ECS763U/P Natural Language Processing

Julian Hough

Week 11: Dialogue Models and Systems

With slides by Matthew Purver

CONTENTS

1) The challenge of dialogue
2) Dialogue act tagging
3) Dialogue System anatomy
   3.1) Focus: Automatic Speech Recognition
   3.2) Focus Information State Update (ISU) Dialogue Management
4) Training systems and evaluation

CONTENTS

1) The challenge of dialogue

Extreme Ellipsis: Dialogue

- *British National Corpus KSP 389-393:*

Christine	What have you been up to?
Steve	Nothing.
Michael	Eating.
Leslie	Any phone calls?
Steve	Nah.

- How could we summarise this dialogue?
  - *e.g. C asked what the others had been up to; S said he hadn't been doing anything, M said he'd been eating. L asked whether there had been any phone calls; S said there hadn't been any.*
- ("Summary" is longer than the dialogue …)

Dialogue

- This is what intelligent assistants need to do, e.g. for meeting summarisation:

  A: Well maybe by uh Tuesday you could
  B: Uh-huh
  A: revise the uh
  C: proposal
  B: Mmm Tuesday let's see
  A: and send it around
  B: OK sure sounds good

- How could we summarise this dialogue?
  - *e.g. A suggested (with C) that B could revise the proposal by Tuesday and send it around. <u>B agreed to do that</u>.*
  - *e.g. B agreed to revise the proposal by Tuesday.*

Extreme Ellipsis: Dialogue

- Can resolve ellipsis via “**Question Under Discussion (QUD)**”:
- *British National Corpus KSP 389-393:*

Christine	What have you been up to?	$ask(c, Q)$	$Q = \lambda\{a, x\}.up\_to(a, x)$
Steve	Nothing.	$answer(s, Q(s, n))$	**-->** $up\_to(s, n)$
Michael	Eating.	$answer(m, Q(m, e))$	**-->** $up\_to(m, e)$
Leslie	Any phone calls?	$ask(l, Q')$	$Q' = \lambda\{a\}.\exists x.call(x)$
Steve	Nah.	$answer(s, \neg Q'(s))$	**-->** $\neg \exists x.call(x)$

- But this is still an active research area …
  - (assigning QUD update & attachment structure is hard!)

Extreme Ellipsis: Dialogue

- Questions Under Discussion (QUD) can be embedded:

- *British National Corpus KSV 282-285:*

Richard	Oh you're disappointed now aren't you, that I, coming back, it's really upset you
Anon 3	That you're coming back?
Richard	Yeah
Anon 3	Yeah

- *British National Corpus KSP 28-32:*

Kevin	Do you er, have you got any whatsername there?
Barry	What?
Kevin	Brochures.
Peter	Brochures?
Unknown	No.
Unknown	No I haven't.

Practical Dialogue Processing

- Human-human dialogue – shallow processing:
    - Dialogue act tagging
    - Topic modelling (& segmentation)
    - Combine DA structures & topics for:
        - Summarisation
        - Decision / action-item detection (e.g. Tur et al, 2010)
        - Dialogue classification/prediction
            - Mental health diagnosis & prediction (e.g. Howes et al 2014)

- Human-computer dialogue – can be deeper:
    - More constrained domain & task
        - Higher accuracy, less variation in structure
    - Potential for interaction
        - Clarification, correction, direction & control of structure

CONTENTS

1) The challenge of dialogue
2) Dialogue act tagging
3) Dialogue System anatomy
   3.1) Focus: Automatic Speech Recognition
   3.2) Focus Information State Update (ISU) Dialogue Management
4) Training systems and evaluation

CONTENTS

2) Dialogue act tagging

Dialogue Acts

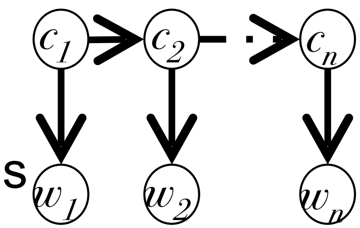
- Speech Acts / Dialogue Acts
  - “How to Do Things with Words” (Searle, 1952)
- Utterances in dialogue are **actions**
  - We ask questions … answer them ...
  - ... greet each other ...
  - ... make promises, threats ...
- And these actions have **effects**
  - introducing questions for discussion ...
  - ... resolving them ...
  - ... greeting, promising, ...
- We need to keep a **record** of actions & effects
  - we need them to give a meaningful summary
  - and can (must) use them to build meaning representations
    - (Ginzburg, 1994; 2012)

Dialogue Act Tagging

- Tag utterances with their action type (**dialogue act**):

Christine	What have you been up to?	ASK	WH-Q
Steve	Nothing.	ANSWER	NP-ANS
Michael	Eating.	ANSWER	NP-ANS
Leslie	Any phone calls?	ASK	YN-Q
Steve	Nah.	ANSWER	NEG-ANS

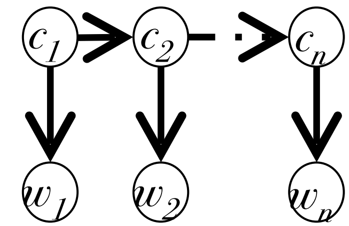
- Sequence modelling task
  - HMMs, CRFs, RNNs
  - Learn from labelled corpus e.g. Switchboard DA corpus



- Need a rich (often domain-dependent) tagset – e.g. Switchboard DA tags.

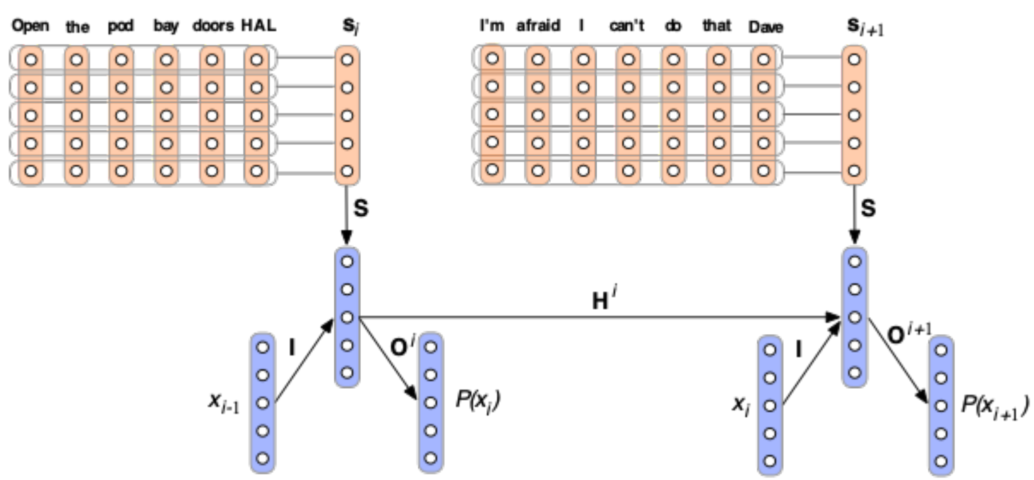
A:	So do you go to college right now?	YN-QUESTION
B:	Yeah	YES-ANSWER
A:	Are yo-	ABANDONED
B:	it's my last year	STATEMENT
A:	What did you say?	CLARIFY
B:	my last year	NP-ANSWER
A:	Oh good for you	APPRECIATION
B:	uh-huh	BACKCHANNEL

