Natural Language Generation for News Automation

Week 1: introduction

Mark Granroth-Wilding

NLG for News Automation

Mark Granroth-Wilding:
mark.granroth-wilding@helsinki.fi

- Reading group / seminar
- 7 weeks
- Subject matter
  - Basic NLG grounding
  - Practical NLG systems: focus on news applications
  - NLG literature relating to news automation
  - Technical perspective

Today

- Introduction to traditional NLG (me)
- Allocation of topics to participants

Next week

- Short overviews
  - Each participant: what they plan to present
  - From initial look at literature
- Discuss what we’re interested in
- Journalists’ view

Topics

- Week 3: Goals and architecture of algorithmic journalism
  - What algorithmic journalism can be (or already is) useful for
- Week 4: NLG pipeline for weather reporting
  - Traditional NLG, more detail on specific, much-studied example
- Week 5 & 6: Statistical NLG
  - More up-to-date techniques, text-to-text generation, neural generation
- Week 7: Interaction with journalists
  - How can journalists interact with generation? Review of existing approaches

- ❖ More detail on course webpage (http://courses.helsinki.fi)
- ❖ Initial suggested reading

Aims

- Review NLG relevant to news automation
- Study previous work in news automation, technical perspective
- Discuss potential future directions

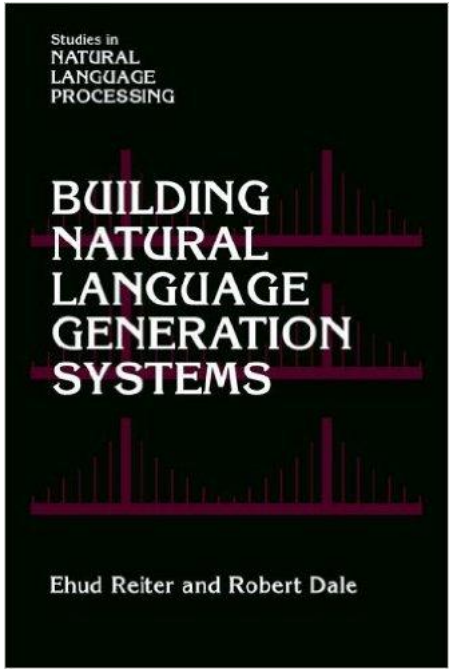
- First: basic grounding in traditional NLG
- Established architecture and sub-tasks involved

Your tasks

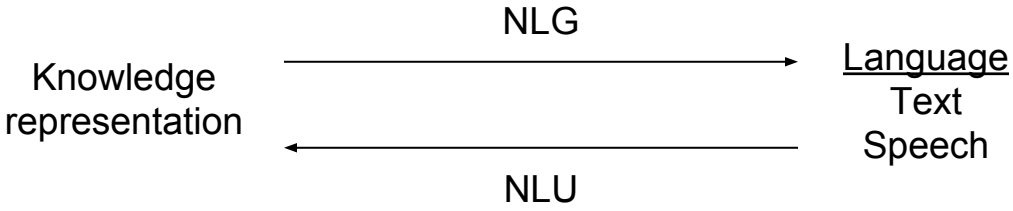
- Decide on a topic
  - After today's talk
- Short presentation of topic / sub-topic
  - Next week
- Presentation
  - Later sessions
- Abstract
  - Short report on what you presented (1-2 pages)
  - Connect your topic to other topics presented / discussed
- Reference list
  - Relevant literature you found
- Active participation in group
  - Some reading
  - Participation in discussions

Introduction to NLG

- Roughly follow:
  Building Natural Language Generation Systems
  (Reiter & Dale, 2000)
- Out of date, lots of work since
- Architecture is basis for much subsequent work
- Division into sub-tasks commonly used
- Later work: more statistical/machine-learning techniques
  - Maybe a bit today, more in later session



Natural Language Processing



Natural Language **Generation** & Natural Language **Understanding**

- Usually not same models
- Some model sharing possible (sometimes)
- *Concept-to-text* generation
  - Later also: *text-to-text* generation

NLG applications

- Teaching
- Marketing
- Entertainment (e.g. joke generation)
- Realisation for creative systems (e.g. storytelling)
- Interactive systems, dialogue
- News customization / automation: our focus

Evaluation

- Not a lot in R&D: not well established in 2000
- Better established methods since
- Difficult problem
- Very important topic!
- Won’t talk much in this group

Architecture of an NLG system

Normal to divide system into modules for sub-tasks

- Reuse components across systems
  - Some domain-/language-/application-specific
  - Others reusable, or can use standard tools
- Modify individual components as necessary
  - Problem or customisation can be addressed in responsible module
  - Easier adaptation across domains
- Human involvement in some modules
  - E.g. don’t want to automate whole process

Architecture of an NLG system

- General pipeline architecture well established
- Much current work focuses on subtasks as divided up here
- In practice, given system may divide up differently
  - May not need some parts
- Not the only way
  - Much recent work uses different divisions
  - Or no division at all – end-to-end generation

Inputs

- Communicative goal:
  - E.g. *inform, request, persuade, obtain information*
- Speaker chooses goal
  - Various formalisations
  - Often trivial: simply inform about data
  - Not true of journalism in general
  - May be true for a lot of new automation we want to do

