CS4025: Content Determination and Document Planning

Document Planning

- First stage of NLG
- Two tasks
  - » Decide on content (our focus)
  - » Decide on rhetorical structure
- Can be interleaved

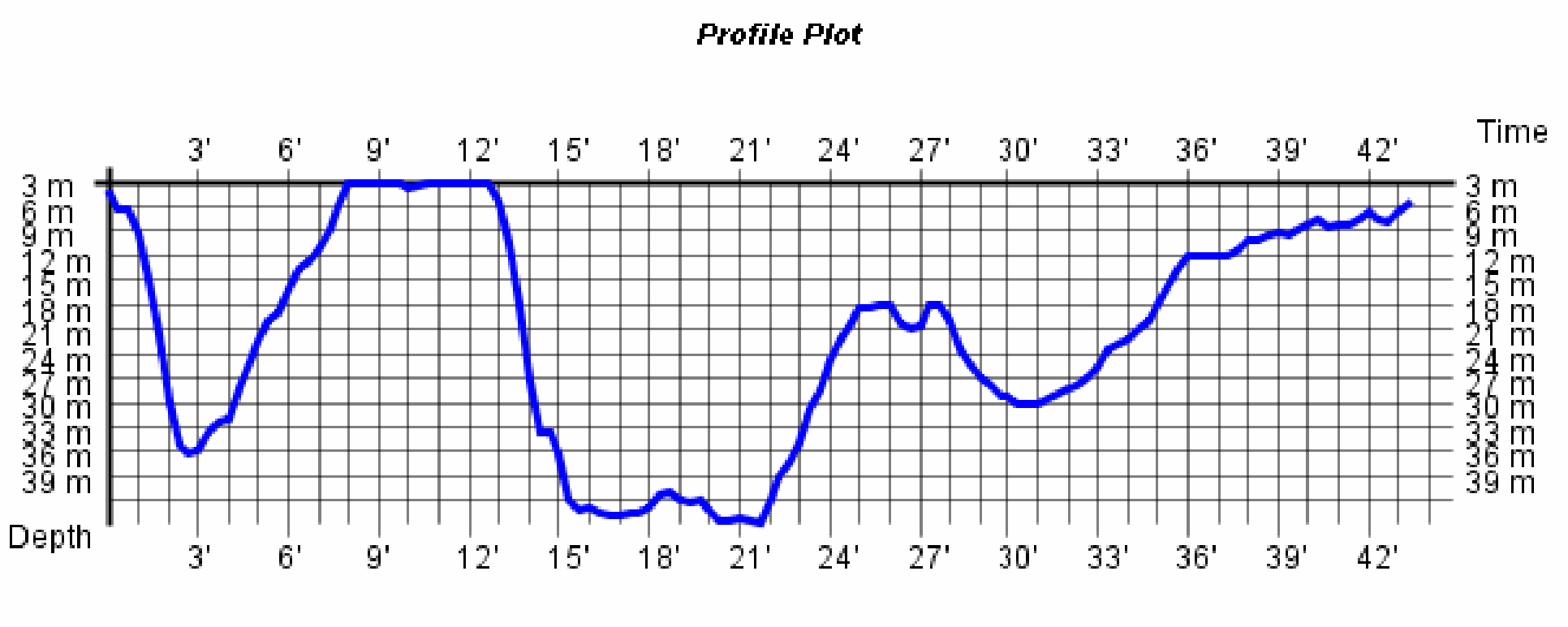
Document Planning

- Problem: Usually the output text can only communicate a small portion of the input data
  - » Which bits should be communicated?
  - » How should information be ordered and structured?

ScubaText: Imitate Corpus

- One approach to document planning is to analyse corpus texts (after aligning them to data), and manually infer content and structure rules.
- Scubatext example

Example Input



Input Segments

diveNo	segNo	itime	ivalue	ftime	fvalue
1460	1	0	1.3	60	6.3
1460	2	60	6.3	140	32.2
1460	3	140	32.2	480	0
1460	4	480	0	760	0
1460	5	760	0	920	38.9
1460	6	920	38.9	1300	41.6
1460	7	1300	41.6	1500	15.5
1460	8	1500	15.5	1860	27.2
1460	9	1860	27.2	2160	9.2
1460	10	2160	9.2	2600	2.7

Corresponding Corpus Text

- Your first ascent was a bit rapid; you ascended from 33m to the surface in 5 minutes, it would have been better if you had taken more time to make this ascent. You also did not stop at 5m, we recommend that anyone diving beneath 12m should stop for 3 minutes at 5m. Your second ascent was fine.