Dialogue Act Tagging



- Sequence modelling task
    - HMMs, CRFs as standard approaches
    - Features?
        - Words; syntax; semantics
        - Utterance length, POS patterns
        - Paralinguistic features e.g. intonation?
    - Transition probabilities?
- Needs training from relevant data
    - What corpus?
- Recurrent neural networks (next semester Deep Learning + NLP course!)
    - Kalchbrenner & Blunsom (2013)

`Who's that?`	`WH-Q`
`Jim`	`NP-ANS`
`Jim?`	`CLARIFY`
`Yes`	`POS-ANS`
`Is he OK?`	`YN-Q`
`No`	`NEG-ANS`



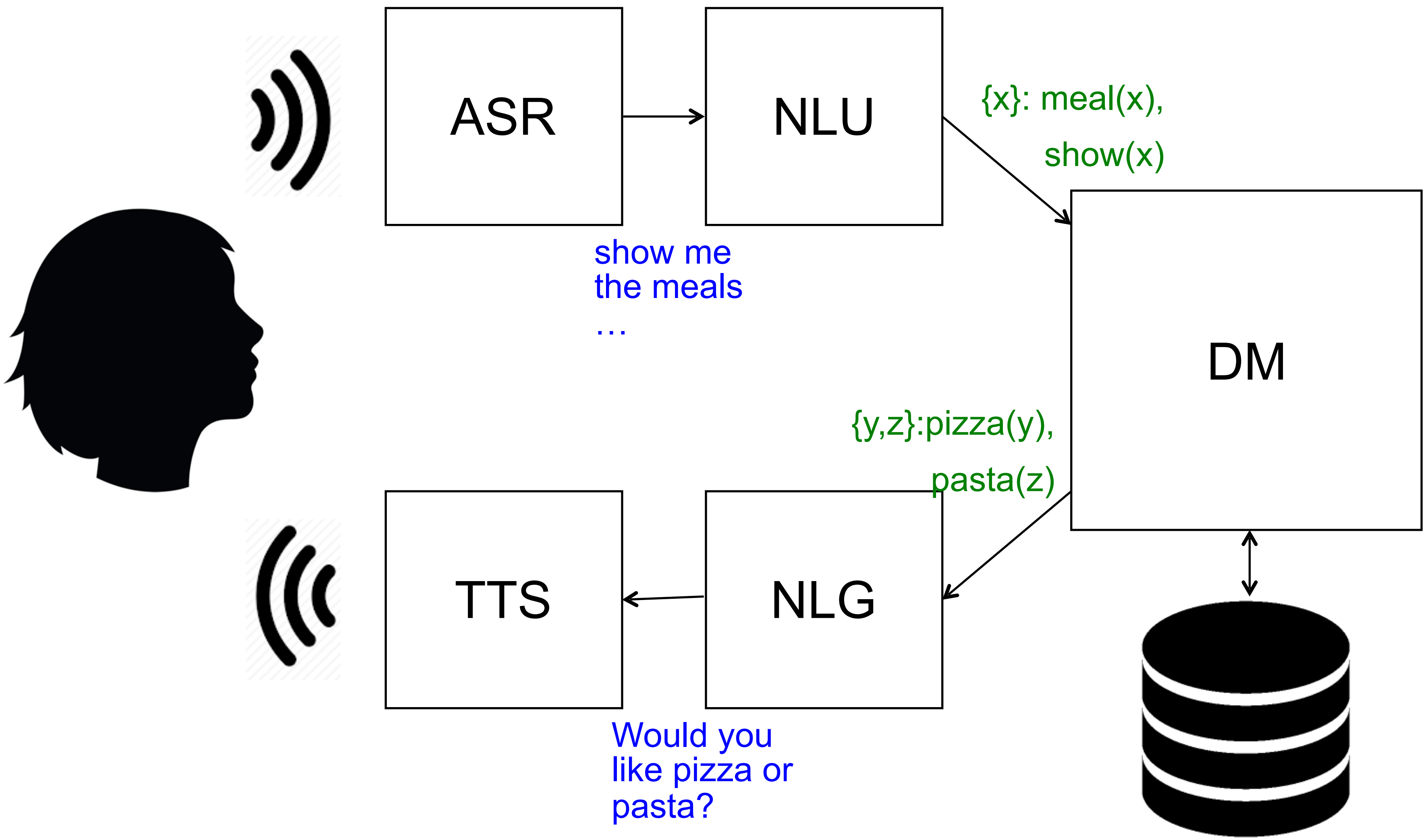
CONTENTS

1) The challenge of dialogue
2) Dialogue act tagging
3) Dialogue System anatomy
   3.1) Focus: Automatic Speech Recognition
   3.2) Focus Information State Update (ISU) Dialogue Management
4) Training systems and evaluation