Inputs

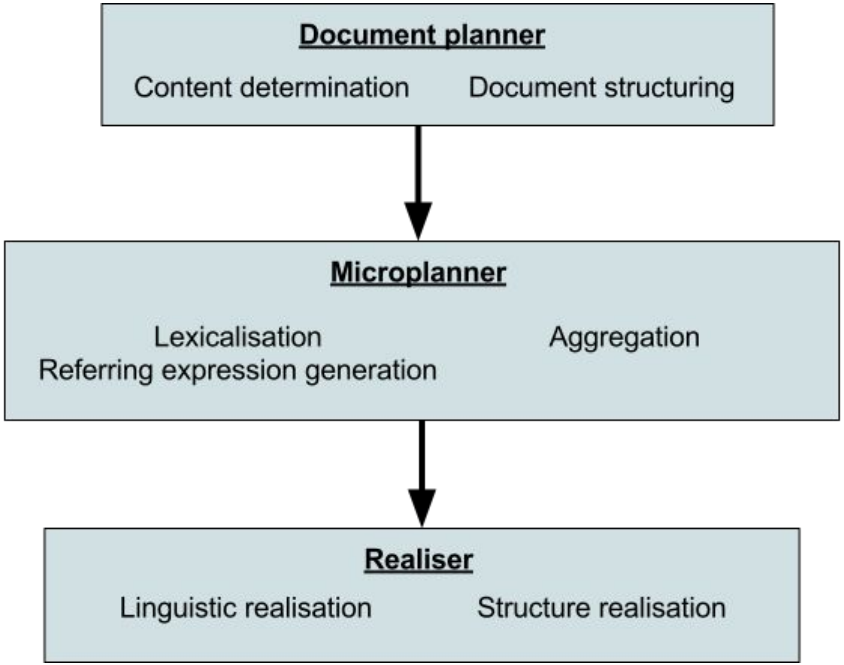
Input 4-tuple: `<k, c, u, d>`

- **K**nowledge source
  Domain knowledge, knowledge base, …
- **C**ommunicative goal
  What to convey. Type (template) + instance (specific information)
- **U**ser model
  Intended audience. Often hard-wired
- **D**iscourse history
  Said so far. Allows for reference, discourse, ...

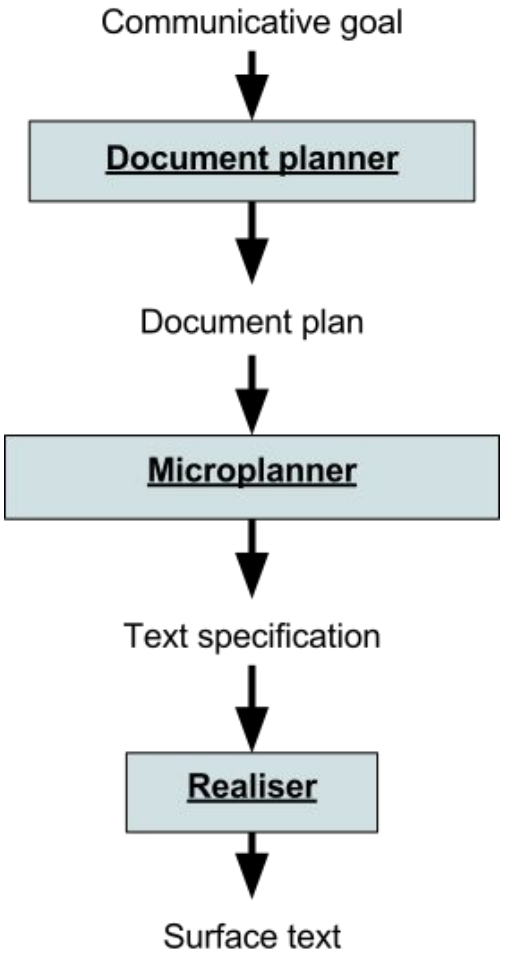
Output

- Typically text
- May include
  - structure
  - markup
  - other rich info as necessary

