ESSLLI

Incremental Speech and Language Processing for Interactive Systems

Timo Baumann, Arne Köhn,
Universität Hamburg, Informatics Department
Natural Language Systems Division
{baumann,koehn}@informatik.uni-hamburg.de

Contents of the Course

- Monday:
  - introduction, major features of incremental processing
- Tuesday:
  - incremental processing for sequence problems
- Wednesday:
  - incremental processing for structured problems
- Thursday:
  - generating output based on structured and partial input
- today:
  - placement of examples, classification, wrap-up and outlook

Branches of NLP that we've mentioned

Branches of NLP that we've mentioned

speech recognition

Branches of NLP that we've mentioned

speech recognition

grapheme-phoneme conversion

Branches of NLP that we've mentioned

part-of-speech tagging

speech recognition

grapheme-phoneme conversion

Branches of NLP that we've mentioned

syntactic parsing

part-of-speech tagging

speech recognition

grapheme-phoneme conversion

Branches of NLP that we've mentioned

syntactic parsing

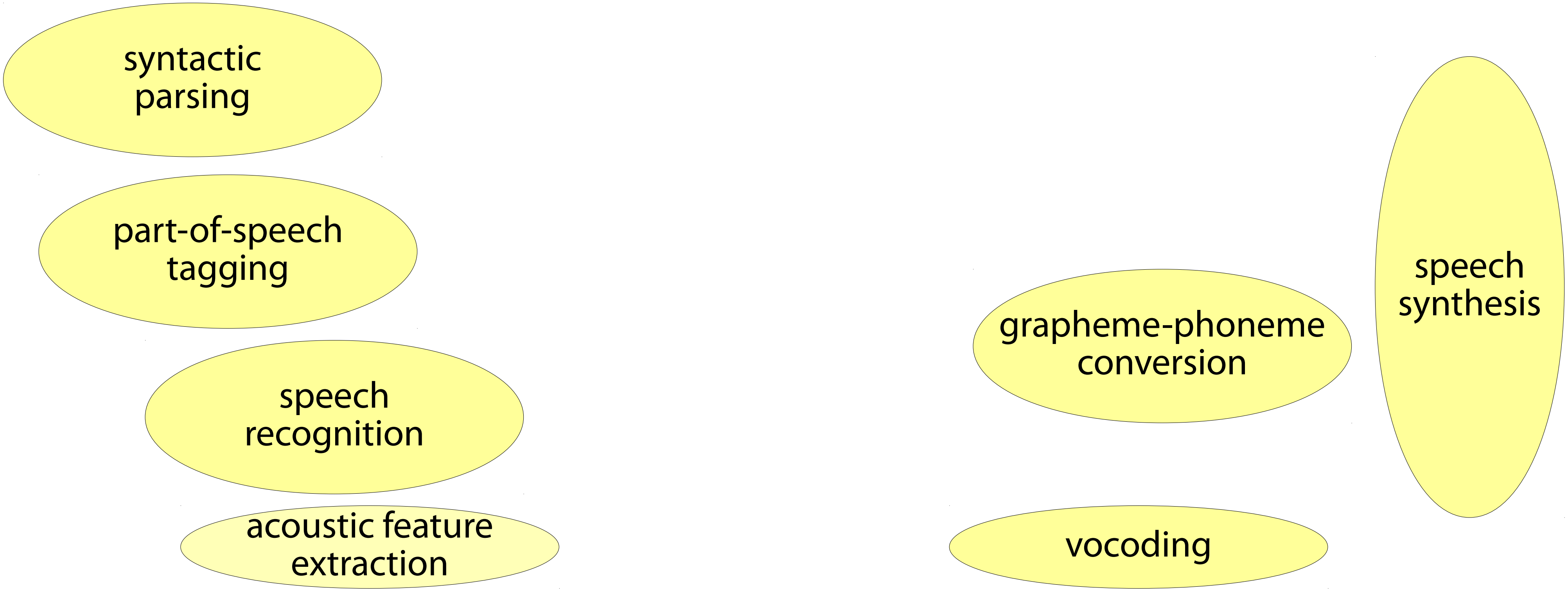
part-of-speech tagging

speech recognition

grapheme-phoneme conversion

speech synthesis

Branches of NLP that we've mentioned



syntactic parsing

part-of-speech tagging

speech recognition

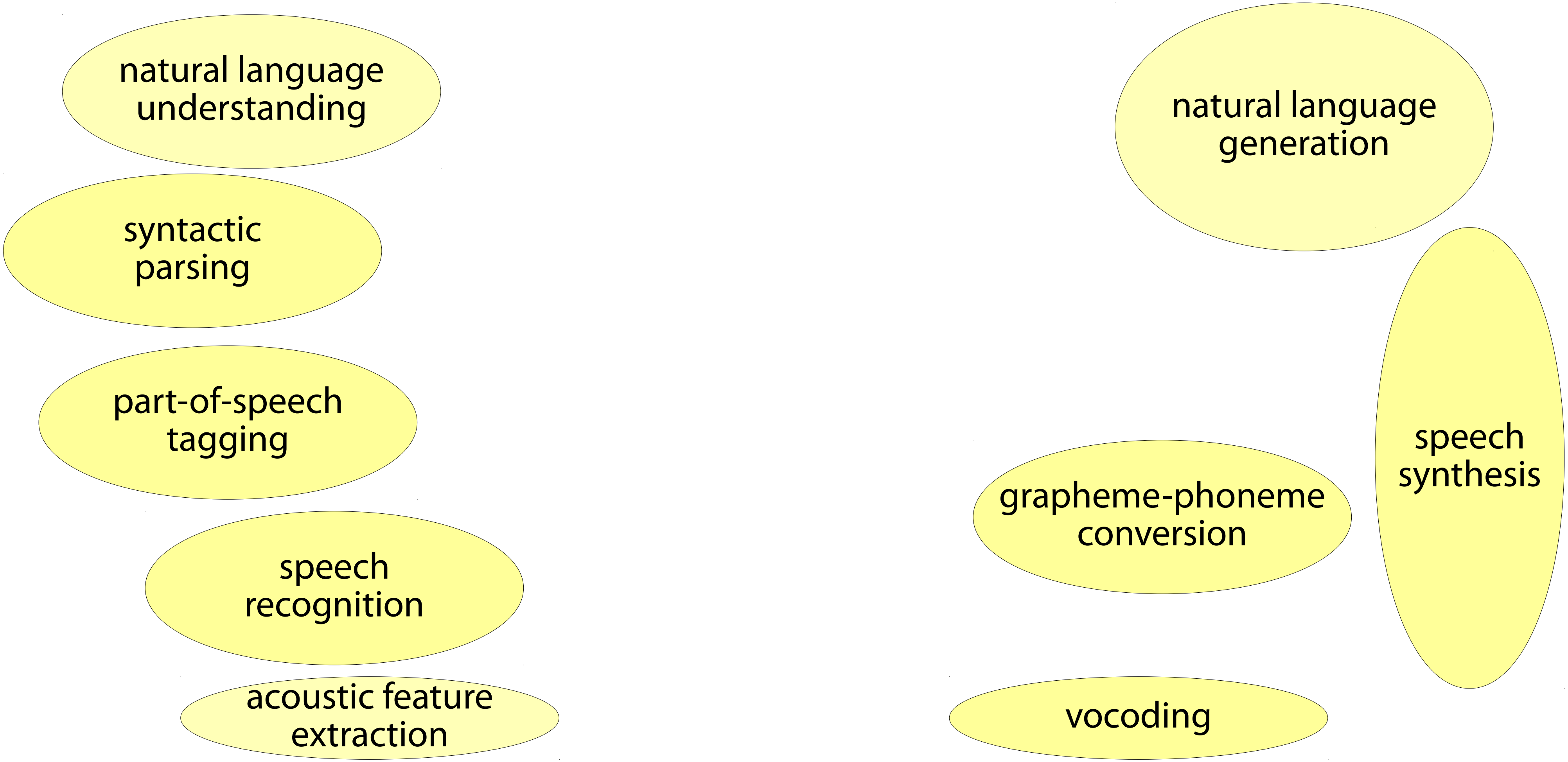
acoustic feature extraction

grapheme-phoneme conversion

vocoding

speech synthesis

Branches of NLP that we've mentioned



natural language understanding

syntactic parsing

part-of-speech tagging

speech recognition

acoustic feature extraction

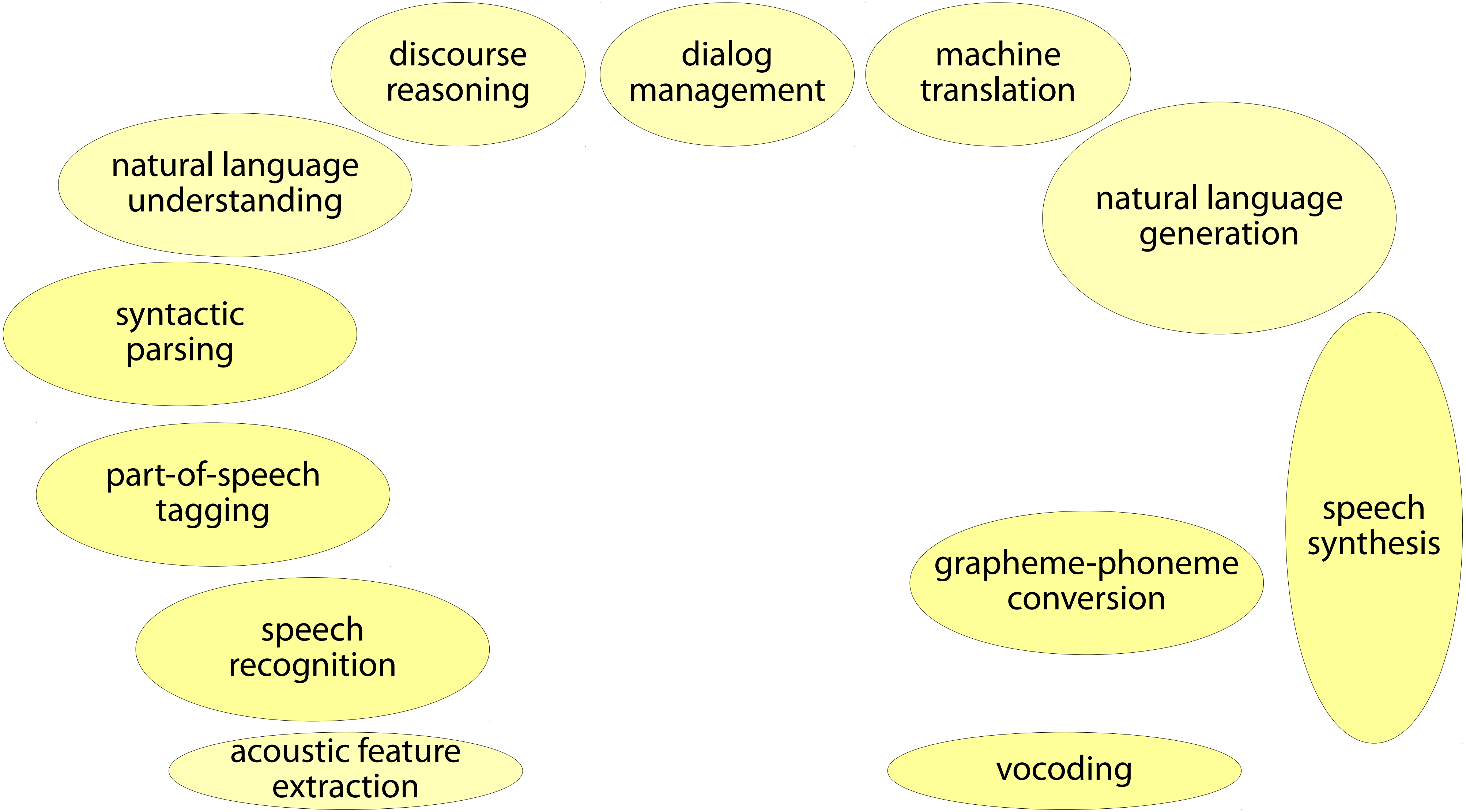
natural language generation

speech synthesis

grapheme-phoneme conversion

vocoding

Branches of NLP that we've mentioned



discourse reasoning

dialog management

machine translation

natural language understanding

natural language generation

