Natural Language Generation

Instructor: Jackie CK Cheung

COMP-550

Goals for Today

Explore different natural language generation settings that have been proposed in the literature.

Discuss approaches and techniques for NLG, including ones that are not currently popular

- Rule-based systems
- Machine learning
- Declarative approaches to optimization

Outline

- Data-to-text generation
  - Components of a NLG system
  - Rule-based methods
- Text-to-text generation
  - Sample tasks
  - Integer linear programming – declarative problem solving
  - Current trends

Extraction vs. Abstraction

Reminder:

Extraction – take snippets from the source text and put them in the summary

Abstraction – compose novel text not found in the source

- Allows better aggregation of information
- Requires **natural language generation**

Natural Language Generation

Let's compare understanding and generation

Concerns of NLU:

- Ambiguity (e.g., get all possible parses)
- Disambiguation
- Underspecification

Concerns of NLG:

- Selecting appropriate content
- Selecting appropriate form to express content

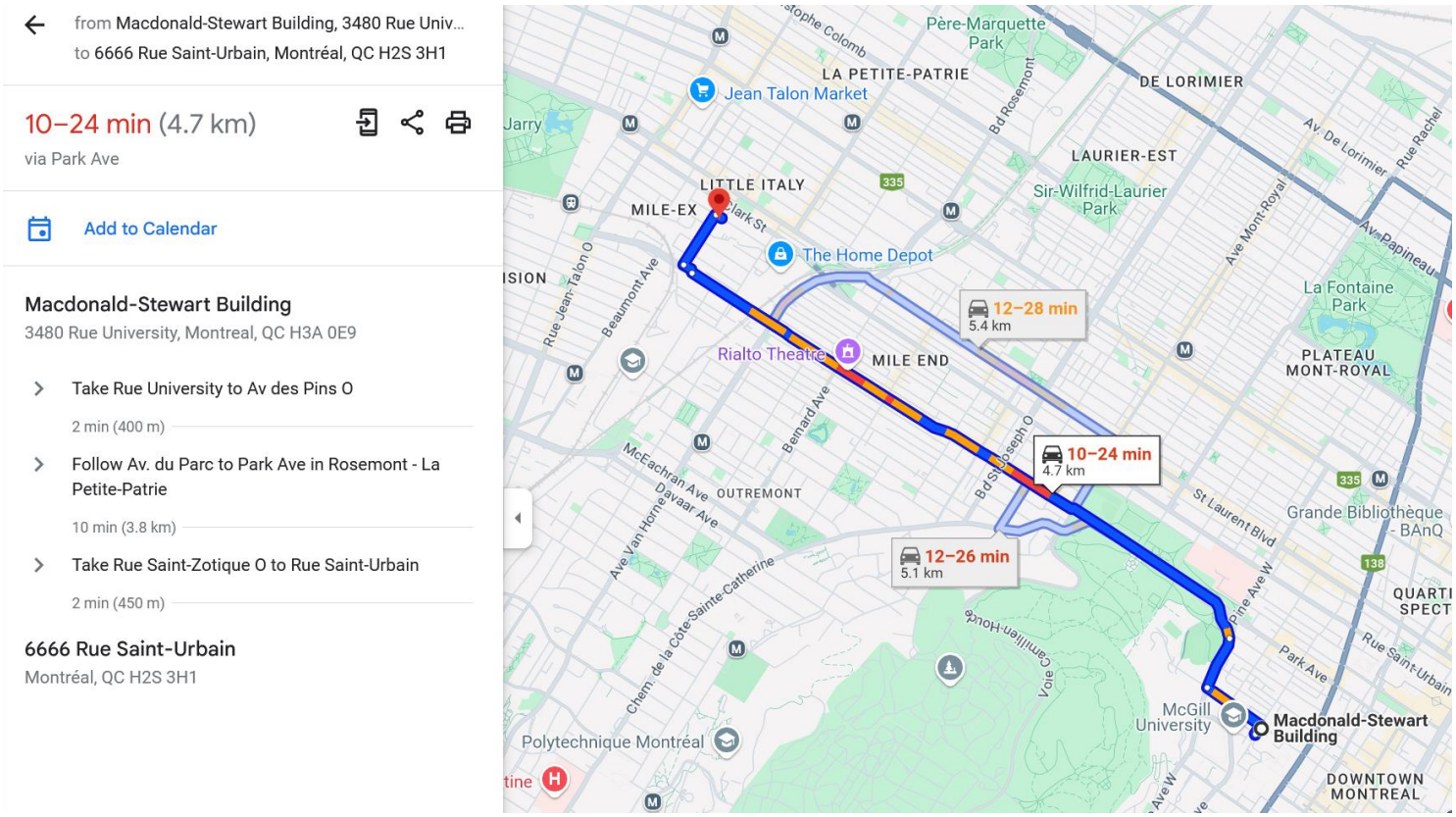
NLG can be divided into **data-to-text** vs. **text-to-text** generation.