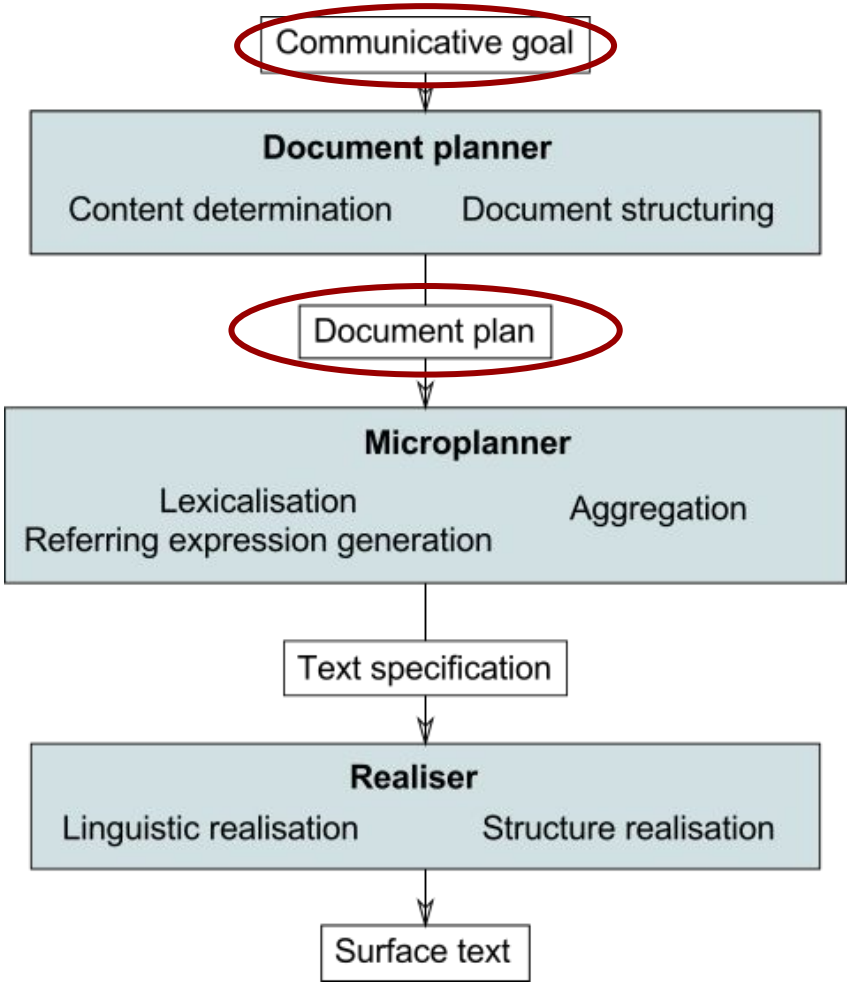
Pipeline


Document planner
Content determination
Document structuring
Microplanner
Lexicalisation
Referring expression generation
Aggregation
Realiser
Linguistic realisation
Structure realisation