Align corpus text with data

Input: 1460 3 140 32.2 480 0

Output: (representation of)

Your first ascent was a bit rapid; you ascended from 33m to the surface in 5 minutes, it would have been better if you had taken more time to make this ascent

Input: 1460 10 2160 9.2 2600 2.7

Output: (representation of)

Your second ascent was fine

Possible content rules

- Describe segments that end (near) 0
  - » And that don't start at 0
  - » Also segment at end of dive
- Give additional info about such segments whose slope is too high
  - » Explain risk
  - » Say what should have happened

Possible ordering rules

- Break up dive into sections, where each section starts and ends at surface
- For each section, start with most important safety issue (or say dive was fine if no safety issue)
- Then add less important safety issues

More examples

- We've just looked at one example here!
- Need to repeat process for at least 20-30 examples, which cover spread of possible cases (including special cases)
- Merge rules and deal with conflicts
  - » Often causes by different corpus authors writing differently; may give priority to one particular author, and imitate his style

Content Determination

- The most important aspect of NLG!
  - » If we get content right, users may not be too fussed if language isn't perfect
  - » If we get content wrong, users will be unhappy even if language is perfect
- Also the most domain-dependent aspect
  - » Based on domain, user, tasks more than general knowledge about language

How Choose Content

- *Theoretical approach*: deep reasoning based on deep knowledge of user, task, context, etc
- *Pragmatic approach*: write schemas which try to imitate human-written texts in a corpus
- *Statistical approach*: Use learning tech to learn content rules from corpus

Theoretical Approach

- Deduce what the user needs to know, and communicate this
- Based on in-depth knowledge
  - » User (knowledge, task, etc)
  - » Context, domain, world
- Use AI reasoning engine
  - » AI Planner, plan recognition system

Theoretical Approach

- Not feasible in practice
  - » Lack knowledge about user
    - – Maybe has other jobs/tasks, eg helicopters
  - » Lack knowledge of context
    - – Eg, which supply boats are approaching
  - » Very hard to maintain knowledge base
    - – New users, boats, tasks, regulations…

Pragmatic Approach: Schema

- Templates, recipes, programs for text content
- Typically based on imitating patterns seen in human-written texts (corpus)
  - » Revised based on user feedback
- Specify structure as well as content

Schema Implementation

- Usually just written as code in Java or other standard programming languages
  - » Some special languages, but publicly available ones not very useful

Pseudocode example

```
Schema ScubaSchema
   for each ascent A in data set
      if ascent is too fast
            add unsafeAscentSchema(A)
     else
            add safeAscentSchema(A)
    set rhetorical relation
```

Pseudocode example

```
Schema unsafeAscentSchema(Ascent A)
   create DocumentElement
   add sentence “Your ascent was too
   fast”
   add sentence “You ascended from
         [A.startValue()] to the surface in
         [A.duration()] minutes”
return
```

Creating Schemas

- Creating schemas is an art, no solid methodology (yet)
- Williams and Reiter (2005)
  - » Create top-down, based on corpus texts
  - » First identify high-level structure of corpus doc
    - – Eg, each ascent described in temporal sequence
    - – Build schemas based on this
  - » Then create low-level schemas (rules)
    - – Eg, for describing a single ascent

Williams and Reiter

- Problems
  - » Corpus texts likely to be inconsistent
    - – Especially if several authors wrote texts
  - » Some cases not covered in the corpus
    - – Unusual cases, boundary cases
- Developer needs to use intuition for such cases
  - » check with experts, users!

Williams and Reiter

- Evaluation/Testing
  - » Essential!
  - » Developer-based:
    - – Compare to corpora
    - – Check boundary cases for reasonableness
  - » Expert-based: Ask experts to revise (post-edit) texts
  - » User-based: Ask users for comments

Statistical content det

- Statistical/learning techniques
  - » Parse corpus, align with source data, use machine learning algorithms to learn content selection rules/schemas/cases
    - – Barzilay and Lapata, 2005
    - – Kondadidi et al 2013
- Worth considering if **large corpora** available

Reuters approach

- http://www.aclweb.org/anthology/P/P13/P13-1138.pdf
- Training: Parse corpus of old newswire articles
  - » Turn sentences, phrases into templates
  - » Cluster templates into “conceptual unit”
  - » Train model which chooses template for 1st, 2nd, 3rd sentence
    - – Features such as “prior conceptual unit”
- Generation
  - » Eliminate templates/conceptual units which dont fit the input data
  - » Use model to choose templates for 1st, 2nd, etc sentences
  - » Fill in holes in template from data

Example

- Corpus: Mr Jones has a Bachelors in Finance from Harvard
- Template: [person] has [degree] in [subject] from [university]
- Conceptual unit: Person/degree/subject/university
  - » Also covers “[person] was awarded a [degree] in [subject] from [university]”