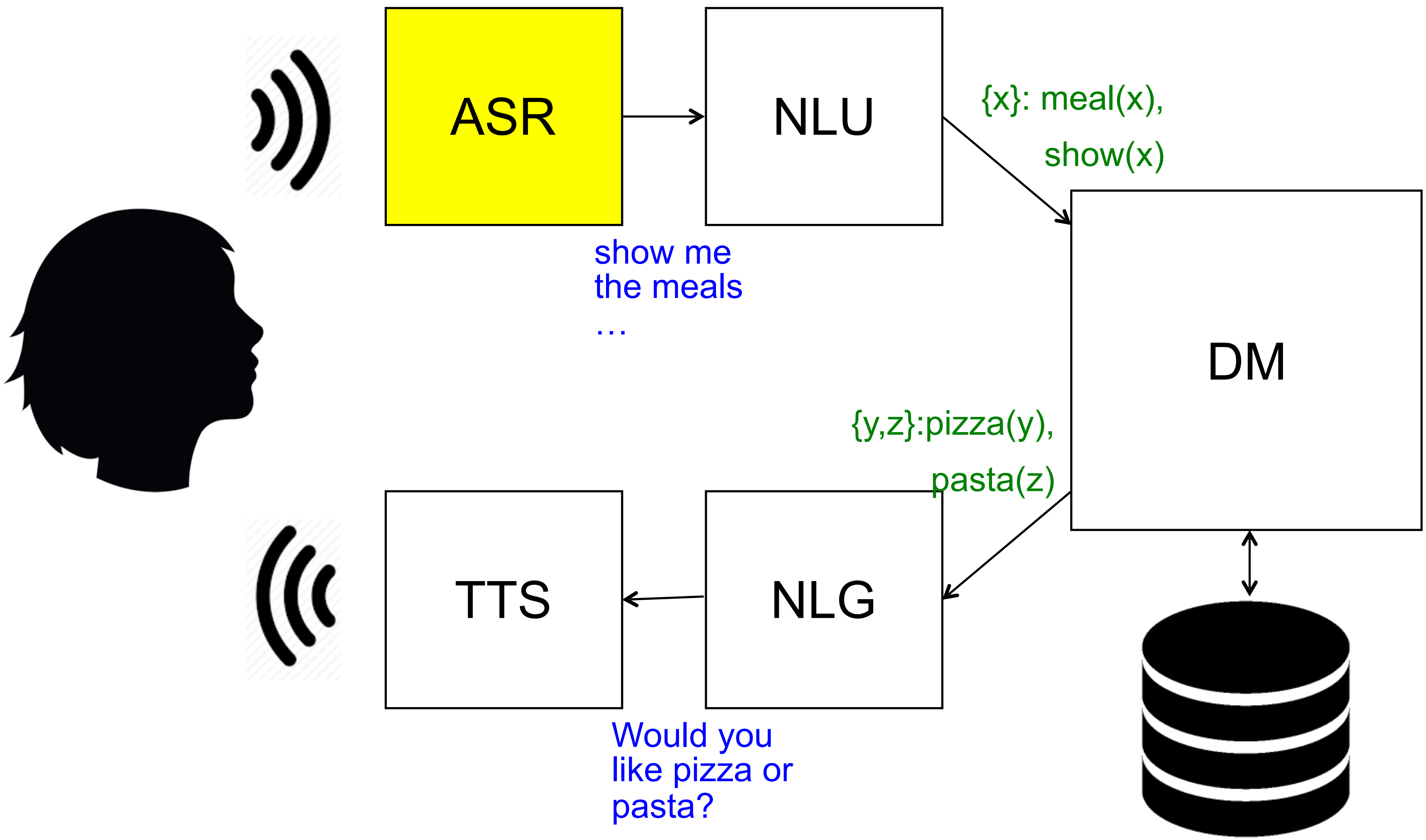
CONTENTS

3) Dialogue System anatomy
   3.1) Focus: Automatic Speech Recognition
   3.2) Focus Information State Update (ISU) Dialogue Management

Dialogue Systems



Dialogue Systems



What is ASR?

- **ASR ('automatic speech recognition**') is the machine transcription of words spoken by a human voice
- Specifically, we are concerned with **continuous speech recognition** (rather than recognition of single words)
- It has had a long history (since the 1950's)
- One of the big challenges in Artificial Intelligence more generally.
- Some claim it's a solved problem, others disagree!

How does ASR work?

- Generally comprises two components:
    - **Acoustic model**
    - **Language model**

- Integral to both is **decoding***:* the process/algorithm which transforms the signal from a human voice into the eventual **word hypotheses**

- Important that both models are statistically trained from data.

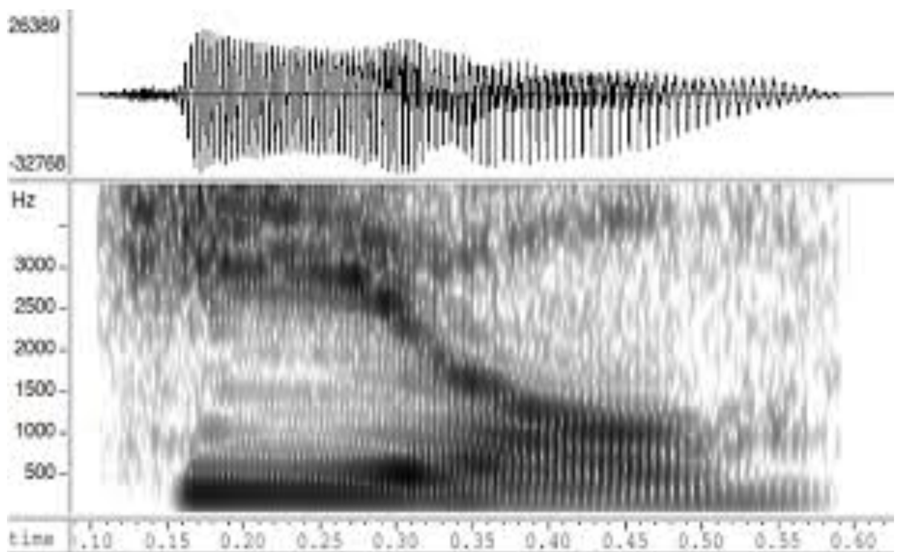
Acoustic Models

- Traditional job is to decode from:
- acoustic signal → **phonemes**
- For someone saying ‘John likes uh loves Mary’

ʤɒn laɪks ʌ lʌvz ˈmeəri

Acoustic Models

- Pre-processing: audio signal converted into small chunks known as frames (approximate duration of 10ms).