Data structures in pipeline

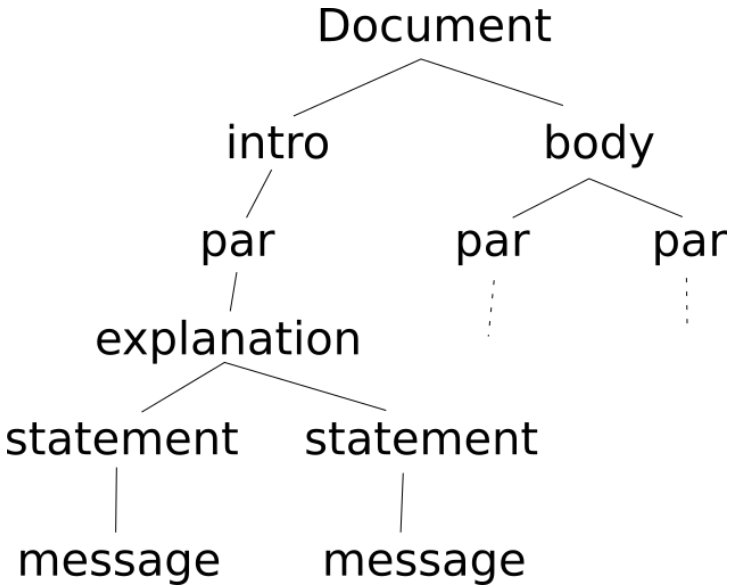


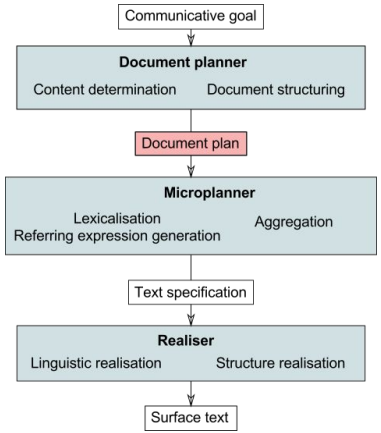
Traditional NLG pipeline



Document plan

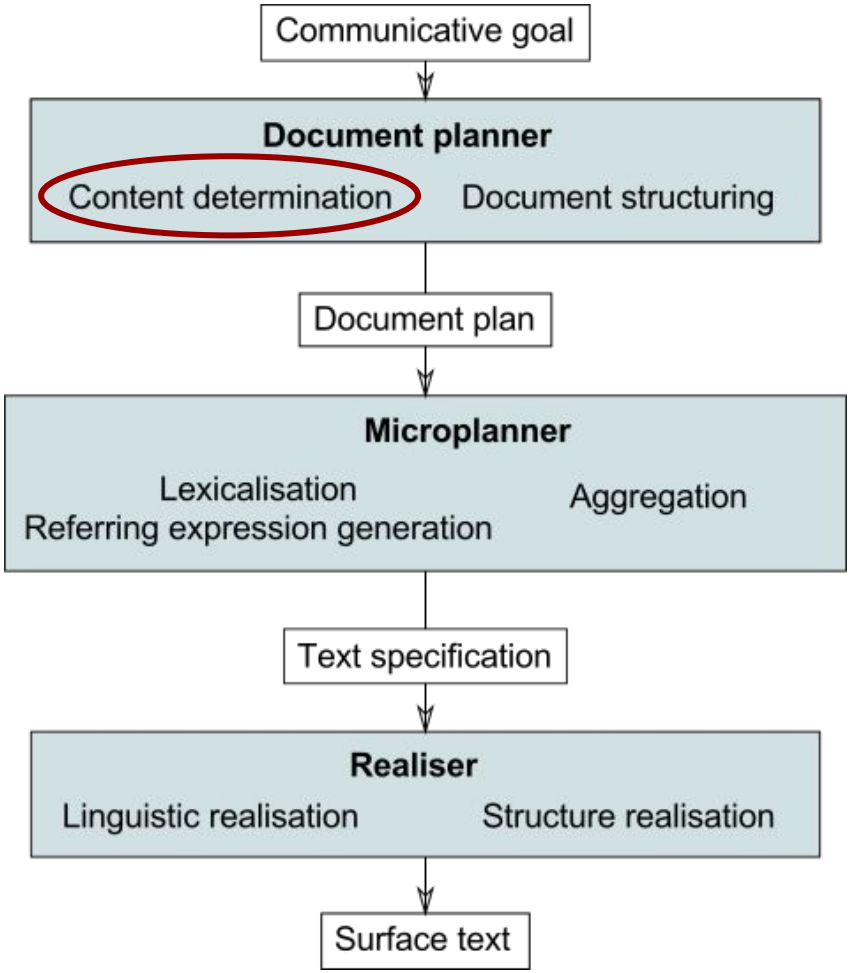
- Tree:
  - Internal nodes: structure, e.g. discourse relations
  - Leaves: *messages*
- Message:
  - Piece of info to express
  - Domain-specific concepts, entities, relations
  - Domain modelling: experts
  - May be later combined into one sentence (aggregation)
- Requires domain knowledge: not just data transformation





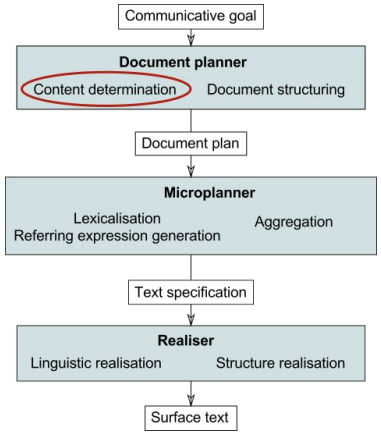
Document planner

Content determination



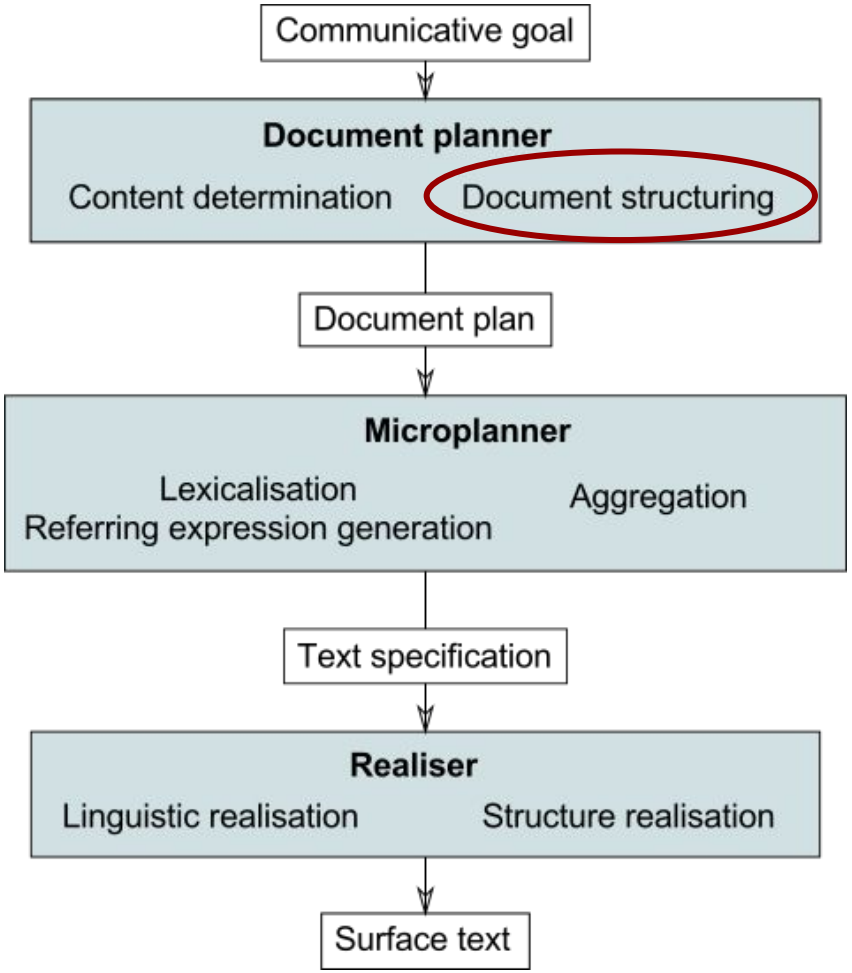
Content determination

- Decide what to communicate
  - May be specified in input
  - Often choose relevant info
- Depends on:
  - Communicative goals – different subsets of available data
  - User – e.g. novice vs. expert
  - Output constraints – e.g. limited space
  - Information available
- Very application dependent: can’t specify general rules.