Canned Text


MARIO
058700
×20
WORLD
1-4
TIME
200
THANK YOU MARIO!
BUT OUR PRINCESS IS IN
ANOTHER CASTLE!

Template Filling

Good for restricted domains.



What is the generation template?

Steps in NLG

One potential architecture for an NLG system:

1. Content selection
2. Document structuring
3. Microplanning
4. Surface realization

Content Selection

Deciding what to say

Ingredients:

- Communicative goal
- Knowledge about the world

Application-specific

- How did we approach content selection last class in multi-document summarization?

Document Structuring

Deciding how to structure the contents of the output

What order should they be presented in? Some factors:

- Importance of the concepts
- Discourse relations
- Coherence

e.g., **Argumentation Theory** gives some guidelines on how to arrange information

- Present main claims first
- Arrange and discuss supporting evidence
- Present and debate opposing evidence

(Carenini and Moore, 2006)

Microplanning

Selecting lexical items

- (BLZRD, -5, -10, 30km/h, MONTREAL) -> *blizzard, low, high, wind speed, Montreal*

Deciding how they fit together into clauses and sentences (**sentence planning** or **aggregation**)

- First sentence: present location and time that weather forecast pertains to
- Second sentence: present details of forecast

Generating referring expressions

- *Justin Pierre James Trudeau PC MP; Justin Trudeau; the Prime Minister; Mr. Trudeau; that guy; he; him*

Surface Realization

Convert fully specified discourse plan to output form (individual sentences, or other kinds of output)

Different possible levels of input specification:

- Highly detailed semantic structure, with all decisions made already (lexical items, tense, aspect and mood of verbs, referring expressions, etc.)
- Shallower kinds of semantics (e.g., similar to a dependency tree)

Reusable Components

There have been a few standard tools and task definitions in NLG:

- Referring expression generation
- Surface realization

Let's look at a surface realization system: FUF/Surge

FUF/SURGE

A cascade of deterministic rules to convert a structured semantic representation to a string:

(Elhadad and Robin, 1996)

