – Works well for generating reports (e.g., weather, sports, financial updates).

Limitations

❌ Limited Creativity – Can only generate predefined structures.

❌ Not Scalable – Needs new templates for different topics.

❌ Cannot Handle Context – Lacks dynamic text variation.

Use Cases

✓ Weather Reports

✓ Financial Summaries

✓ Sports Game Reports

✓ Personalized Email Templates

---

2. Rule-Based NLG

Rule-based systems use linguistic rules (grammar, syntax, and vocabulary) to generate text dynamically. Unlike template-based approaches, they allow some flexibility in sentence construction.

Working Mode :

- Uses if-then rules to determine sentence structures.
- Applies grammar rules to combine words correctly.
- Uses lexical databases (e.g., WordNet) to vary vocabulary.

Example: News Report Generation

Rule:

- If Team A wins → "Team A defeated Team B with a score of X to Y."
- If Match is Drawn → "The match between Team A and Team B ended in a draw."

Input Data:

- Team A = "Manchester United"
- Team B = "Chelsea"
- Score = "2-1"

Generated Output:

"Manchester United defeated Chelsea with a score of 2-1."

Advantages:

✓ More Flexibility Than Templates – Can generate dynamic sentences.

✓ Ensures Correct Grammar – Uses linguistic rules.

✓ Good for Structured Domains – Works well for reports, legal texts, and business documents.

Limitations

❌ Requires Manual Rule Creation – Time-consuming and difficult to scale.

❌ Rigid Language Handling – Cannot generate natural, creative text.

❌ Fails in Unstructured Data – Cannot generate open-ended conversations.

Use Cases

✓ Automated Journalism

✓ Legal Document Drafting

✓ Chatbots (Basic Responses)

✓ Customer Support Responses

---

3. Machine Learning-Based NLG

Machine Learning (ML)-based NLG improves upon rule-based approaches by learning linguistic patterns from data rather than relying on predefined rules.

Working Mode:

- Uses statistical models to predict words based on previous text.
- Trained on large datasets to learn grammar, syntax, and structure.

- Uses probabilistic methods to generate sentences dynamically.

  Example: Email Autocomplete
- Input: *"Looking forward to"*
- ML Model Suggests: *"seeing you tomorrow!"*

The model learns from historical data and predicts the most likely completion.

Types of ML-Based Models

1. N-Gram Models – Predicts the next word based on previous N words.
2. Hidden Markov Models (HMMs) – Uses probabilities to generate sentences.
3. Conditional Random Fields (CRFs) – Used in sequence prediction tasks.

Advantages

✓ Learns from Data – Can adapt without manually defining rules.

✓ More Flexible – Can generate unseen sentences dynamically.

✓ Improved Accuracy – Learns from large datasets over time.

Limitations

❌ Requires Large Training Data – Needs labeled datasets to learn patterns.

❌ Limited Context Understanding – Cannot handle long-range dependencies.

❌ Not Fully Context-Aware – Predictions are based only on previous words.

Use Cases

✓ Predictive Text (e.g., Smartphone Keyboards)

✓ Basic Chatbots

✓ Email Auto-Completion

✓ Product Description Generation

---

4. Deep Learning-Based NLG

Deep Learning (DL) takes machine learning further by using neural networks to generate human-like text. These models understand context better and generate more coherent, natural language.

Working Mode :

- Uses Recurrent Neural Networks (RNNs) or Transformers to generate text.
- Trained on massive datasets (millions of books, articles, and conversations).
- Can generate long, contextually relevant responses.

Key Deep Learning Models

1. Recurrent Neural Networks (RNNs)
    - Handles sequential data (e.g., chatbot responses).
    - Struggles with long-term context retention.
2. Long Short-Term Memory (LSTM) / Gated Recurrent Units (GRU)
    - Improved RNNs that remember long-term dependencies.
    - Used in conversational AI and predictive text.
3. Transformer Models (State-of-the-Art)
    - Uses self-attention to process entire sentences at once.
    - Generates high-quality, fluent, and context-aware text.
    - Examples: GPT-3, GPT-4, BERT, T5, BART.

Example: AI Chatbot Response

Input: *"Tell me about the Eiffel Tower."*

Generated Output:

"The Eiffel Tower, located in Paris, France, is a 330-meter-tall structure built in 1889. It is one of the most famous landmarks in the world."

Advantages

✓ Generates Human-Like Text – More fluent and natural.

✓ Understands Context and Grammar – Can generate creative, informative responses.

✓ Handles Long-Form Content – Good for essays, articles, and summaries.

Limitations

❌ Computationally Expensive – Requires high processing power.

❌ Risk of Bias – Models can learn biases from training data.

❌ Not 100% Reliable – May generate incorrect or nonsensical text.

Use Cases

✓ AI Chatbots (ChatGPT, Google Bard)

✓ Content Writing (Marketing, Blogging)

✓ Text Summarization

✓ Code Generation (e.g., GitHub Copilot)

---

5. Hybrid NLG

Hybrid architectures combine multiple approaches (rule-based, ML, and deep learning) for better control and accuracy.

Working Mode :

- Uses rule-based systems for structured content.
- Uses machine learning/deep learning for natural, creative text.
- Example: A news generator uses rules for headlines but deep learning for article content.

  Example: AI-Powered Customer Support
- Rule-Based Part: Greets users and asks for issue details.
- ML-Based Part: Classifies the user's issue.
- Deep Learning Part: Generates a personalized response.

Advantages

✓ More Control – Uses rules where precision is needed.

✓ Balances Accuracy & Creativity – Best of both worlds.

✓ Handles Multiple Use Cases – Good for structured and unstructured content.

Limitations

❌ Complex to Develop – Requires integrating different systems.