syntactic parsing

part-of-speech tagging

speech synthesis

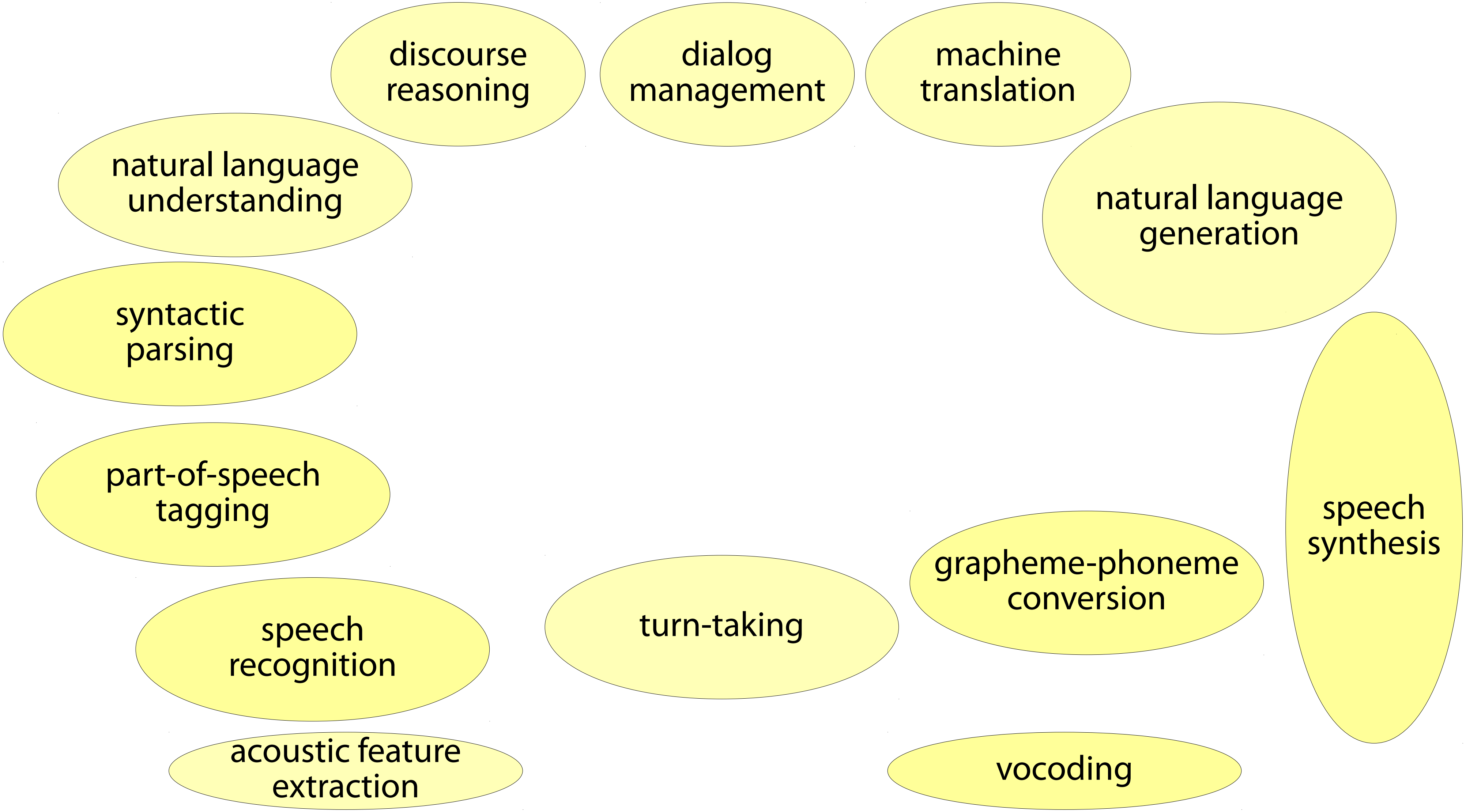
grapheme-phoneme conversion

speech recognition

acoustic feature extraction

vocoding

Branches of NLP that we've mentioned



discourse reasoning

dialog management

machine translation

natural language generation

speech synthesis

grapheme-phoneme conversion

vocoding

turn-taking

acoustic feature extraction

speech recognition

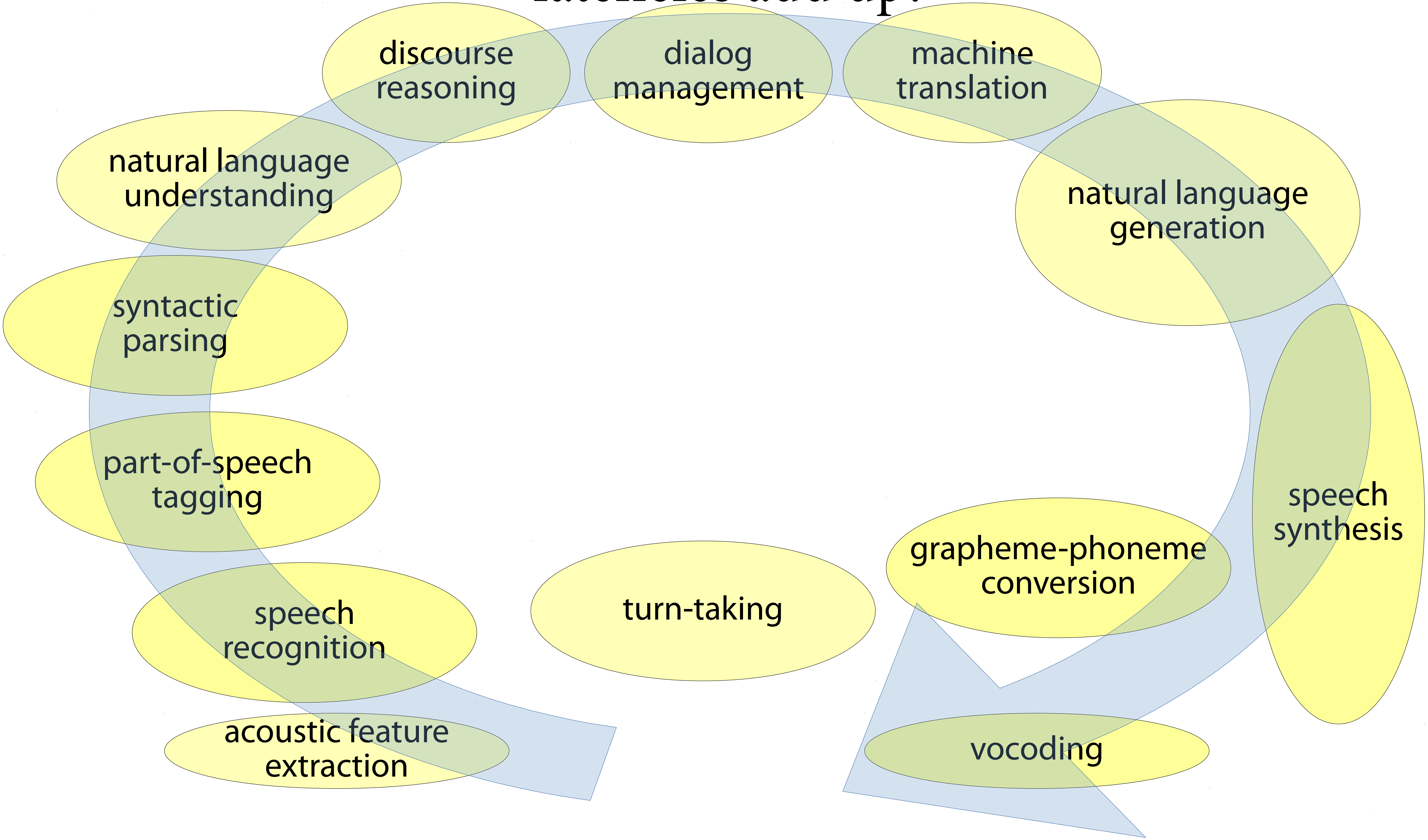
part-of-speech tagging

syntactic parsing

natural language understanding

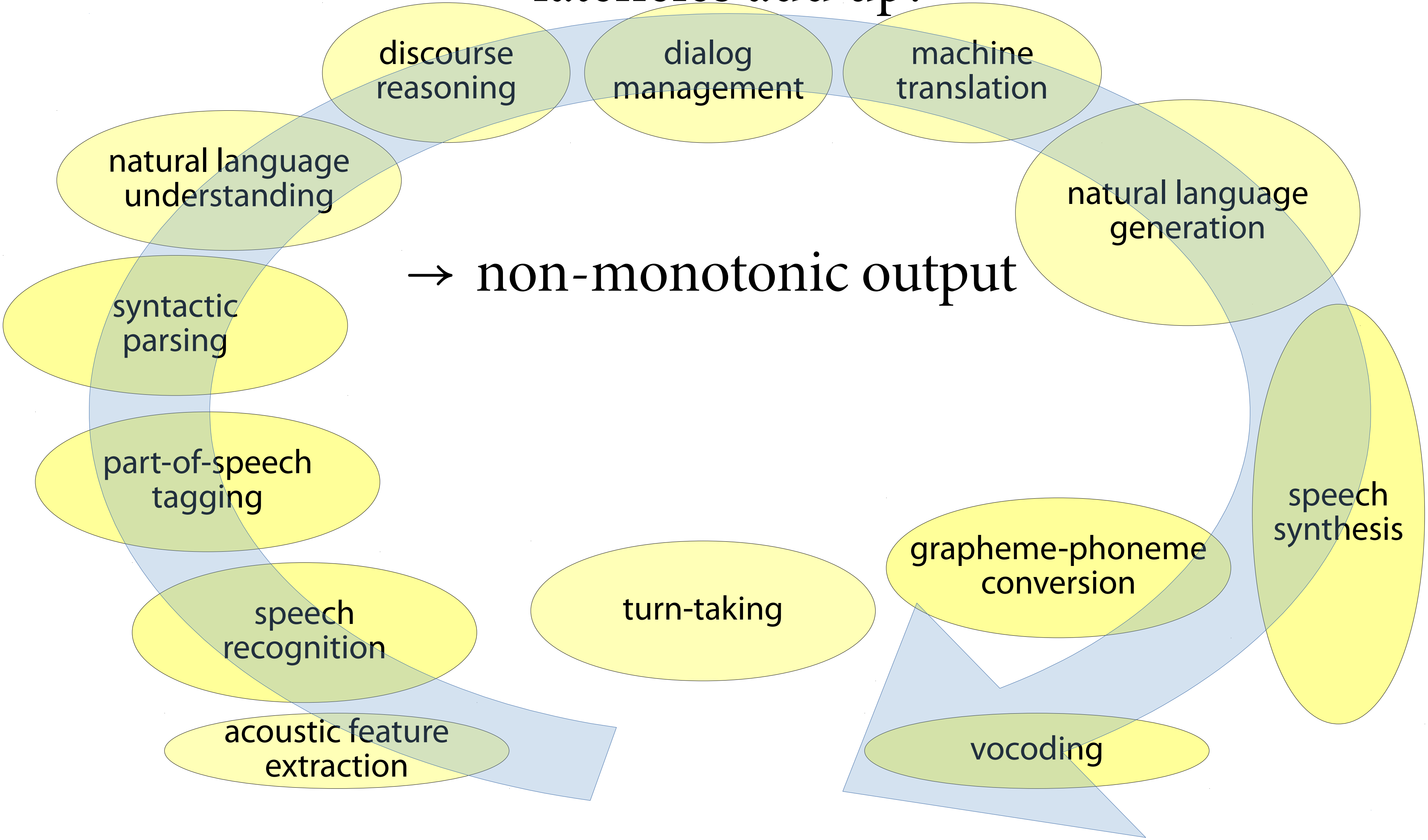
Branches of NLP that we've mentioned

latencies add up!



Branches of NLP that we've mentioned

latencies add up!



acoustic feature extraction

speech recognition

part-of-speech tagging

syntactic parsing

natural language understanding

discourse reasoning

dialog management

machine translation

natural language generation

speech synthesis