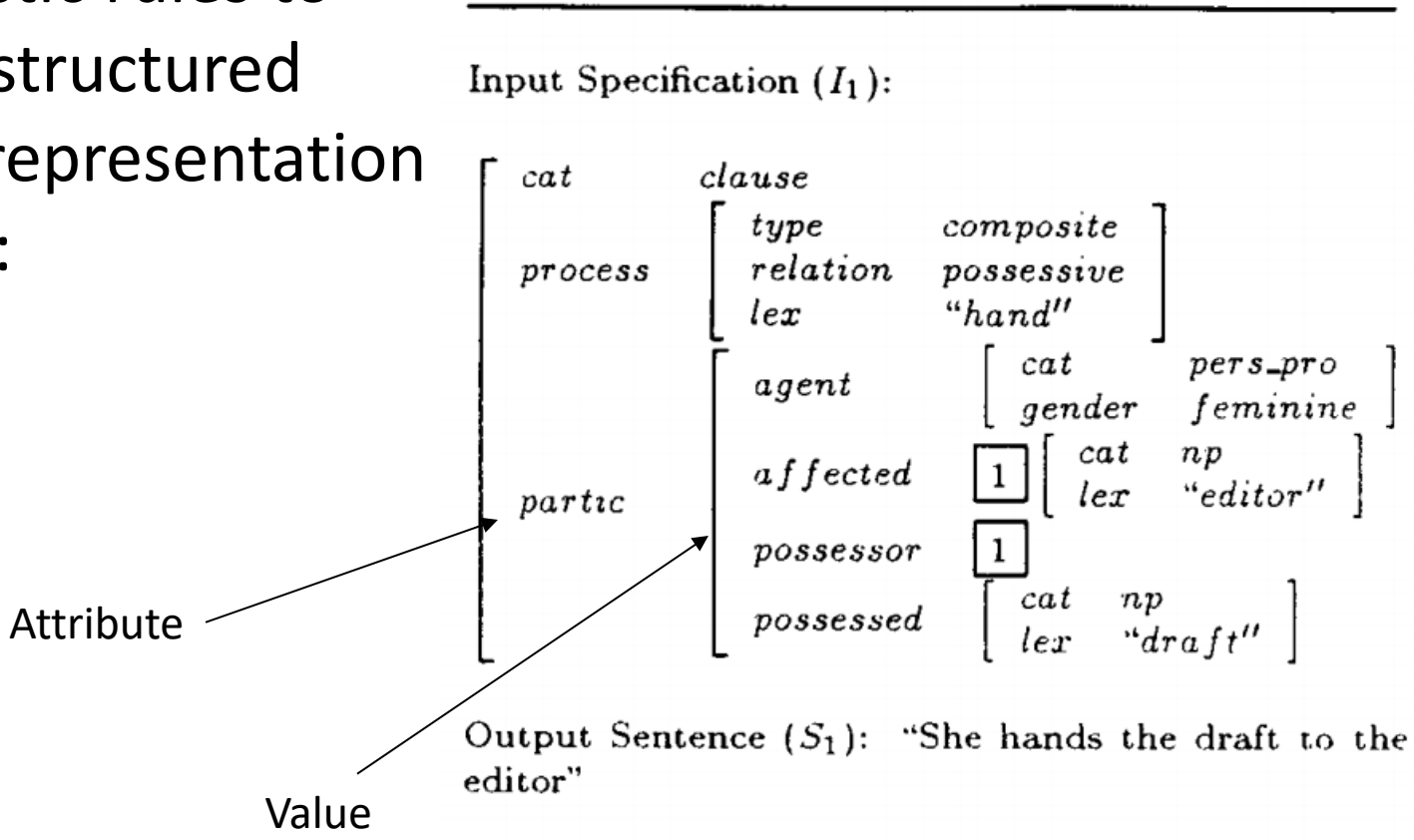


Figure 1: An example SURGE I/O

Components in FUF/SURGE

1. Map thematic structures (i.e., semantic roles) to syntactic roles

   e.g., agent -> subject

2. Handle syntactic alternations

   e.g., active-passive, dative alternation

3. Fill in default features, agreement features

   e.g., NPs are definite, if not otherwise specified

   subject and verb agree in number

4. Handle closed-class words

   e.g., [cat pers_pro, gender feminine] -> *she*

Components in FUF/SURGE

5. Order components with respect to each other

   e.g., subject > verb-group > indirect-object > direct object

6. Fill in inflections

   e.g., *to hand -> hands*

7. Linearize the tree into the final string, using precedence constraints

Neural NLG

Use a neural language model to generate the summary

Train a **conditional** language model:

$$P(x^{t+1} \mid source, x^{1 \cdots t})$$

- E.g., sequence-to-sequence model (Rush et al., 2015)