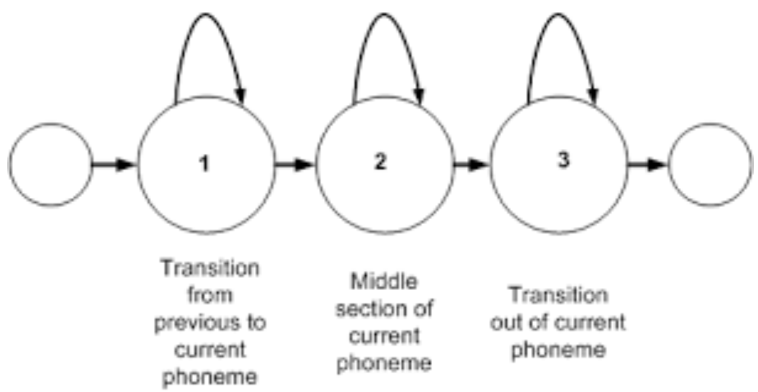
Acoustic Models

- Feature extraction: The raw audio signal from each frame can be transformed by applying the **mel-frequency cepstrum.** The coefficients from this transformation are commonly known as **mel frequency cepstral coefficients (MFCC)s.**

- MFCCs used as an input to individual **Gaussian** distributions for each phone along with other features.

Acoustic Models

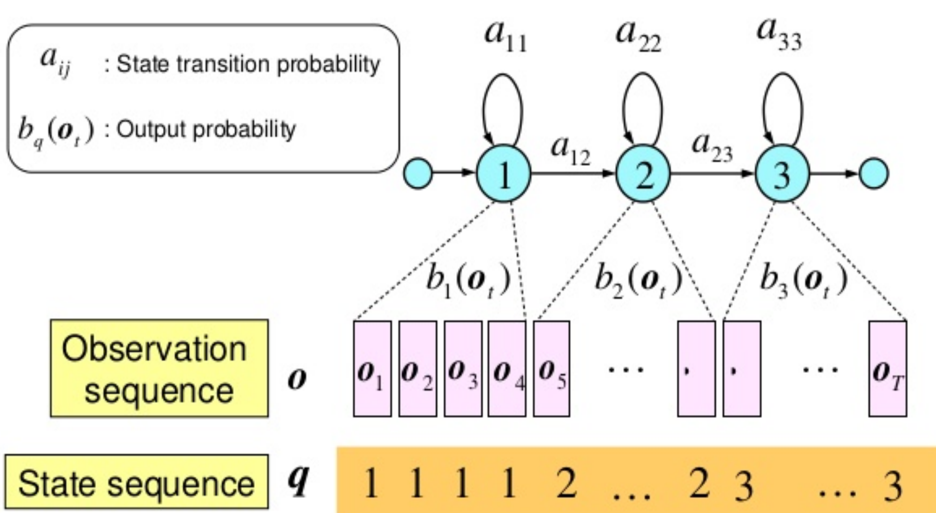
- Each **phone** is modelled as a **hidden markov model (HMM)** with three states:



- **Lexical/Pronunciation model:** Then, each word can also be modelled as a **word model** as an HMM (with states corresponding to phones)

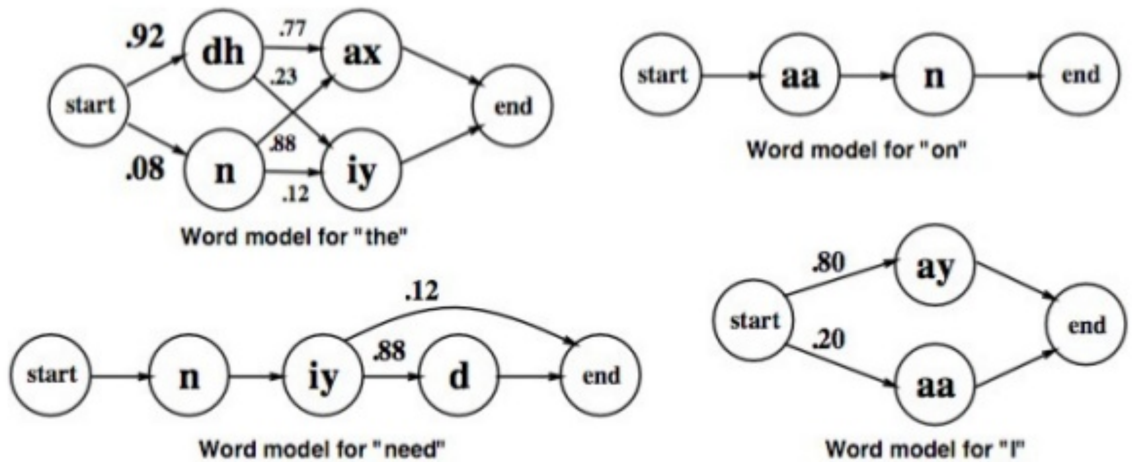
Acoustic Models

- **HMM** (brief overview, revision)
- A statistical (probabilistic) sequence model with **states** and **observations**
- Only observation sequence *o* available as input- the state sequence *q* is **hidden**



Acoustic Models

- Lexical/pronunciation modelling with HMM **word models**:



- Note these are hidden states $q_i \ldots q_n$, not directly observed. So it is **a noisy channel model.**

$$argmax_q p(q \mid o) = argmax_q p(o \mid q) p(q)$$

Language Models

- Traditional job of the ASR using the input from the acoustic model is to decode from:
- Phonemes → words

ʤɒn laɪks ʌ lʌvz ˈmeəri

John likes uh loves Mary

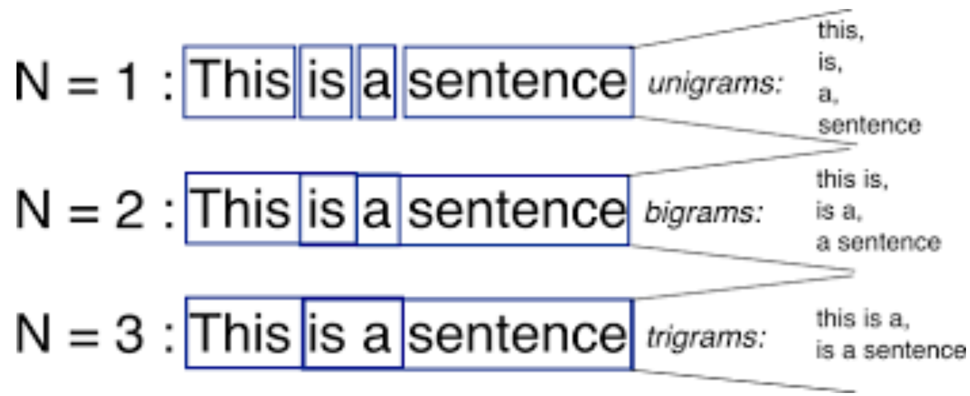
Language Models (revision)

- An LM answers the question: *what is the probability of this sequence of words being in a given language?*
- A conditional probability distribution over words, or sequences of words.
- The outcome conditioned on is the context of previous words in a sequence- i.e. given the previous sequence, what is the likelihood of the next word? e.g. bigram MLE:

$$p\left(w2 = likes \mid w1 = john\right) = \frac{\left|w1 = john \cap w2 = likes\right|}{\left|w1 = john\right|}$$

Language Models (revision)

- This can go up to any arbitrary length ('order'), e.g. 7-gram etc. In general **n-gram models** (Shannon ,1948).