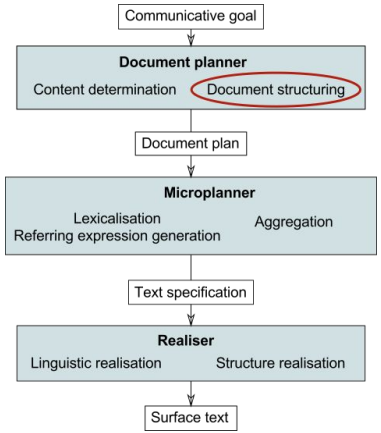
Document planner

Document structuring



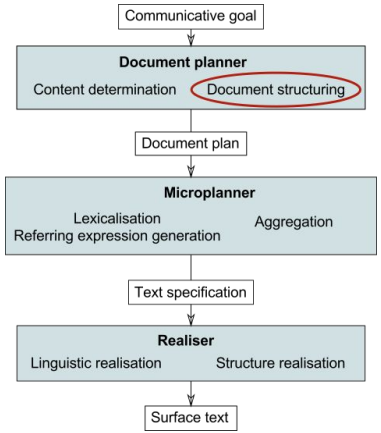
Document structuring

- Impose ordering / structure on information
- Closely related to **genre** of text
- Simplest: just ordering
- Often think of hierarchical structure. May define in terms of:
  - Subject matter / content
  - Discourse relations – explanation, elaboration, ...
- Sometimes signalled explicitly, sometimes left to reader
  - E.g. “therefore”
  - Rich output – markup, sections
- Also highly application-dependent