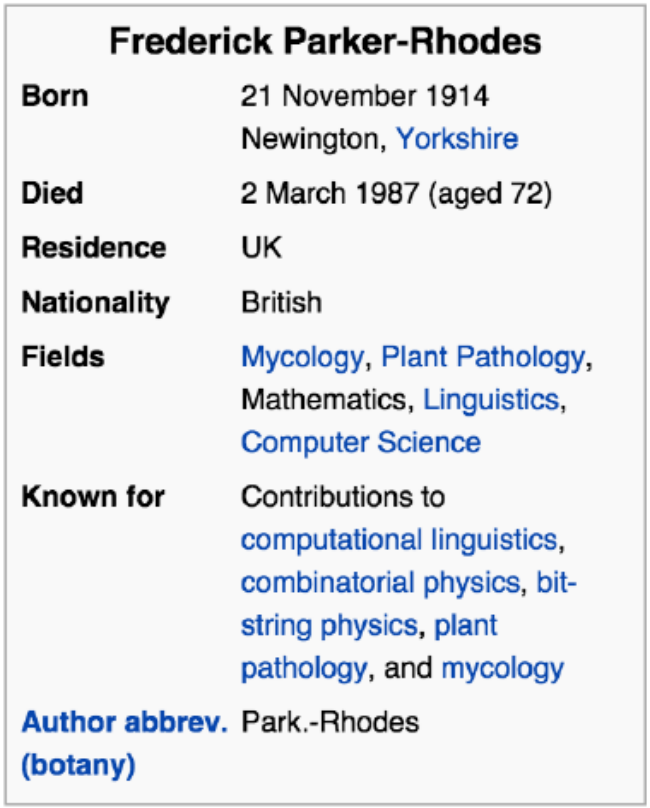
Neural Data-to-Text Generation

Basic idea:

- Decide which steps of the NLG pipeline should be their own modules (e.g., separate planning step?)
- Come up with a way to embed the input data structures into a format that is easy for the neural model to process
- Add mechanisms in the decoder that are appropriate for the generation task (e.g., attention; copy mechanism)

Generation of Wikipedia Biographies

Task: generate first sentence of Wikipedia biography article from its infobox



Frederick Parker-Rhodes	
Born	21 November 1914 Newington, Yorkshire
Died	2 March 1987 (aged 72)
Residence	UK
Nationality	British
Fields	Mycology, Plant Pathology, Mathematics, Linguistics, Computer Science
Known for	Contributions to computational linguistics, combinatorial physics, bit-string physics, plant pathology, and mycology
Author abbrev. (botany)	Park.-Rhodes

Frederick Parker-Rhodes (21 March 1914 – 21 November 1987) was an English linguist, plant pathologist, computer scientist, mathematician, mystic, and mycologist.

(Lebret et al., 2016)

Representation of Tables

Each word is associated with:

- word embedding,
- embedding that depends on word's presence in the infobox tables
  - (field_name; index_from_start; index_from_end)

Table (g_f, g_w)

name	John Doe
birthdate	18 April 1352
birthplace	Oxford UK
occupation	placeholder
spouse	Jane Doe
children	Johnnie Doe

input text (c_t, z_{c_t})

	John	Doe	(	18	April	1352	)	is	a
c_t	13944	unk	17	37	92	25	18	12	4
z_{c_t}	(name,1,2)	(name,2,1) (spouse,2,1) (children,2,1)	∅	(birthd.,1,3)	(birthd.,2,2)	(birthd.,3,1)	∅	∅	∅

output candidates $(w \in \mathcal{W} \cup \mathcal{Q})$

	the	...	april	...	placeholder	...	john	...	doe
w	1	...	92	...	5302	...	13944	...	unk
z_w	∅		(birthd.,2,2)		(occupation,1,1)		(name,1,2)		(name,2,1) (spouse,2,1) (children,2,1)

Typical neural LM training can then be applied.