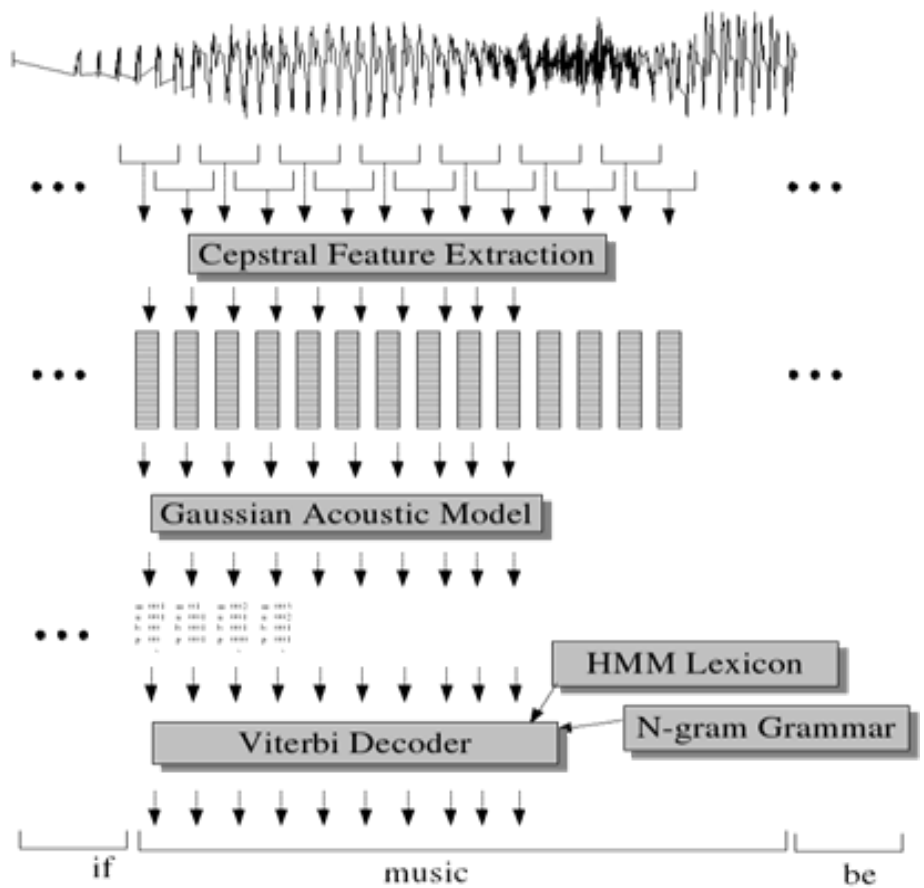


- General method is to extract the relevant n-grams (word sequences) according to the order *n*. In training this can be used to build the probability distributions.
- At decoding time, **smoothing** (Kneser-Ney etc.) on counts/raw probs invariably used to improve results.

ASR Decoding

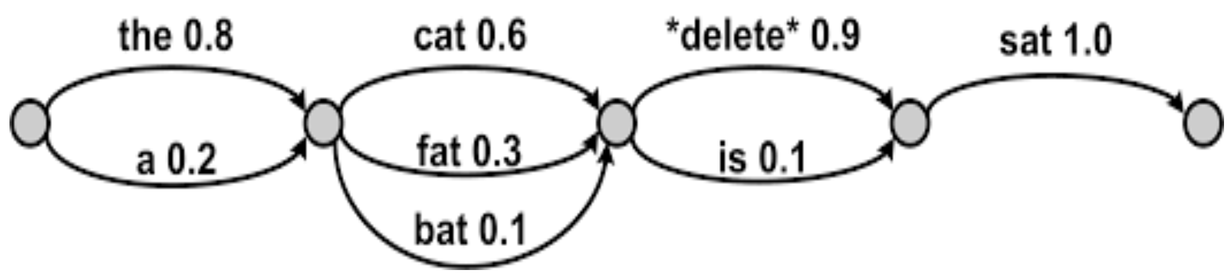
- In ASR the **language model scores combine with the acoustic model** scores to get the best possible overall score from the different hypotheses.
- Both sources of knowledge- acoustic and linguistic are used.
- Accents in vowel quality and noise may prove troublesome for the acoustic model.
- Unfamiliar domain may trip up the language model, even if acoustic model is near perfect.
- <u>**Look at automatic subtitles in movies and try to work out which errors might be due to acoustic modelling problems, and which might be due to language modelling problems?**</u>

ASR Decoding



ASR Decoding

- Using Viterbi decoding or other methods, multiple 'top' hypotheses can remain
- The possible outcomes can be stored in a **word-confusion network** (sausage) or a **lattice:**



ASR Training

- *Audio Data:* Audio can be encoded at different **sampling rates** (i.e. samples per second – the most common being: 8, 16, 32, 44.1, 48, and 96 kHz), and different bits per sample or **bit-rate** (bits per sample, the most common being: 8-bits, 16-bits, 24-bits or 32-bits).

- Speech recognition engines work best if the acoustic model they use was trained with speech audio which was recorded at the **same sampling rate/bit-rate** as the speech being recognized.

ASR Training

- **Reference transcriptions:** Must be high quality with consistent spelling.
- **G2P: Grapheme to phoneme** conversion can be used to get the phones for training the acoustic model.
- **Training sets**: The acoustic model and language model can be trained separately (and even on separate data). Size depends on the complexity of the domain. Smoothing for LM may be done separately looking at **perplexity testing.**
- **Heldout and Test sets:** To avoid over-fitting and a genuine test, some of the data must not be used in training and instead used only to test how well the ASR does.