grapheme-phoneme conversion

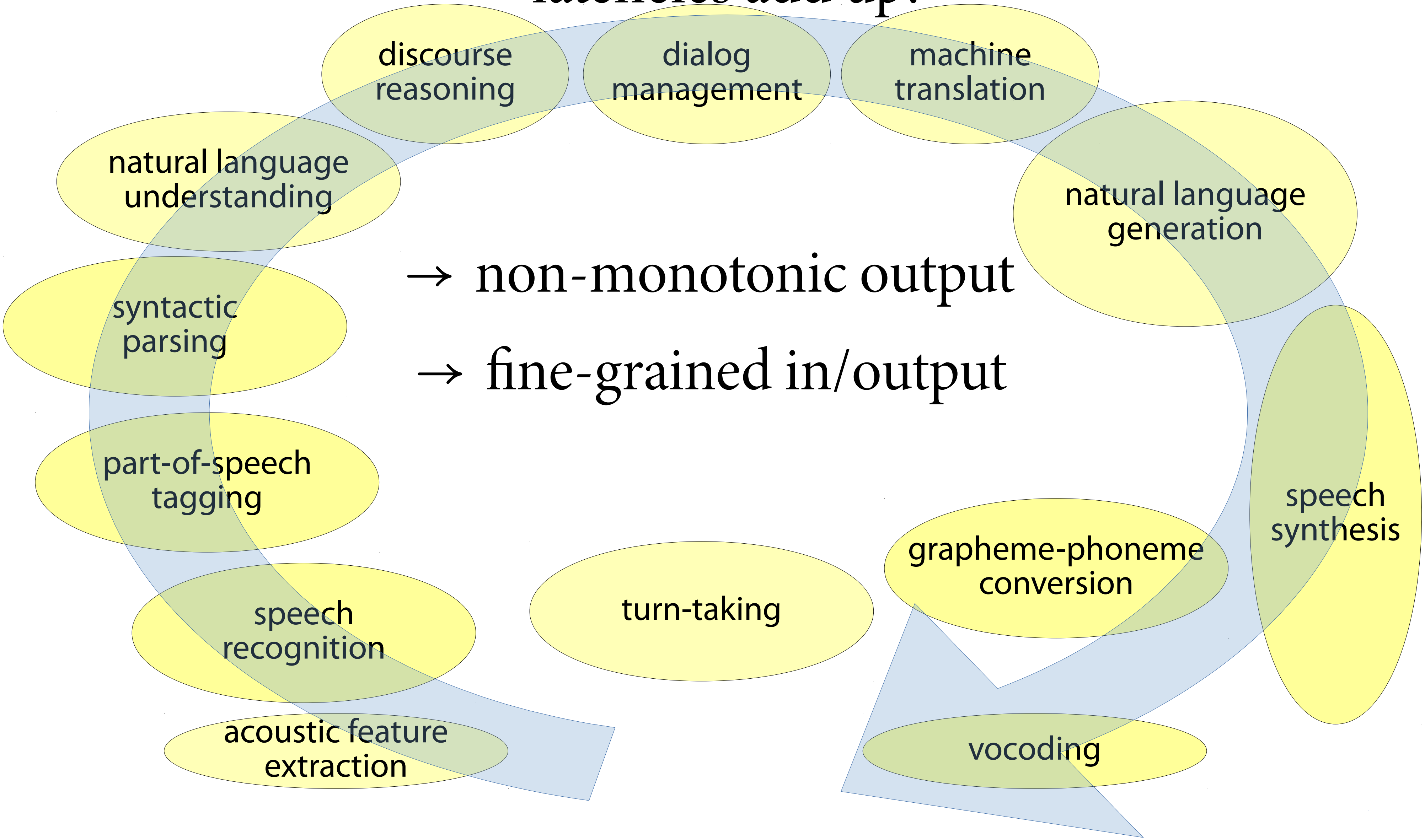
vocoding

turn-taking

→ non-monotonic output

Branches of NLP that we've mentioned

latencies add up!



→ non-monotonic output

→ fine-grained in/output

Granularity

- Some problems are trivially incremental at some level of granularity
  - Grapheme to Phoneme: words as basic unit
  - Syntax: sentences as basic unit
- More fine-grained processing
  - more room for error
  - room for improvements
- Usually pays off

Input-Output relation

- 1:1 POS tagging
- n:1 frame semantics
- 1:n language generation
- n:m Grapheme-to-Phoneme conversion

Incremental Processing Types

Classifying Incremental Processors

Non-monotonicity possible

Input Output	Sequence	Structured
Discrete	PoS tagging	Parsing
Continuous	Speech recognition	?

Only Monotonic output

Input Output	Sequence	Structured
Discrete	Speech synthesis	Natural language generation
Continuous	?	?