Text-to-Text Generation

Traditional NLG: data-to-text

We can also start from **other text**.

Since we start with text, there must be something about the input that we'd like to change:

- Language (machine translation)
- Length (summarization)
- Complexity (text simplification, **sentence fusion**)
- Style (style transfer)

We'll talk a bit more about sentence fusion to present a new technique: **integer linear programming**

Sentence Fusion

(Barzilay and McKeown, 2005; Filippova and Strube, 2008; Thadani and McKeown, 2013; Cheung and Penn, 2014)

Combine information from multiple sentences. Take a *union* of information.

Bohr studied at the University of Copenhagen and got his PhD there.

After graduating, he studied physics and mathematics at the University of Copenhagen.

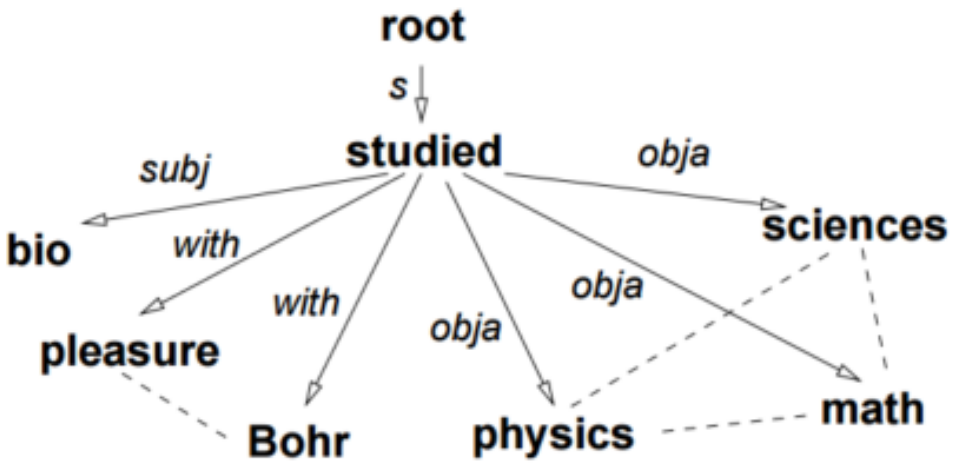
After graduating, Bohr studied physics and mathematics at the University of Copenhagen and got his PhD there.

Step 1: Sentence Graph

Create a **sentence graph** by merging the input sentences' dependency trees at the nodes with the same words.

e.g.: *He studied sciences with pleasure.*

+ He studied math and physics with Bohr.



(Filippova and Strube, 2008)

Step 2: Extract a New Sentence

Select a subset of nodes in sentence graph that will form a new dependency tree, from which a new sentence can be generated.

Problem: many desiderata and constraints

- Nodes must form a tree
- Selected nodes must contain the important words
- Selected nodes should make sense in relation to each other
- Desired output length

Would like a method that allows us to write down all of these hard and soft constraints

Solution: Integer Linear Programming

For each edge in the sentence graph from word *h* to word *w* with label *l*, create a variable x_{hw}^{l}.

$$x_{hw}^{l} = \begin{cases} 1 & \text{select this edge} \\ 0 & \text{don't select this edge} \end{cases}$$

Optimize the following objective:

$$f(X) = \sum_{x} x_{hw}^{l} \times P(l|h) \times I(w)$$

"Grammaticality" – how often this head word generates a dependent with this label

Importance of the dependent

Constraints in ILP

maximize $f(X) = \sum_x x_{hw}^l \times P(l|h) \times I(w)$

subject to

$$\forall w \in W, \sum_{h,l} x_{hw}^l \leq 1$$

$$\forall w \in W, \sum_{h,l} x_{hw}^l - \frac{1}{|W|} \sum_{u,l} x_{wu}^l \geq 0$$

First constraint ensures each word has at most one head

Second ensures that selected nodes form a connected tree

How would we constrain the number of words in the output?