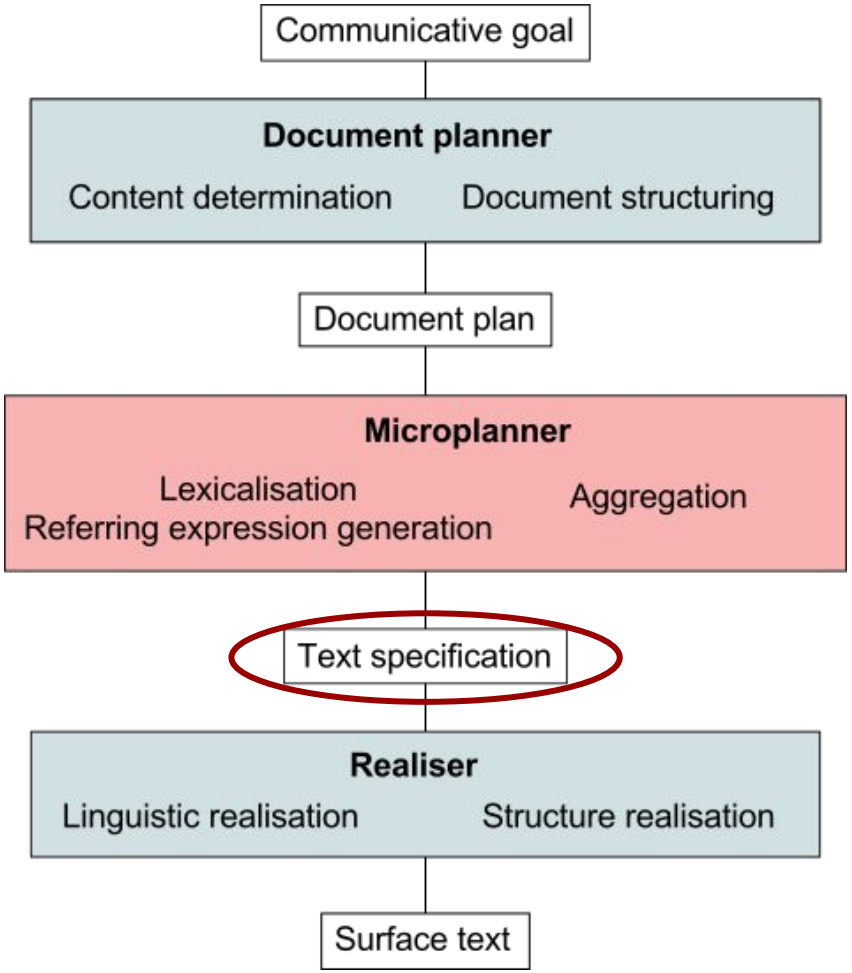


Discourse relations

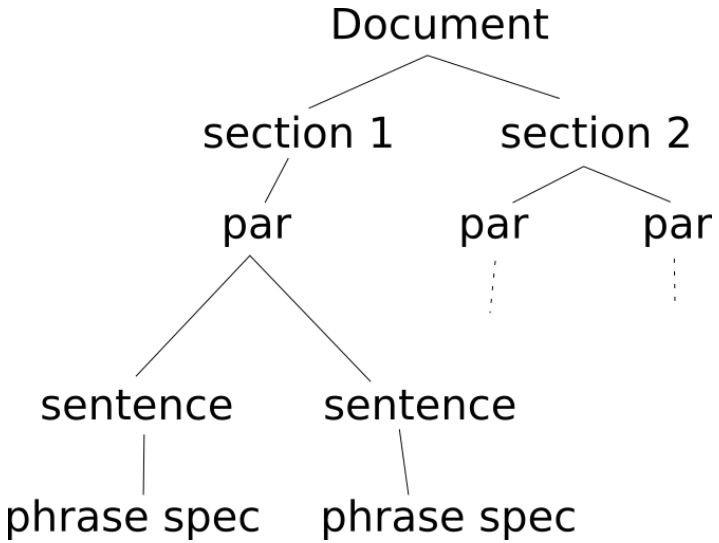
- Logical relationship between segments of discourse
- Examples:
  - Explanation:
    Mark switched off the light. It was time to go home.
  - Result:
    Mark switched off the light. The room was dark.
  - Narration:
    Mark switched off the light. He left the building and took a bus home.
- Various formal theories: **D**iscourse **R**elation **T**heory
- Hierarchical structure
- Sometimes explicitly marked:
  - *Mark switched off the light, **because** it was time to go home*



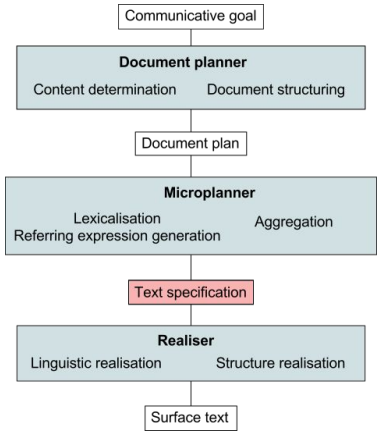
Text specification



Text specification

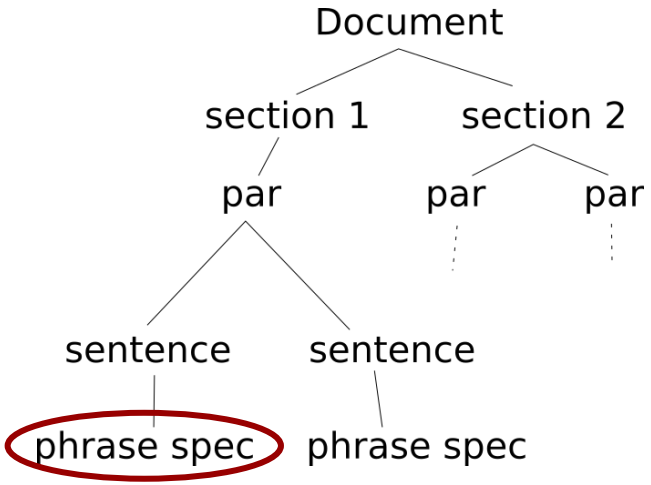


- All information required to realise the text
- Tree:
  - Internal nodes: text structure elements
  - Leaves: phrase specifications
- Structures like **discourse relations** gone
  - Some mapped onto text elements (e.g. *because*)
  - Some implicit
- Multiple **messages** may have been combined into **phrases**
- Ordering fully specified



Phrase specification

- Each corresponds to a **phrase**
  - Usually a **sentence**
- May be specified in various ways
  - Generated string
  - Fragment of canned text
  - Abstract syntactic structure
  - Lexicalised case frame

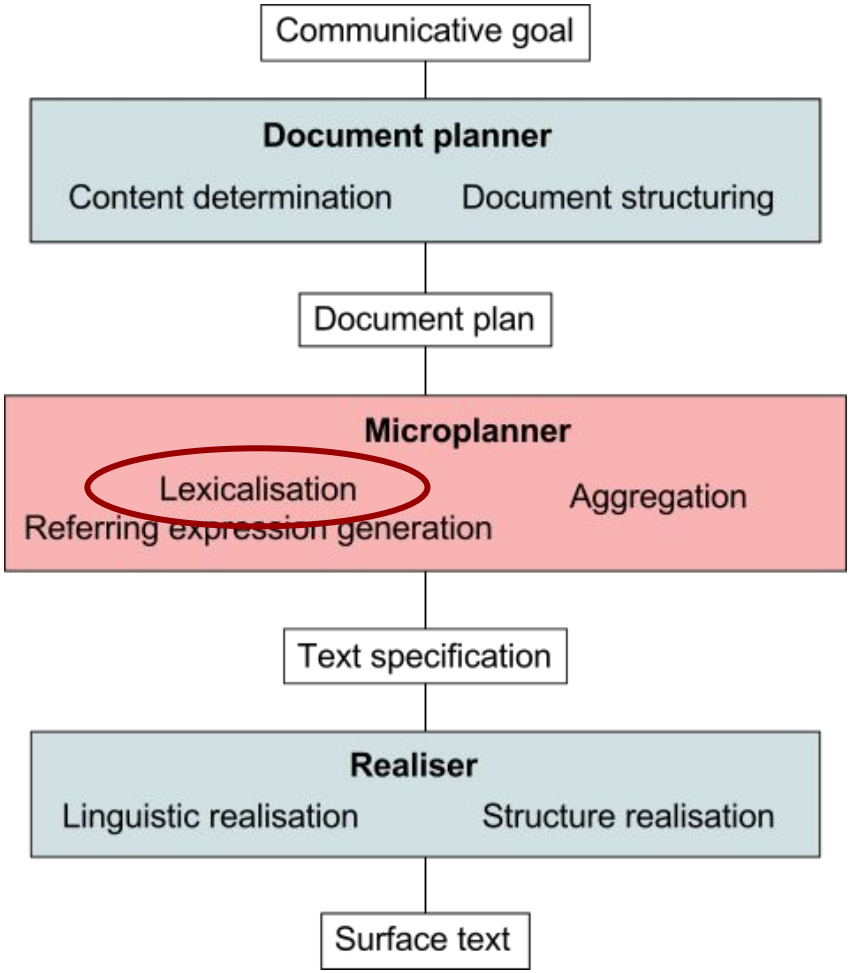


- ❖ Structure of sentence
- ❖ Uninflected content words
- ❖ Syntactic features for words