Discrete to Sequence

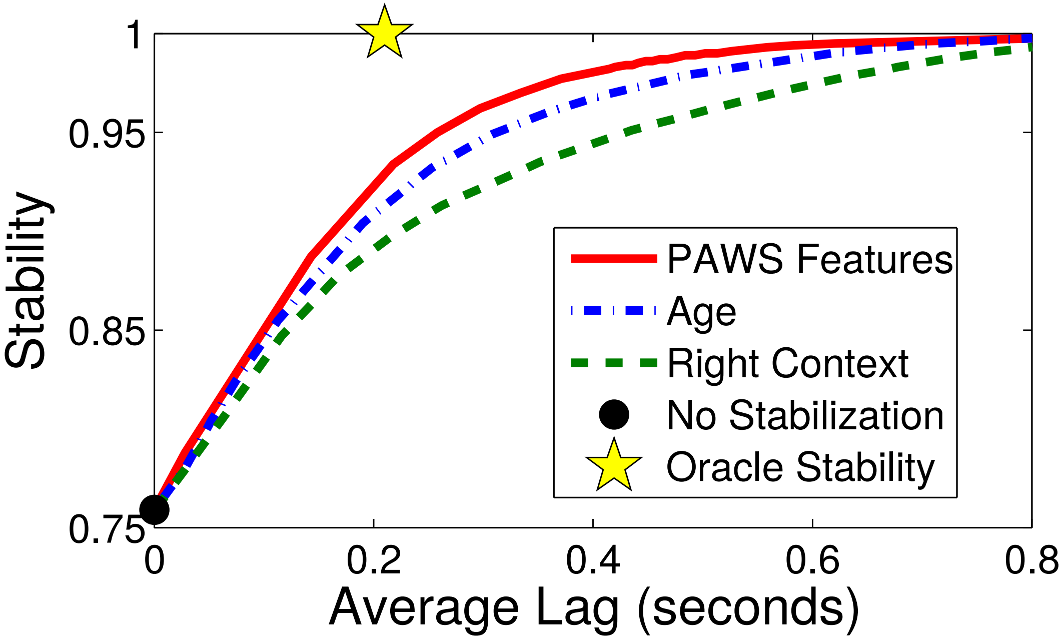
- The easiest one
- Monotonic Delay output until enough context available
  - Fixed number
  - Dynamic based on estimates
  - If everyone does that, you degrate to non-incrementality!
- Non-monotonic output
  - Maybe guarantee monotonicity for output in the past
  - Give stability estimates
- Multiple Alternatives
  - Pass the problem on to downstream applications

e.g. G2P,
PoS tagging

Continuous to Sequence

- Output can be created all the time
  - creates lots of noise, but is quickest
- Delay based on the age of hypothesis (or smarter)
  - estimate trade-off curve
  - pick operating point

e.g.
Speech recognition



Discrete to Structured

- Need to devise intermediate structure format
- Maximize information
  - predict what's predictable
  - High commitment cost if monotonic guarantee
- Adapt training objective
  - Adapt data and/or
  - Adapt your algorithm

e.g. parsing

Incremental Output Generation

- Output is inherently monotonic
- [Suboptimal output] + [Incremental] > [Optimal output]
  - People might prefer your output just because it's faster
- Be slightly suboptimal at the start
  - Change word ordering etc.
  - Better than crashing at the end
  - e.g. use re-inforcement learning for optimization

e.g.
Translation,
speech synthesis

Algorithms

Incremental Algorithms

- Extend monotonically left-to-right
- Use beam
- Output best item in beam at each time point
  - Results in non-monotonic output
- Much harder for structured prediction
- ?How to do this for
  - structured input?
  - non-monotonic input?

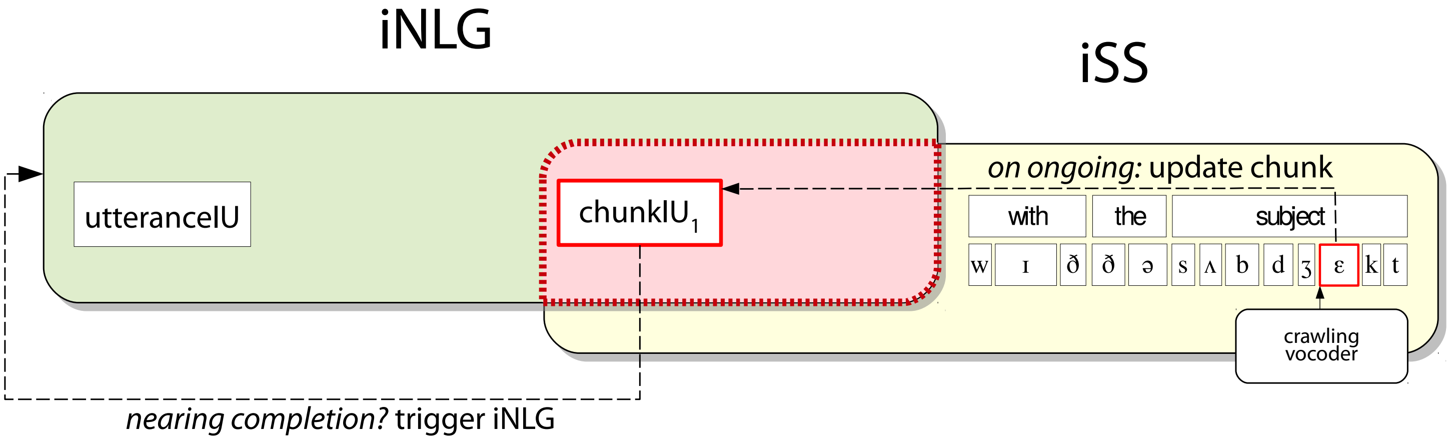
Restart-incremental

- Often the first and easiest step
- Uses more CPU time
- No monotonicity guarantees
- Monotonicity usually not even enforceable
  - for visible output non-monotonicity is limited
- Non-monotonic input is no problem

Incremental Units Model

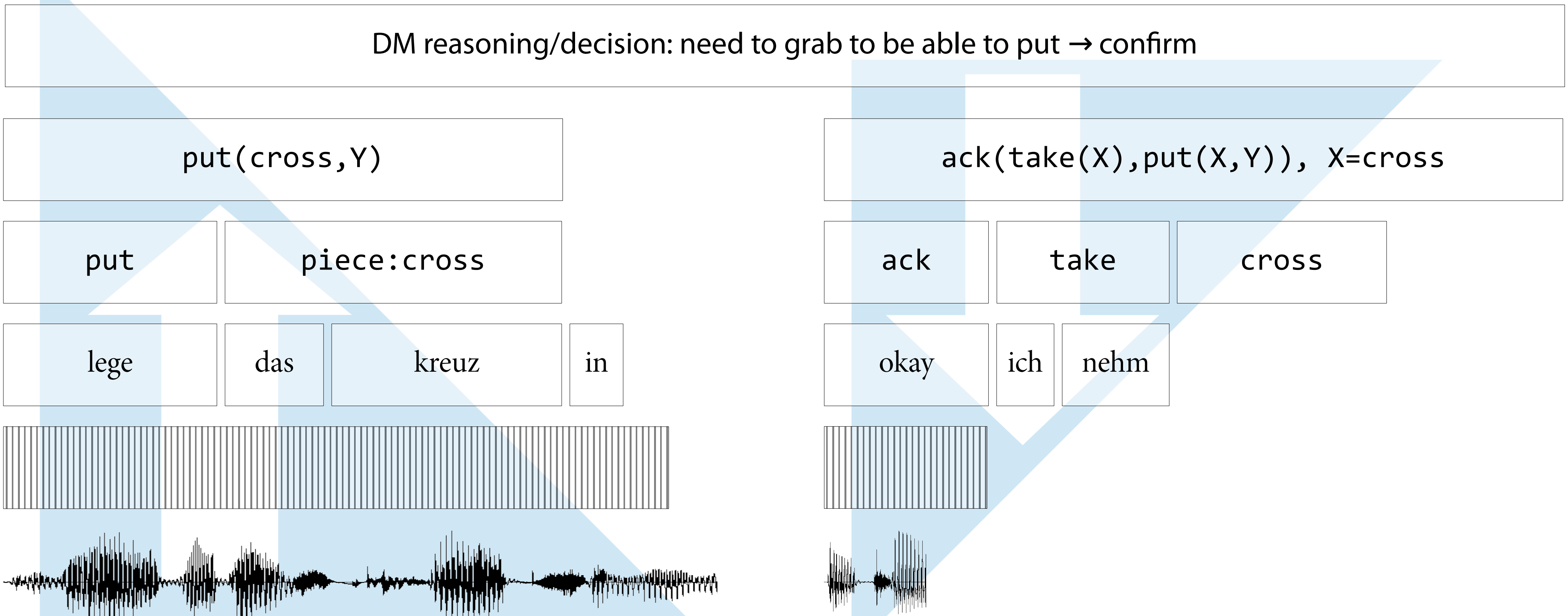
Incremental Units Model

- incrementality is mostly fun in end-to-end systems
  - modular systems in practice