ASR Training

- **Speaker-dependent:** train a model for one person's voice (and test it on that voice)
- **Speaker-independent:** train a model for all voices (and test it on new voices not in training)
- Typical training methods:
  - **Forward-Backward training** assigns a probability that each vector was emitted from each HMM state (fuzzy labeling)
  - **Viterbi training** just assigns a feature vector to a particular state (most likely state from the best path)

ASR Evaluation

- Standard evaluation metric for speech recognition systems is the **word error rate (WER).**
- WER is based on how much the word string returned by the recognizer (hypothesis) **differs** from a correct or reference transcription.
- Given such a correct transcription, WER is computed as the minimum number of word *substitutions*, word *insertions*, and word *deletions* necessary to map between the correct and hypothesized strings (perfect = 0%, can be great than 100%, if all words replaced):

$$WordErrorRate = 100 \times \frac{\#Insertions + \#Substitutions + \#Deletions}{\#TotalWords \in CorrectTranscript}$$

ASR Evaluation

- Current performance is reaching human parity for transcription (Xiong et al. (Microsoft), 2016). **5% WER on conversational telephone speech.**
- However this is only when lots of in-domain data is available. Also, it is only reaching human *transcription* ability, not human speech recognition.
- Why string closeness, shouldn't we be more concerned about **semantic error rate**?
  - Yes, but this is normally idiosyncratic to your application. Often called **concept error rate** in dialogue systems research.

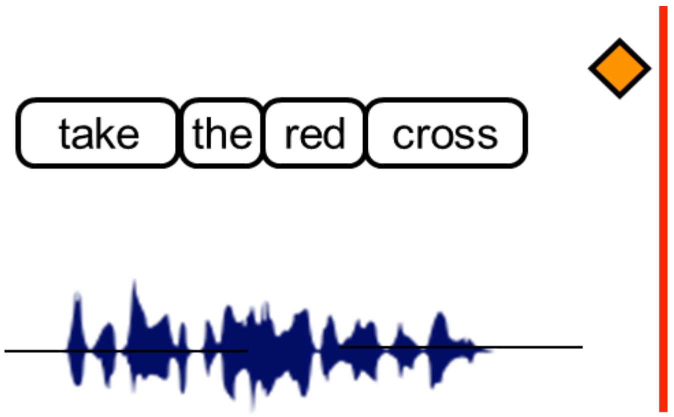
ASR Evaluation

- Performance differs with domain (and complexity) greatly:

TASK	**Vocabulary size**	**WER %**
Digits	11	0.5
Wall St. Journal Read Speech	5k	3.0
Wall St. Journal Read Speech	20k	3.0
Broadcast News	64k+	10.0
Conversation Telephone	64k+	20.0

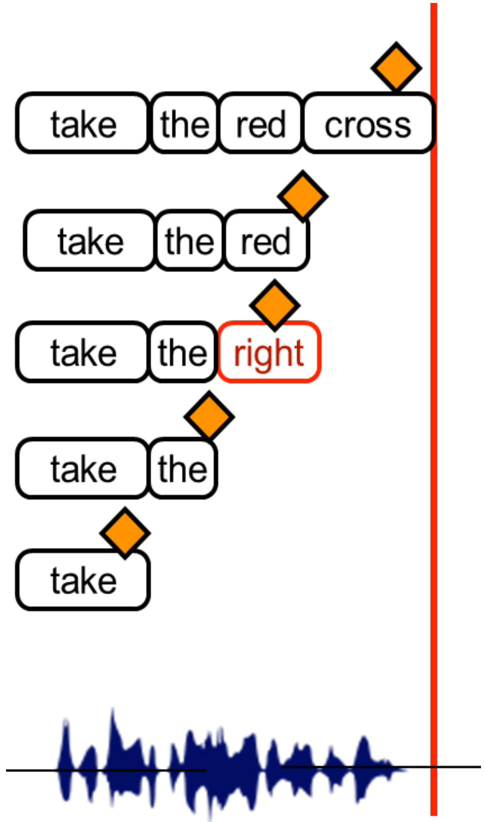
ASR Incremental Performance

- The need for speed! Not just accuracy, but how much delay in getting the hypotheses? (latency)



ASR Incremental Performance

- How does the evolution of the output happen over time? (stability)



ASR Tools

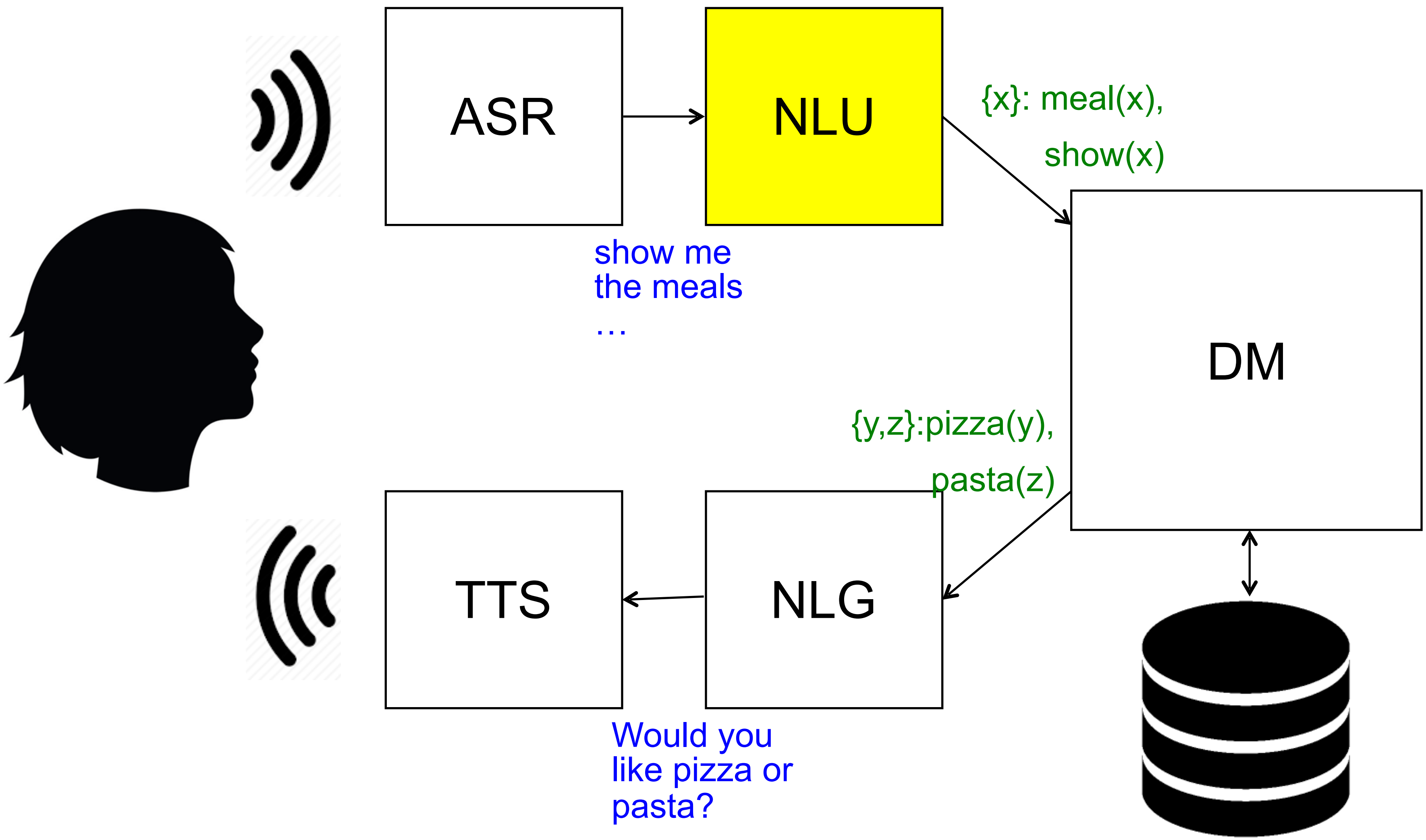
- Language modelling:
  - Off-the-shelf ASR (e.g. **Google Speech API**) is good now
  - But often need to train ASR for your language/domain to improve accuracy (**Sphinx**). Use the techniques you've used on this course!
- Grammar-based models
  - Much more limited, but you can write them without data
  - Sometimes we **want** a more limited model (constraints)
  - Java Speech Grammar Format (Java Speech API)
    - http://java.sun.com/products/java-media/speech/forDevelopers/JSGF/

```
public <basicCmd> = <startPolite> <command> <endPolite>;

<startPolite> = (please | kindly | could you) *;
<endPolite> = [ please | thanks | thank you ];

<command> = <action> <object>;
<action> = /10/ open |/2/ close |/1/ delete |/1/ move;
<object> = [the | a] (window | file | menu);
```