- ❖ More abstract structure
- ❖ Semantics roles
- ❖ Filled template

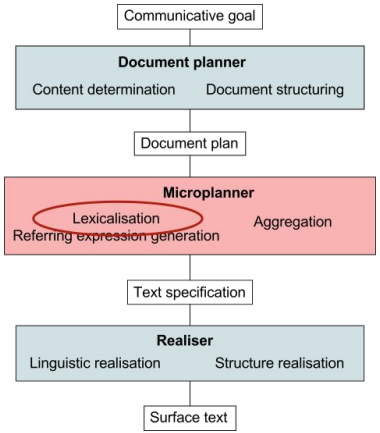
Microplanner

Lexicalisation



Lexicalisation

- Choose content words to express selected content
- May also involve choosing linguistic structures
- May depend on:
  - User – what sort of language is appropriate / effective
  - Pragmatic factors – e.g. what to emphasise
  - Communicative goals
- Multilingual systems
  - Some choices fairly language-independent
    E.g. broad-scale linguistic structures
  - Others obviously require different components for different languages
    E.g. picking content words



Referring expression generation

- How to refer to entities
- Initial reference
  - In practice, often heavily constrained by data
  - E.g. person name
  - Therefore, usually straightforward
- Subsequent reference
  - Abbreviate without ambiguity
  - E.g. use pronouns
  - Difficult task
  - Must take context into account
- Common approach:
  determine properties to be expressed, lexicalise

BlackRock Inc, the world's biggest asset manager, *reported better-than-expected quarterly profits on Friday,*

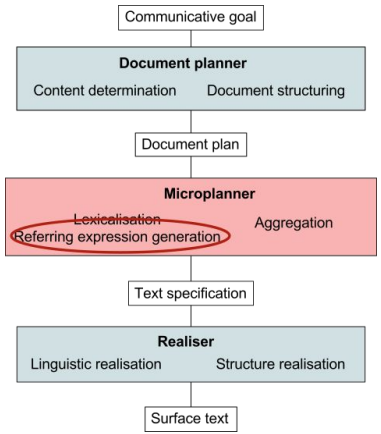
*as **it** clamped down on expenses.*

*Investors poured $88 billion into **the company**'s market-tracking index investments.*

?

?

- it
- the company
- BlackRock Inc
- BlackRock Inc, the world's biggest asset manager



Aggregation

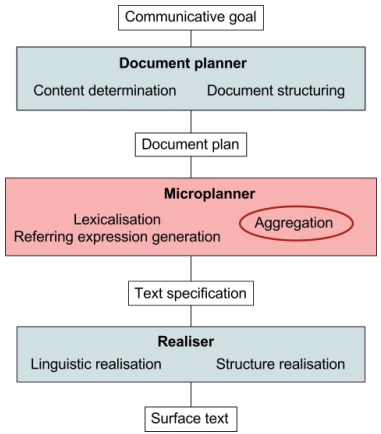
- Combine **messages** into **phrases**
- Message – one piece of information
- Build phrases to convey several pieces of information

Investors bought its index investments

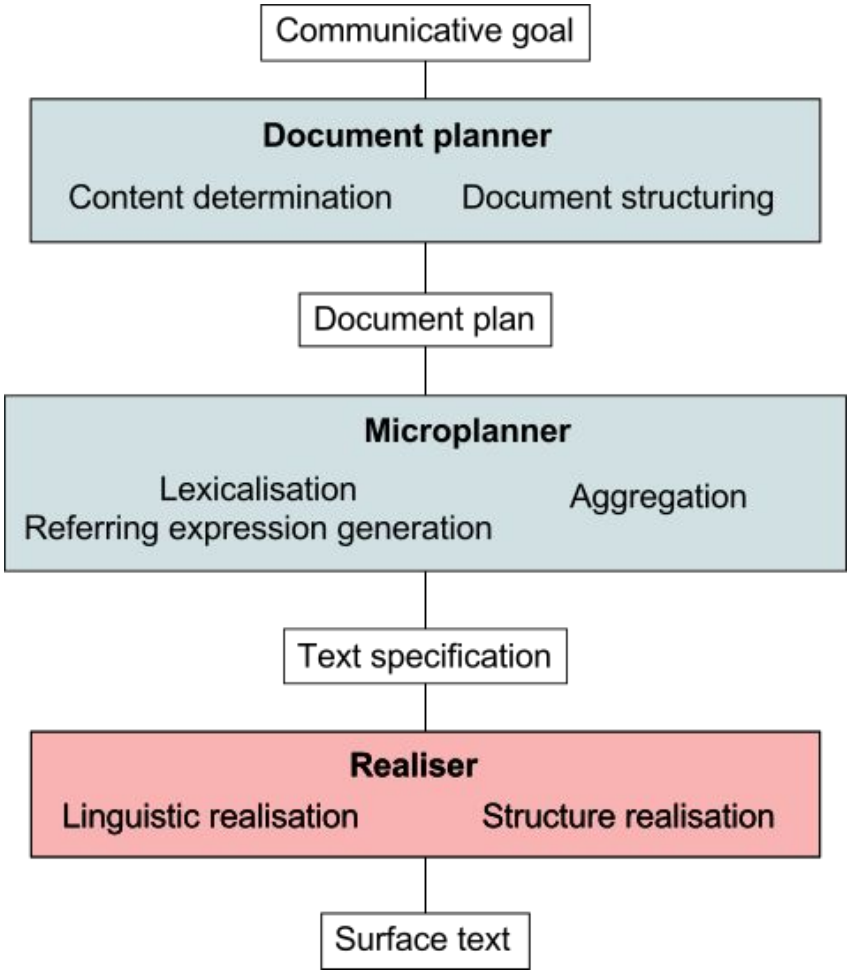
Investors bought its iShares exchange-traded funds