Incremental Units Model

- incrementality is mostly fun in end-to-end systems
  - modular systems in practice
  - many problems require grounded/non-modular input
    - aligning gestures with speech requires timed words (not just words)
    - alignment of referring expressions



DM reasoning/decision: need to grab to be able to put → confirm

put(cross,Y)

ack(take(X),put(X,Y)), X=cross

put

piece:cross

ack

take

cross

lege

das

kreuz

in

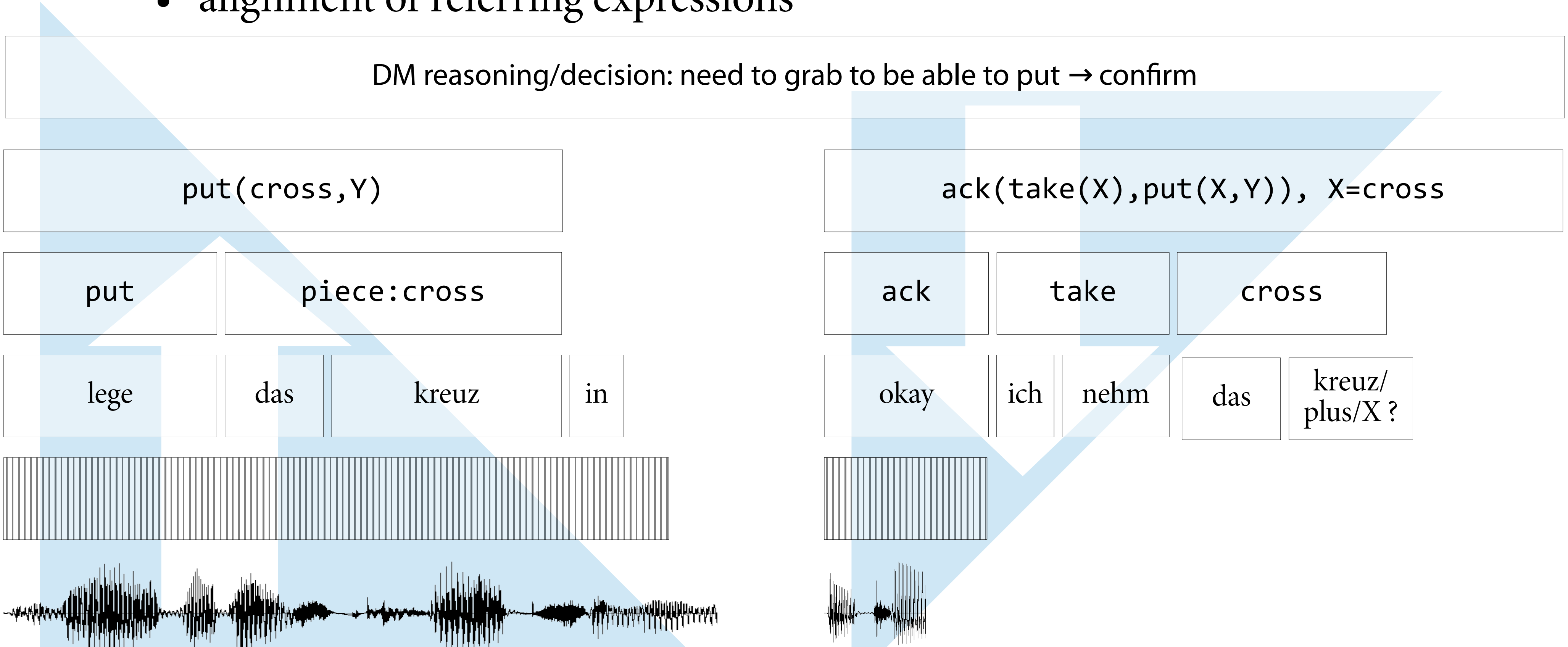
okay

ich

nehm

Incremental Units Model

- incrementality is mostly fun in end-to-end systems
  - modular systems in practice
  - many problems require grounded/non-modular input
    - aligning gestures with speech requires timed words (not just words)
    - alignment of referring expressions



DM reasoning/decision: need to grab to be able to put → confirm

`put(cross,Y)`

`put` `piece:cross`

lege das kreuz in

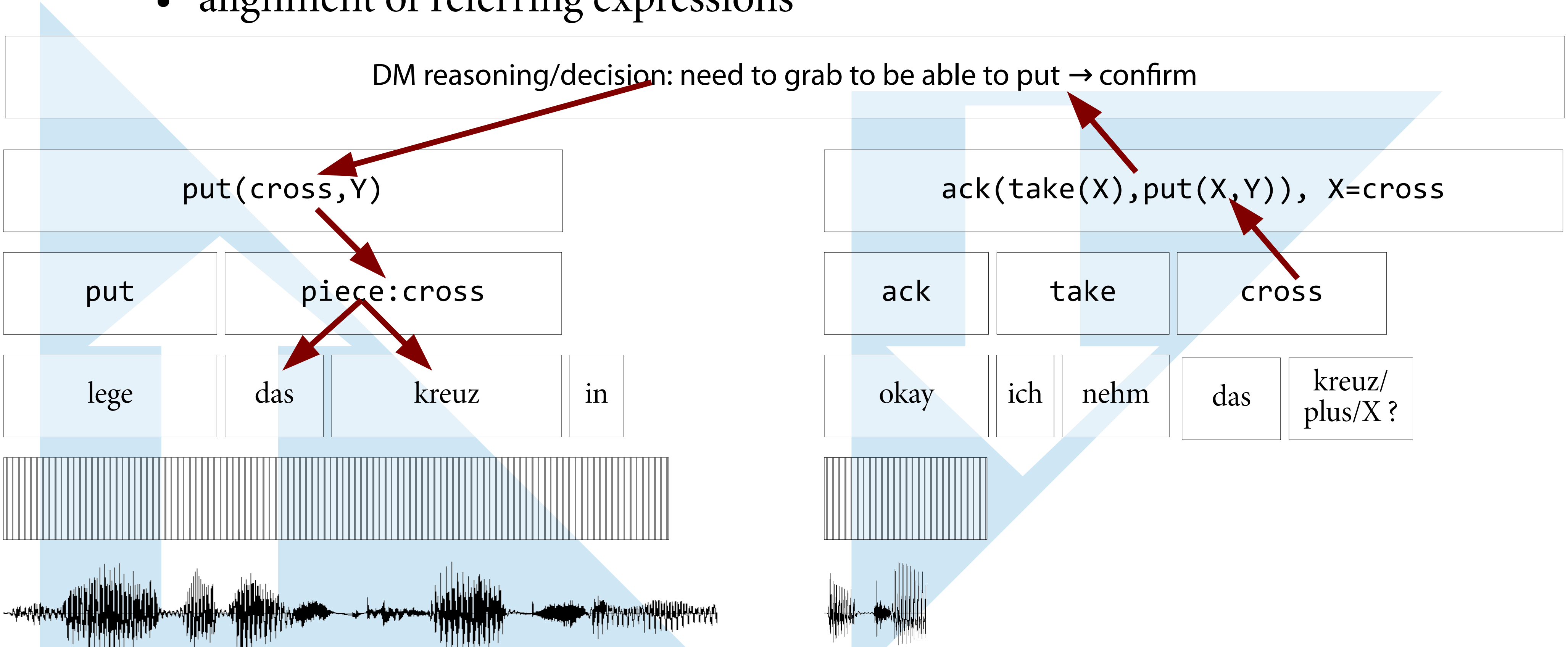
`ack(take(X),put(X,Y)), X=cross`

`ack` `take` `cross`

okay ich nehm das kreuz/
plus/X ?