Example

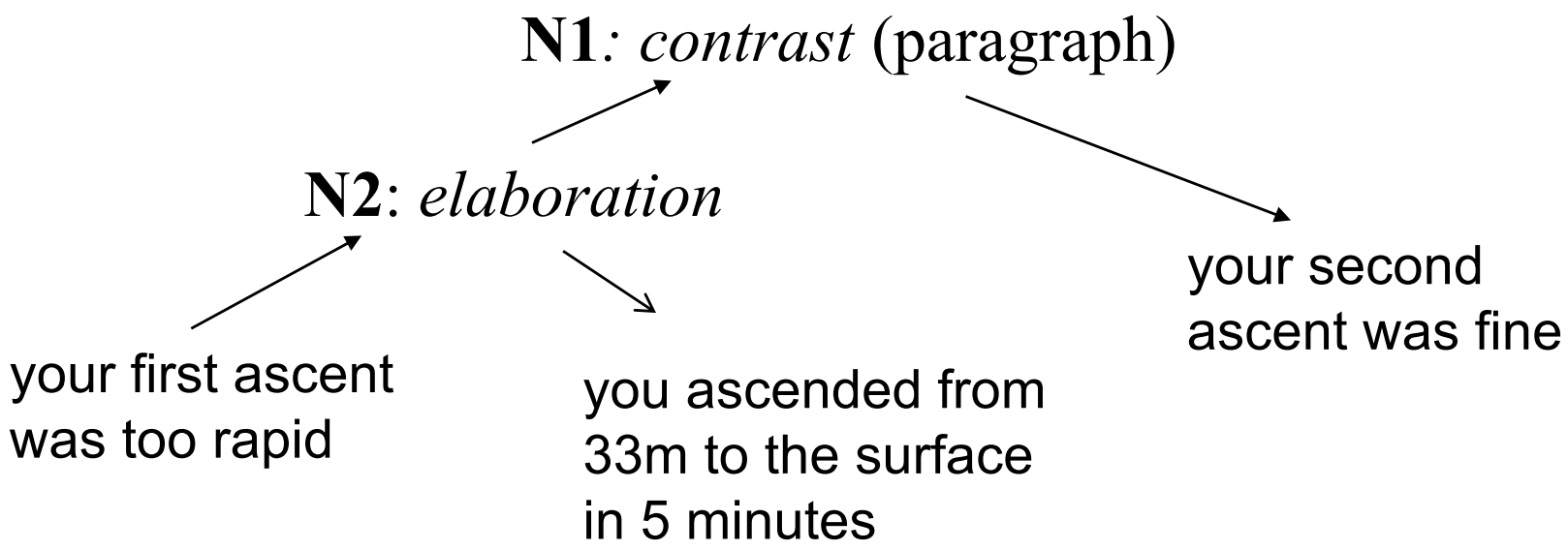
- Data includes tuple <Miss Smith, MSc, Computing, Aberdeen>
- Ranking module chooses previous template as top candidate for sentence 2
- Include sentence “Miss Smith has an MSc in Computing from Aberdeen”
  - » Second sentence in article

Text Structure

- DP chooses a text structure
  - » Tree, leaves represent sentences, phrases
    - – For now, assume leaves are strings
    - – Look at other representations next week
  - » Internal nodes group leaves, lower nodes
    - – Can mark sentences, paragraphs, etc breaks
    - – Can express rhetorical relations between nodes

Example tree

- Your first ascent was too rapid, you ascended from 33m to the surface in 5 minutes. However, your second ascent was fine.



Rhetorical Rel

- Shows how messages relate to each other
- RR can be expressed via cue phrases.
  - » Best cue phrase for RR depends on context
    - – Your first ascent was a bit rapid. **However**, your second ascent was fine.
    - – Your first ascent was a bit rapid, **but** your second ascent was fine.
  - » Also readers like cue phrases to be varied, not same one used again and again
    - – Eg, don't overuse "for example"
  - » Hence better to specify abstract RR in text spec

Common Rhetorical Rels

- CONCESSION (*although, despite)*
- CONTRAST (*but, however*)
- ELABORATION (usually no cue)
- EXAMPLE (*for example, for instance*)
- REASON (*because, since*)
- SEQUENCE (*and, also*)
- Research community does not agree
  - » Many different sets of rhetorical rels proposed

Advanced: Narrative

- We often want text to be a story or narrative
  - » Not a collection of “bullet points”
- Rhetorical relations help
  - » Link messages together
- Research topic

Advanced: User-Adaptation

- Texts should depend on
  - » User's personality
  - » User's domain knowledge (how much do we need to explain)
  - » User's vocabulary (can we use technical terms in the text)
  - » User's task (what does he need to know)
- Hard to get this information…

Personality and Perspectives

- Text can communicate perspectives, "spin", as well as raw data. Eg,
  - » Smoking is killing you
  - » If you keep on smoking, your health may keep on getting worse
  - » If you stop smoking, your health is likely to improve
  - » If you stop smoking, you'll feel better

Perspectives

- How to choose between these?
- Depends on personality of reader
  - » Some people react better to positive messages, others to negative messages
  - » Some react better to short direct messages, others want these weakened (“may”, “is likely to”)
  - » Hard to predict…

Conclusion

- Content determination is the first and most important aspect of NLG
  - » What information should we communicate?
- Mostly based on imitating what is observed in human-written texts
  - » Using schemas, written in Java
- Also decide on structure
  - » Tree structure, rhetorical relations