ILP for NLG

Various other syntactic and semantic constraints

e.g., ensure that conjoints are similar to each other (*math and physics* is likely, *math and Bohr* is unlikely)

In general, ILP has advantages for NLG:

- Allows *declarative* specification of diverse objectives and constraints
- Can be solved fairly efficiently using off-the-shelf solvers

  http://lpsolve.sourceforge.net/5.5/

  http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/

Trends in Past Five Years

1. Handling OOV, named entities
   - Sub-word representations
   - Copy mechanism (using attention to be able to copy from the input text)
2. Correctness of generated output text
3. Controllable text generation
4. Generation with pre-trained language models

Correctness in Generation

A problem with abstractive summarization systems

Article: Jerusalem (CNN)The flame of remembrance burns in Jerusalem, and a song of memory haunts Valerie Braham as it never has before. (...) "Now I truly understand everyone who has lost a loved one," Braham said. Her husband, Philippe Braham, was one of 17 people killed in January's terror attacks in Paris. He was in a kosher supermarket when a gunman stormed in, killing four people, all of them Jewish. (...)

Original: **Valerie** braham was one of 17 people killed in january's terror attacks in paris. *(inconsistent)*

Corrected: **Philippe** braham was one of 17 people killed in january's terror attacks in paris. *(consistent)*

Train a post-editing model to detect and correct errors

$P(s \mid s', d)$

s: corrected summary
s': corrupt summary
d: source document

(Cao et al., 2020)

Controllable Text Generation

Ensure output has a certain style, formality, or a certain semantic content (or avoids saying something bad!)

MICROSOFT \ WEB \ TL;DR

Twitter taught Microsoft's AI chatbot to be a racist asshole in less than a day

By James Vincent | Mar 24, 2016, 6:43am EDT

Via The Guardian | Source TayandYou (Twitter)

SHARE

It took less than 24 hours for Twitter to corrupt an innocent AI chatbot. Yesterday, Microsoft unveiled Tay — a Twitter bot that the company described as an experiment in

Simple Approach: Control Token

Suppose you have a corpus of texts labelled with a property you care about.

Could you please … : polite

Do this now. : impolite

1. Prepend special tokens <POLITE> or <IMPOLITE> in front of the sequence and train a LM.
2. Then, at test, time control the output by prepending the token corresponding to the target property.

You could also put in desired words; contents; plan into decoder before starting generation.

Current approach: place in prompt of pretrained LLM.

Unlikelihood Training

Tell the decoder what NOT to generate

Assumption: you are able to derive this set of negative contrast words

Minimize the following loss:

$$\mathcal{L}^t_{\text{UL-token}}(p_\theta(\cdot|x_{<t}), \mathcal{C}^t) = -\alpha \cdot \underbrace{\sum_{c \in \mathcal{C}^t} \log(1 - p_\theta(c|x_{<t}))}_{\text{unlikelihood}} - \underbrace{\log p_\theta(x_t|x_{<t})}_{\text{likelihood}}.$$

(Welleck et al., 2019)

References

Cao, Dong, Wu and Cheung. 2020. Factual Error Correction for Abstractive Summarization Models. EMNLP.

Carenini and Moore. 2006. Generating and evaluating evaluative arguments. *Artificial Intelligence.*

Elhadad and Robin. 1996. An Overview of SURGE: A Reusable Comprehensive Syntactic Realization Component. *INLG*.

Filippova and Strube. 2008. Sentence Fusion via Dependency Graph Compression. *EMNLP.*

Knight and Marcu. 2000. Statistics-based Summarization – Step One: Sentence Compression. *AAAI*.

Lebret et al., 2016. Neural Text Generation from Structured Data with Application to the Biography Domain.

Rush et al., 2015. A neural attention model for abstractive sentence summarization

Welleck et al., 2019. Neural text (de)generation with unlikelihood training.