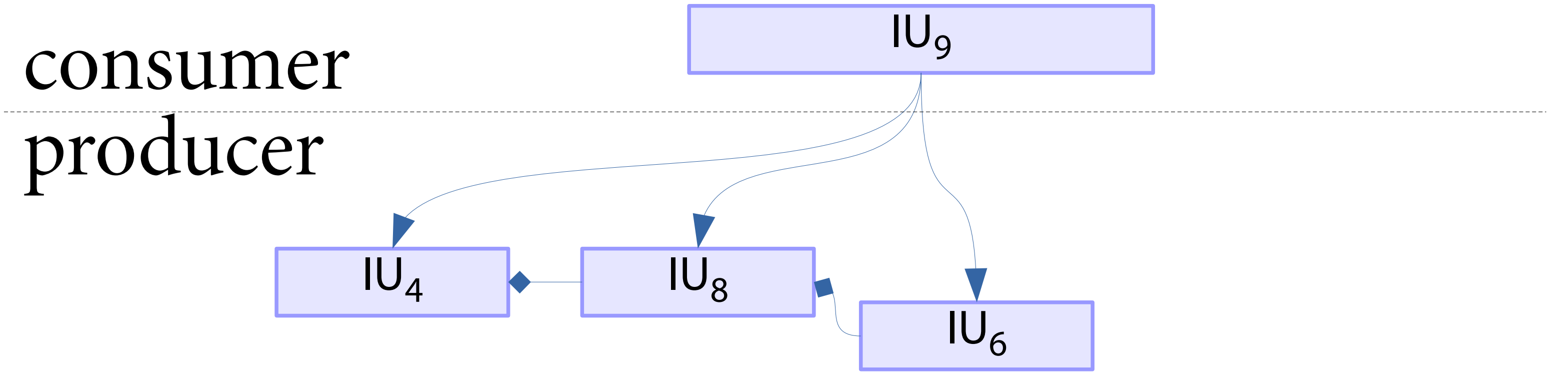
Incremental Units Model

- incrementality is mostly fun in end-to-end systems
  - modular systems in practice
  - many problems require grounded/non-modular input
    - aligning gestures with speech requires timed words (not just words)
    - alignment of referring expressions


DM reasoning/decision: need to grab to be able to put → confirm
put(cross,Y)
ack(take(X),put(X,Y)), X=cross
put
piece:cross
ack
take
cross
lege
das
kreuz
in
okay
ich
nehm
das
kreuz/
plus/X ?

Incremental Units Model

- also supports N-best hypotheses
  - to the point of forwarding the full beam



consumer

producer

IU_9

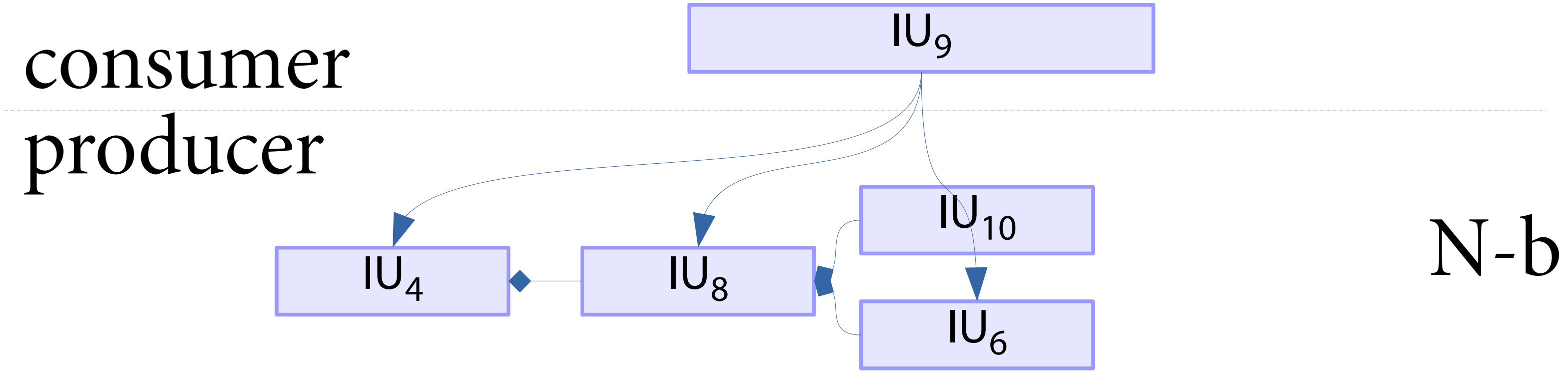
IU_4

IU_8

IU_6

Incremental Units Model

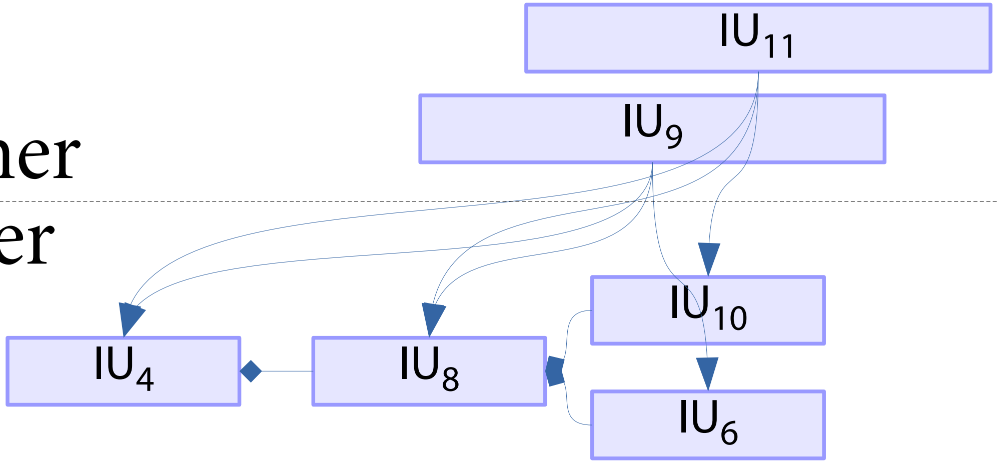
- also supports N-best hypotheses
  - to the point of forwarding the full beam



N-best input

Incremental Units Model

- also supports N-best hypotheses
  - to the point of forwarding the full beam



consumer

producer

potential for N-best output

N-best input

Current and Future Research Opportunities

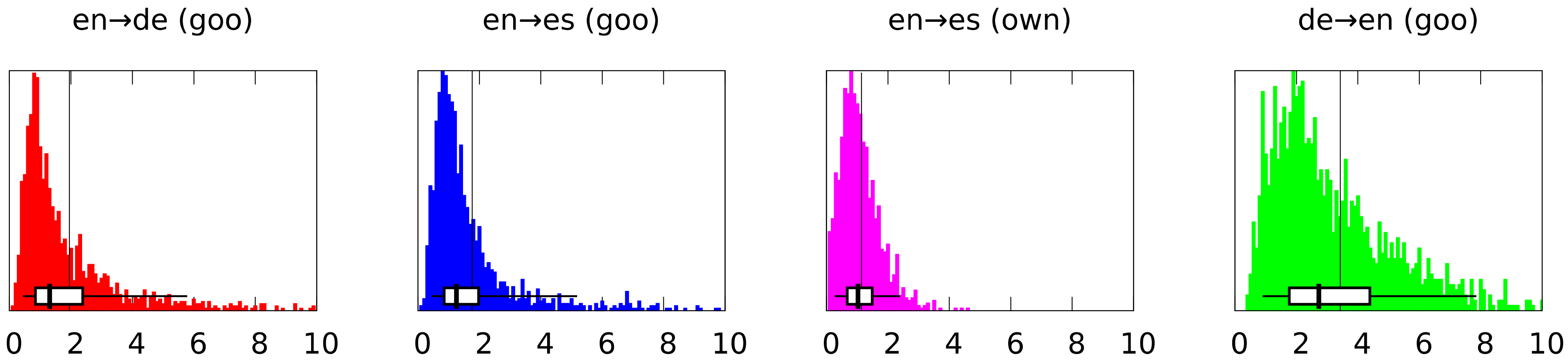
Speech-to-speech translation

- in its simplest form: ASR + translation + TTS
- incrementally: how much latency?
  - estimate effect of latency on accomodating all reordering