❌ Higher Computational Cost – Needs both rule-based and AI components.

Use Cases

✓ AI-Powered News Writing

✓ Personalized Marketing Emails

✓ Customer Support Chatbots

Generation Task and Representations in Natural Language Generation (NLG)

Natural Language Generation (NLG) involves producing human-like text based on given input data. The process relies on generation tasks and different types of representations that define how information is structured before being converted into natural language.

1. Generation Tasks in NLG

Generation tasks vary based on the type of content that needs to be created. Some tasks focus on producing structured reports, while others generate open-ended text. Below are key NLG tasks:

1.1 Text Summarization

- Goal: Generate a concise version of a given document while preserving key information.
- Types:
  - Extractive Summarization: Selects and rearranges important sentences from the input text.
  - Abstractive Summarization: Generates new sentences to summarize the main ideas (e.g., using transformer models like T5 and BART).
- Example:
  - Input: "The Eiffel Tower, an iconic structure in Paris, was built in 1889. It stands 330 meters tall."
  - Output: "The Eiffel Tower, built in 1889, is a 330-meter landmark in Paris."

1.2 Machine Translation

- Goal: Convert text from one language to another while preserving meaning.
- Examples:
  - English: "Hello, how are you?"
  - French (Generated Output): "Bonjour, comment ça va ?"
- Common Models: Google Translate (based on transformers like T5, mBART, and MarianNMT).

1.3 Dialogue Generation (Conversational AI)

- Goal: Generate human-like responses in a conversation.
- Types:
  - Rule-Based: Predefined responses.
  - Retrieval-Based: Selects best response from a database.
  - Generative: Creates new responses dynamically (e.g., GPT models).
- Example:
  - Input: "Tell me about Mars."
  - Output: "Mars is the fourth planet from the Sun, known for its red surface and thin atmosphere."

1.4 Story and Content Generation

- Goal: Generate creative text such as news articles, stories, or marketing content.

- Example:
  - Input Prompt: "Write a fantasy story about a dragon."
  - Generated Output: "Once upon a time, a mighty dragon guarded an ancient kingdom, breathing fire to protect its people."
- Common Models: GPT-4, GPT-3, Bard, Claude.

1.5 Data-to-Text Generation

- Goal: Convert structured data into natural language text.
- Example (Weather Report Generation):
  - Input (Data Table): {City: "New York", Temp: "25°C", Condition: "Sunny"}
  - Output: "The weather in New York is currently 25°C with sunny skies."
- Applications: Sports reports, financial summaries, automated journalism.

1.6 Code Generation

- Goal: Generate code from natural language descriptions.
- Example:
  - Input: "Write a Python function to calculate the factorial of a number."
  - Generated Output:

```
def factorial(n):
    return 1 if n == 0 else n * factorial(n-1)
```

- Common Models: OpenAI Codex, GitHub Copilot, AlphaCode.

---

2. Representations in NLG

To generate text, models need a structured way to represent information before converting it into language. There are different representation types:

2.1 Symbolic Representations

- What it is: Uses logical rules, ontologies, and templates.
- Example:
  - Knowledge graphs and rule-based systems.
- Used in: Rule-based NLG, Template-Based NLG.

2.2 Statistical Representations

- What it is: Uses probabilities to determine the most likely word sequences.
- Example: N-gram models, Hidden Markov Models (HMM).
- Used in: Early machine learning models.

2.3 Vector Representations (Embeddings)

- VR Represents words as mathematical vectors in high-dimensional space.
- Example: Word2Vec, GloVe, FastText.
- Used in: Deep learning-based NLG (e.g., GPT, BERT).

2.4 Transformer-Based Representations

- What it is: Uses self-attention mechanisms to process entire sequences at once.
- Example:
  - BERT: Context-aware bidirectional embeddings.
  - GPT: Autoregressive transformer-based text generation.
- Used in: State-of-the-art NLG models.

Applications of NLG

Natural Language Generation (NLG) is a subset of Artificial Intelligence (AI) and Natural Language Processing (NLP) that involves generating human-like text from structured or unstructured data. It has numerous applications across various industries. Here are some key applications:

1. Chatbots & Virtual Assistants

- NLG powers AI-driven chatbots and virtual assistants like Siri, Alexa, and Google Assistant.
- It enables them to generate human-like responses based on user queries.
- It enhances customer service, automates responses, and improves user engagement.

2. Automated Content Generation

- NLG is used to create articles, summaries, and reports in domains like finance, sports, and journalism.
- Example: News agencies use NLG to generate financial reports or sports match summaries.
- Helps reduce manual effort and speeds up content creation.

3. Business Intelligence & Data Reporting

- Converts structured data into meaningful narratives.
- Used in business dashboards to provide insights in natural language rather than raw numbers.

- Example: NLG in financial analytics generates earnings reports, market insights, and data-driven forecasts.

4. E-commerce & Personalized Recommendations

- Generates product descriptions, reviews, and personalized recommendations based on user behaviour.
- Example: Amazon and eBay use NLG for automated product descriptions.
- Enhances customer experience and increases conversion rates.

5. Healthcare & Medical Reporting

- Helps in generating clinical reports, medical summaries, and diagnostic insights from patient data.
- Example: IBM Watson uses NLG to provide medical insights from patient records.
- Reduces the workload of doctors and improves accuracy in documentation.

6. Legal Document Automation

- Assists in drafting contracts, legal summaries, and case reports.
- Reduces time spent on legal documentation.
- Example: AI-powered legal assistants generate contracts based on predefined templates.

7. Education & E-Learning

- Generates personalized learning content and automated feedback for students.
- Example: AI tutors use NLG to explain concepts and provide real-time learning assistance.
- Helps in adaptive learning and enhances student engagement.