ASR Summary

- ASR is the machine transcription of words.
- WER can be used to measure its accuracy (closeness of hypothesis to reference).
- Acoustic Models: HMMs are a popular method for modelling sub-phone and phone sequences.
- Language models: N-gram models are used to estimate the likelihood of a sequence of words.
- In decoding, both acoustic and language models are used to get the optimal word sequence.
- ASR training can require lots of data, can be speaker-dependent or speaker-independent.

Dialogue Systems

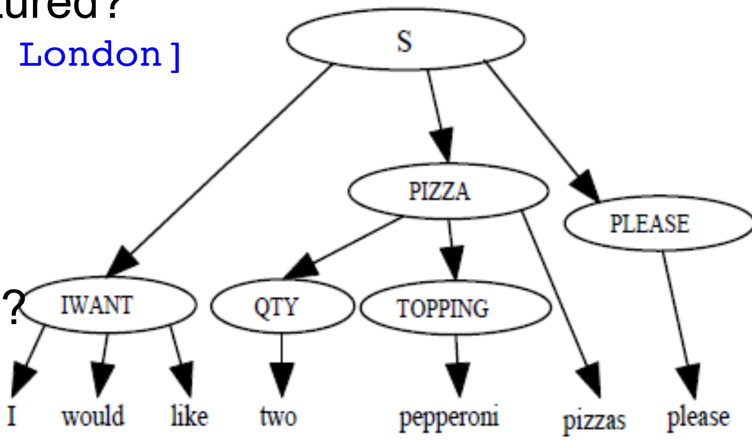


NLU: Natural Language Understanding

- This is the part you already know how to do (classification and parsing)
  - (and you know how ambiguous/errorful it can be …)
- But you will have many design choices:
  - Representation (LF) format – how deep/structured?

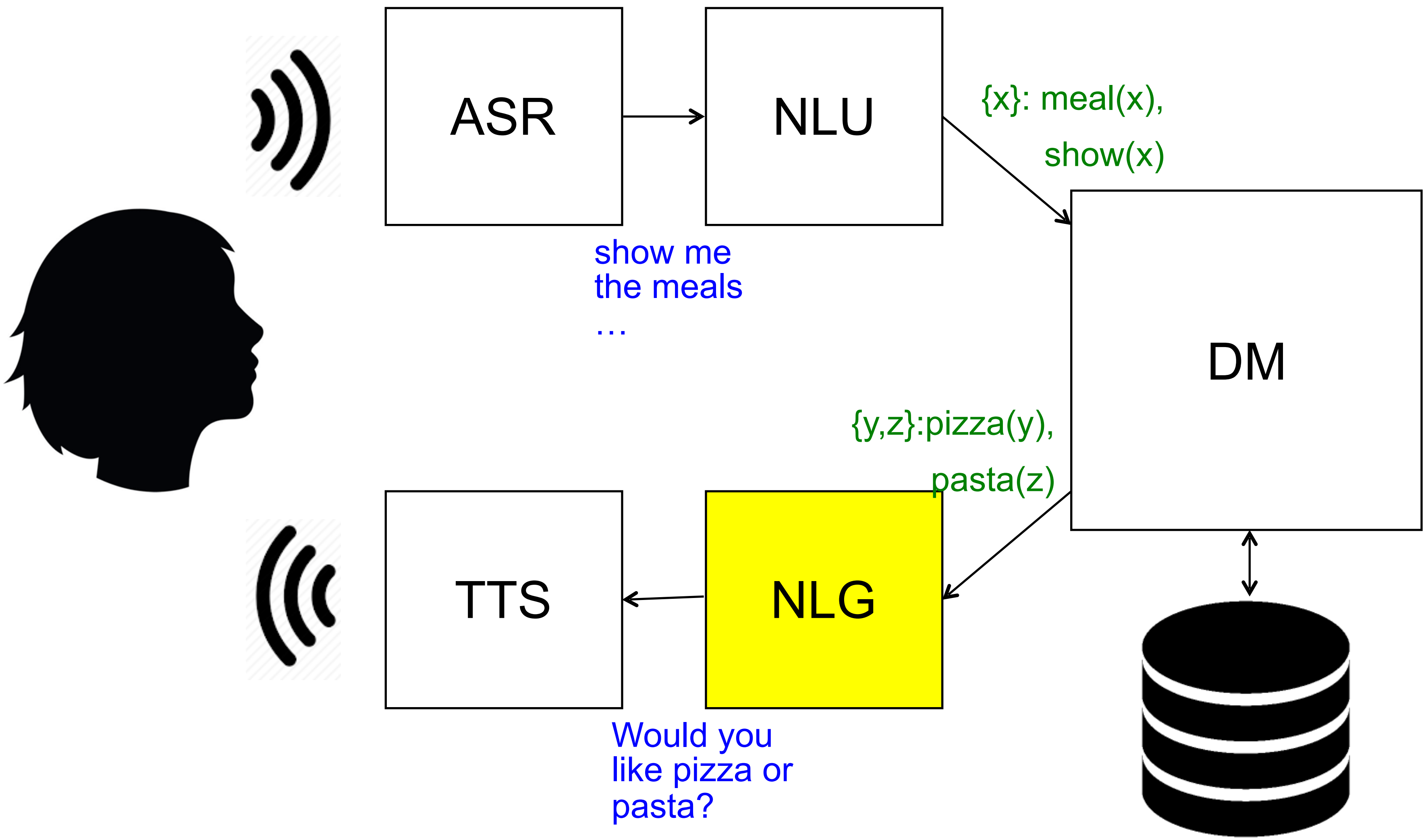
    `[action = go, start = Stockholm, end = London]`

