NLG Tutorial

Natural Language Generation Tutorial

By Pablo Duboue
Mad Scientist
What is NLG

• This tutorial: NLG from data
  • Tabular, structured, cold numbers
• When to use it
  • Text Output vs. Graphs
  • Capture generalization across dimensions
• Why should you care
  • You use templates all the time, what are you going to do if they are not enough?
• Beyond templates
Tutorial: 20 Botanicalls™

- Botanicalls™ explained as a mini NLG system
- Messages, verbalization
- 20 plants: a twitterbot
  - floorplan
- New data, new messages
- Simple approach
  - Drawbacks and counter examples

Three stages
- Decisions made at each stage
- Each stage adds information
  1. Content Planning
    - Choosing messages
    - Ordering and relating messages
  2. Text Planning
    - Naming entities
    - Aggregating clauses
  3. Surface Realization
    - all your base are belong to us: putting words
    - the 140 characters
• if moistAverage < DRY && lastMoistAvg >= DRY:
  • posttweet("URGENT! Water me!") [U]
• if moistAverage < MOIST && lastMoistAvg >= MOIST:
  • posttweet("Water me please.") [W]
• if waterVal >= SOAKED && lastWaterVal < MOIST:
  • posttweet("Thank you for watering me!") [T]
• if waterVal >= SOAKED && lastWaterVal >= MOIST:
  • posttweet("You over watered me.") [O]
• if waterVal < SOAKED && lastWaterVal < MOIST:
  • posttweet("You didn't water me enough.") [E]
Scenario 1

Living Room

Kitchen

Bedroom

Window

Couch

Bed

Bedroom
Scenario 2

- Living Room
  - 1
  - 5
  - 8
  - 9
- Kitchen
  - 2
  - 6
  - 7
- Bedroom
  - 10
  - 11
  - 12
  - 13
  - 15
  - 16
  - 17
  - 19
- Window
  - 4
- Couch
  - 3
- Bed
  - 14
  - 18
  - 20
Scenario 3

Living Room

Kitchen

Bedroom

window

couch

bed
How Not To Do It

- Need Water: 1 7 9 10 17; Urgent Need Water: 2 3 4; Thanks For Watering: 5 6; Over Watered: 11 12 13 14; Under Watered: 18 19 20
  - 129 characters!
  - Humans don't talk like that
    - Most at least ;-)
  - Defeat the \textcopyright botanicalls purpose
Content Selection

- Prioritizing the information
  - Urgent < Need Water < Soaked < Dry < Thanks
- Choosing at most two
- Choosing only one if too many instances (>50% plants) to verbalize
Document Structuring

- Two simple orderings:
  - By location
  - By color

- Which one to chose?
  - Heuristically, pick the most discriminant
  - Decision trees theory
  - Mutual information

- Minor orderings
  - Kitchen < Living Room < Bedroom
  - Red < Yellow < Blue
Data Structures So Far

• We have a document plan:
  • The selected messages with plants, colors and locations
  • An ordering over the messages, with a rationale
Scenario 1

{ByLocation} [ [W1,U2,U3,U4,W7],[W9,W10],[W12] ]
Scenario 2

{ByLocation} [ [] [U8, W9, W10, W11, W12] [U13, U14, W15, W16, W17, W18, W19, W20] ]
Scenario 3

{ByColor} [ [E3, E7, E8, E13, E14] [T1 T2 T4 T5 T6 T9 T10 T11 T12 T15 T16 T17 T18 T19 T20] ]
Scenarios Example

• Scenario 1
  • {Location ordered} [ [W1, U2, U3, U4, W7], [W9, W10], [W12] ]

• Scenario 2
  • {Location ordered} [ [] [U8, W9, W10, W11, W12] [U13, U14, W15, W16, W17, W18, W19, W20] ]

• Scenario 3
  • {Color ordered} [ [E3, E7, E8, E13, E14] [T1 T2 T4 T5 T6 T9 T10 T11 T12 T15 T16 T17 T18 T19 T20] ]
Aggregating

• Plant 1 and Plant 2 both need watering.
• Special case: all plants in the kitchen need watering.
• Output: list of data structures representing clauses. Each clause contains a predicate (water, thanks, etc) and a list of agents.
Sentence Level Aggregation

• Combine clauses into sentences.
• Take into account the length of the sentence.
• Do not cross document plan boundaries.
• Can add connectives likes “but” or “moreover”
Referring Expressions Generation

• Improve over “Plant 1 and Plant 7”
• Focus on distinguishing features
• Generate expressions for all the plants in a clause at the same time:
  • All plants in the kitchen
  • All blue plants in the kitchen
• If a specific plant needs to be named, use the shortest most distinguishing characteristic:
  • Second yellow plant from the living room
Output So Far

- List of data structures representing sentences
  - Each containing a list of data structures representing clauses plus connectives between clauses
  - Each clause contains a predicate, plus string, the resolved referring expression of the predicate
Scenario 1

{ByLocation} [ [ [:moreover: W("the first yellow by the window and the red opposite to the window") U("the other plants by the window")], [W("the blue and the first yellow")], [W("the yellow further from the bed")]] ]
Scenario 2

{ByLocation} [ [] [U(“the red”, W(“the yellows and the blue”))] [U(“the reds”), W(“the yellows and the blues”)] ]
{ByColor} [ [E(“all reds in the kitchen, the living room and the bedroom”)] [T(“all yellows in the kitchen, the living room and the bedroom”)] [T(“all blues in the kitchen, the living room and the bedroom”)] ]
Scenarios Example

- Scenario 1
  - \{ByLocation\} [ [ [:moreover: \text{W(“the first yellow by the window and the red opposite to the window”)}, \text{U(“the other plants by the window”)}, \text{W(“the blue and the first yellow”)}, \text{W(“the yellow further from the bed”)}] ]

- Scenario 2
  - \{ByLocation\} [ [] \text{U(“the red”), \text{W(“the yellows and the blue”)}} \text{U(“the reds”), \text{W(“the yellows and the blues”)}} ]

- Scenario 3
  - \{ByColor\} [ \text{E(“the ones in the kitchen”), E(“the one in the living room”), E(“the ones in the bedroom”)}} \text{E(“the ones in the kitchen”), E(“the ones in the bedroom”)}} \text{T(“the ones in the kitchen”), T(“the ones in the living room”), T(“the ones in the bedroom”)}, \text{T(“the ones in the kitchen”), T(“the ones in the living room”)}, \text{T(“the ones in the bedroom”)}, \text{T(“the ones in the kitchen”), T(“the ones in the living room”), T(“the bedroom”)}} ]
Surface Realization

- Transform each sentence into a string
- Use a list of possible verbalizations for the different predicates, chose one based on size
- The possible verbalizations need to take into account singular and plural referring expressions
- Also verbalize the connectives
- Finally, if the text exceeds 140 characters, some messages might need to be render in “emergency mode”.
Kitchen: Plant 7 and the window plants need urgent water. Living room: Plant 9 and 10 need water. Bedroom: Plant 17 needs water.
Living room: The red (urgent), the yellows and the blue need water. Bedroom: the reds (urgent), the yellows and the blues need water.
All reds in the kitchen, living room and bedroom were underwatered. All yellows and blues in the kitchen, living room and bedroom thank you.
Where to go from here

- Learning more about NLG:
  - Ehud Reiter book, website, simpleng java toolkit
  - Matthew Stone, NLG for games!
  - SIGGEN

- Starting from text
  - Processing the text: UIMA, Gate, LingPipe, etc
  - Summarization: NYU / Columbia
  - Machine translation: statistical

- Think outside the template!