*Investors bought its **index investments** and **iShares exchange-traded funds**.*

- Interaction with lexicalisation
  - Some lexicalisations permit aggregations that others don't

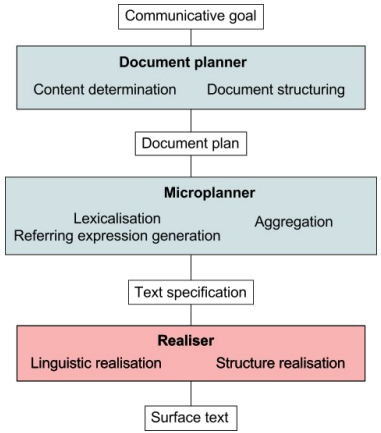


Realiser


Communicative goal
Document planner
Content determination
Document structuring
Document plan
Microplanner
Lexicalisation
Referring expression generation
Aggregation
Text specification
Realiser
Linguistic realisation
Structure realisation
Surface text

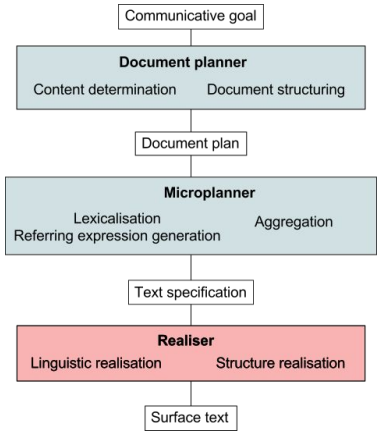
Linguistic realisation

- Produce text
- Traverse text spec & phrase specs
- Output from nodes
  - Trivial for some types: e.g. string
  - More complex for others: e.g. abstract syntactic structure
- More abstract representations better for multilingual systems
  - Leave language-specific choices till later in pipeline
- Well-established, public-domain systems exist



Surface realisation

- Convert document structure information into annotations
- E.g. markup, paragraph breaks, section headings



Future topics

- Week 3: Goals and architecture of algorithmic journalism
  - What algorithmic journalism can be (or already is) useful for
- Week 4: NLG pipeline for weather reporting
  - Traditional NLG, more detail on specific, much-studied example
- Week 5 & 6: Statistical NLG
  - More up-to-date techniques, text-to-text generation, neural generation
  - Week 5: Statistical approaches to components of pipeline
  - Week 6: Alternative architectures for statistical NLG
- Week 7: Interaction with journalists
  - How can journalists interact with generation? Review of existing approaches

Course information

http://courses.helsinki.fi

NATURAL LANGUAGE GENERATION FOR NEWS AUTOMATION

582767, Lecture Course, 2 cr, Mark Granroth-Wilding, Hannu Toivonen, 17.01.2017 - 28.02.2017 Teaching language en

TRADITIONAL NLG PIPELINE AND NLG FOR NEWS

We cover a grounding in the traditional stages of the NLG pipeline, through to recent advances, with a focus on news.

This course will begin with an overview of the typical technical architecture of a Natural Language Generation (NLG) system and some of the techniques used at different stages of the pipeline. In the following sessions, each presented by a different participant, we will look more closely at recent advances in different technologies that form parts of the traditional pipeline, as well as more recent alternatives to the pipeline, including NLG using neural networks.

Natural Language Generation for News Automation

Group 99 (queue - if the other groups are full or the times are not suitable)

Initial reading

- TIMETABLE
- MATERIAL
- TOPIC: GOALS AND ARCHITECTURE OF ALGORITHMIC JOURNALISM
- TOPIC: NLG PIPELINE FOR WEATHER REPORTING
- TOPIC: STATISTICAL NLG, PART I: STATISTICAL APPROACHES TO PIPELINE COMPONENTS
- TOPIC: STATISTICAL NLG, PART II: ALTERNATIVE ARCHITECTURES
- TOPIC: INTERACTION WITH JOURNALISTS
- CONDUCT OF THE COURSE