Baumann, Bangalore & Hirschberg (2014)

Speech-to-speech translation

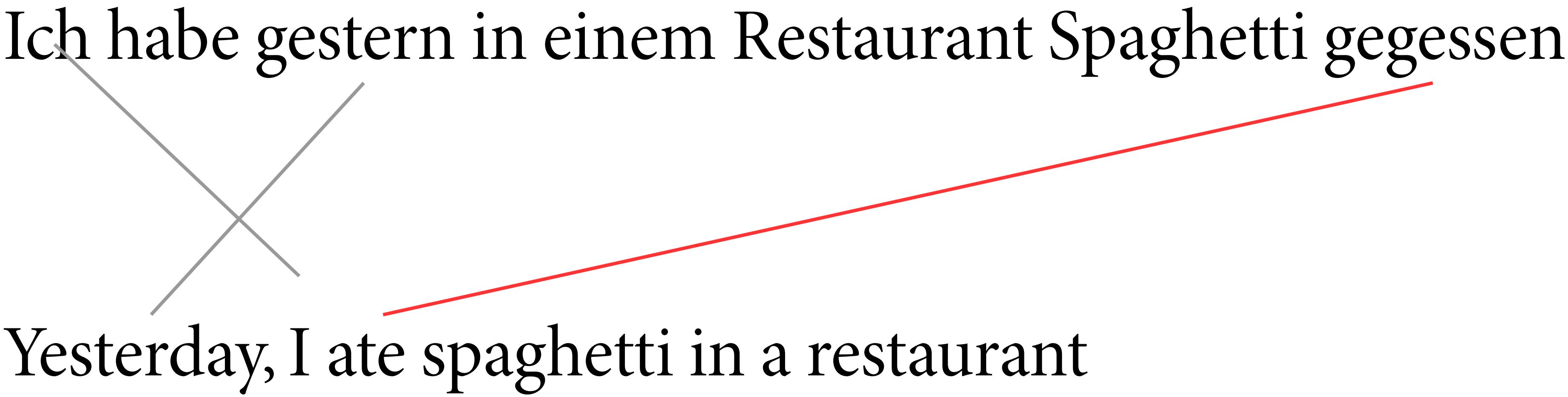
- in its simplest form: ASR + translation + TTS
- incrementally: how much latency?
  - estimate effect of latency on accomodating all reordering



Delay necessary to account for all re-orderings before speech can start.
German is worse on average, but all languages have a long tail.

Baumann, Bangalore & Hirschberg (2014)

Interactive Translation



Ich habe gestern in einem Restaurant Spaghetti gegessen

Yesterday, I ate spaghetti in a restaurant

- Predict final verb, correct if wrong (or keep suboptimal)

  e.g. (Grissom et al. 2014)

- Reorder target language

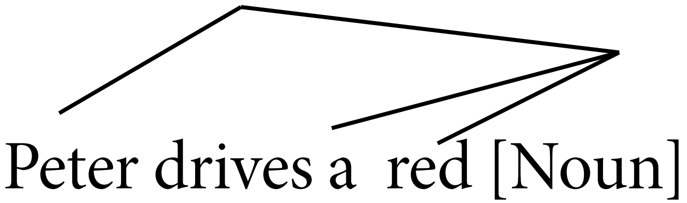
  e.g. (He et al. 2015)

Learning without Incremental Gold Standard

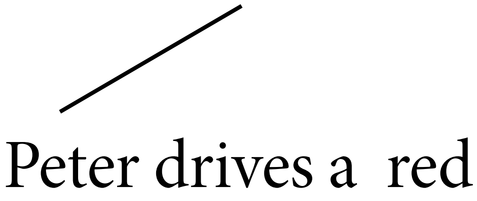
- Generated incremental gold standard unsatisfactory
  - Maybe more can be predicted
  - Predictions could be more fine-grained
- Predict word identities
  - “Invert” objective function to create predictions
  - Only possible if we still know the words

Structure to Structure Processing

- Not discussed this week
- Conceptually most difficult (? – not left-to-right)
- Example: Syntax → Semantics



Peter drives a red [Noun]

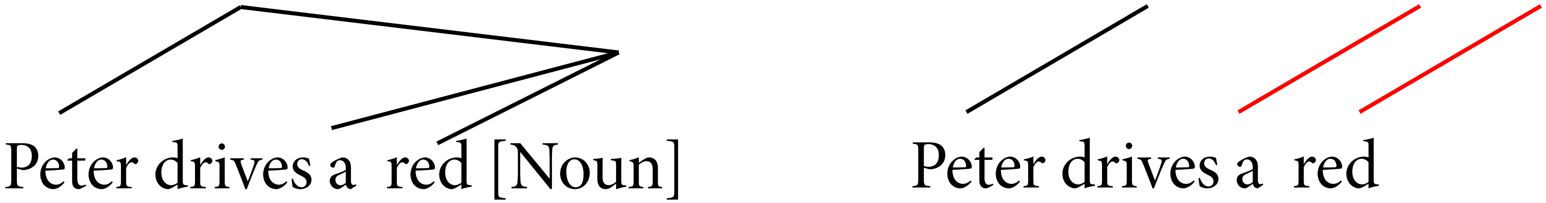


Peter drives a red

IMP_Q x8 (P) (Q)	IMP_Q x8 (P) (Q)
Peter[x8,]	Peter[x8,]
SUBJ[x9,x8,]	SUBJ[x9,x8,]
drive[x9,]	drive[x9,]
OBJA[x9,x10,]	
exists x10 (P) (Q)	exists x10 (P) (Q)
red[x10]	red[x11]

Structure to Structure Processing

- Not discussed this week
- Conceptually most difficult (? – not left-to-right)
- Example: Syntax → Semantics



Peter drives a red [Noun]

Peter drives a red

IMP_Q x8 (P) (Q)	IMP_Q x8 (P) (Q)
Peter[x8,]	Peter[x8,]
SUBJ[x9,x8,]	SUBJ[x9,x8,]
drive[x9,]	drive[x9,]
OBJA[x9,x10,]	
exists x10 (P) (Q)	exists x10 (P) (Q)
red[x10]	red[x11]

Speech and Gesture Recognition

- Input: Speech and Gestures (e.g. pointing)
- Integration at different levels possible
- Tight: One HMM trained with two (raw) inputs
  - Needs coupled training data
- Use candidate beams, find good matches
  - Can change both speech and gesture stream output
  - Variant: one-way integration with dominant channel
- Loose coupling: only create matches for streams

Further Speculation?

Thank you.

{baumann,koehn}@informatik.uni-hamburg.de
get the code at inprotk.sf.net.