    `[n=2, type=pepperoni]` vs.



  - Parsing method – grammar? HMM? RNN?
  - Single tag classification or sequence labelling?
  - Knowledge vs data?
- Java Speech API (JSGF) allows:
  - simple keyword-spotting
    - "... delete ..." → `[delete]`
  - pattern-matching/slot-filling
    - "I want to (go|fly|...) from {START} to {DEST} [on {DATE}]"
    - → `[start = START, dest = DEST, date = DATE]`

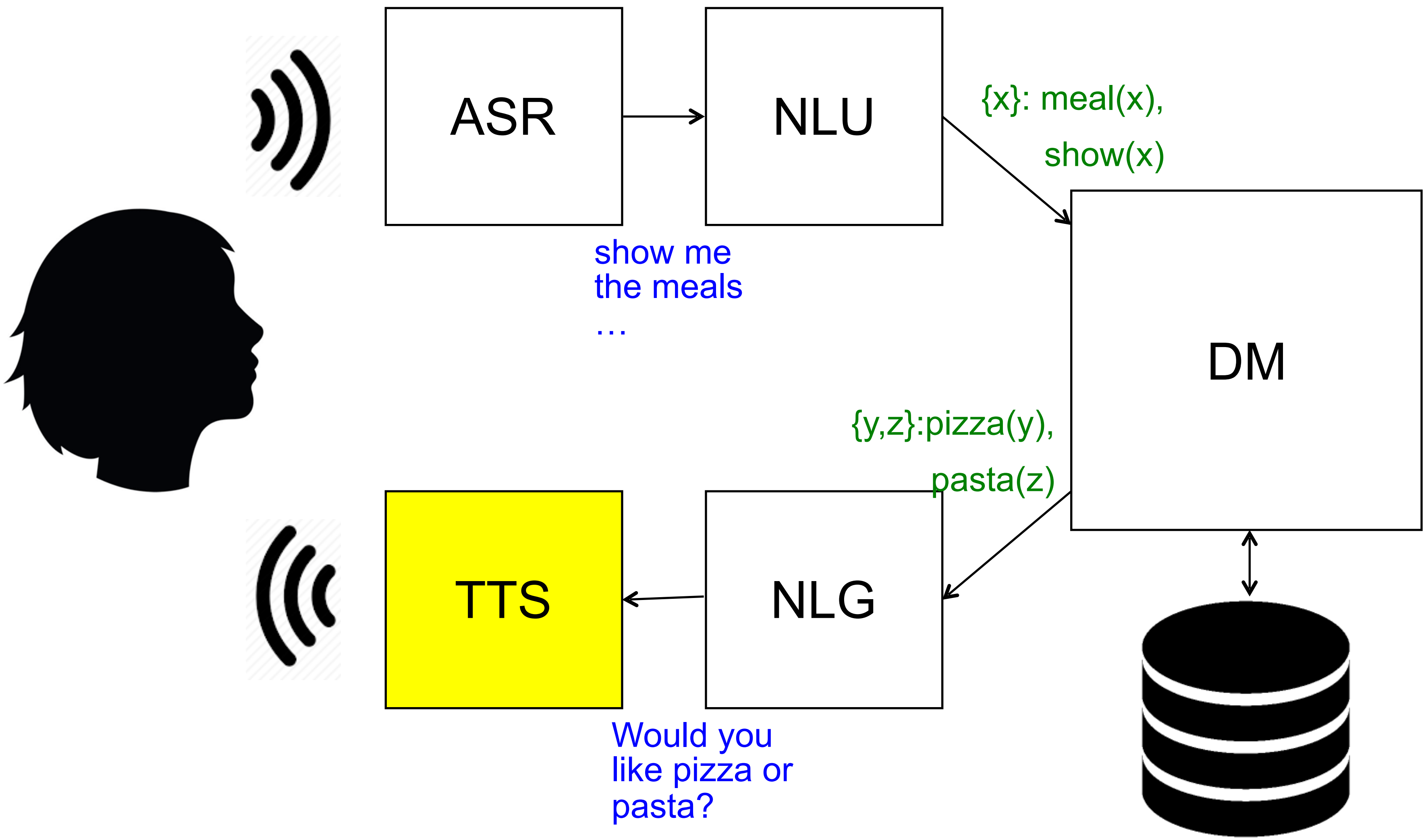
Dialogue Systems



NLG: Natural Language Generation

- The opposite of semantic parsing (NLU):
  - Input = semantic representations
  - Output = word sequences
- Often a **PLAN -> MICROPLAN -> REALIZE** pipeline (Dale and Reiter 2000).
- In limited domains, usually still template-based
  - “Getting flight details from {START} to {DEST} on {DATE}. One moment please.”
  - High-quality, simple
  - But time-consuming to engineer, can be monotonous
- Grammar-based:
  - Use NLU(-like) grammars, generation algorithm
  - More variation
  - Very time-consuming to engineer
- Statistical:
  - Learn e.g. sequence models, RNNs from annotated data
  - More chance of errorful output
  - Need a lot of data

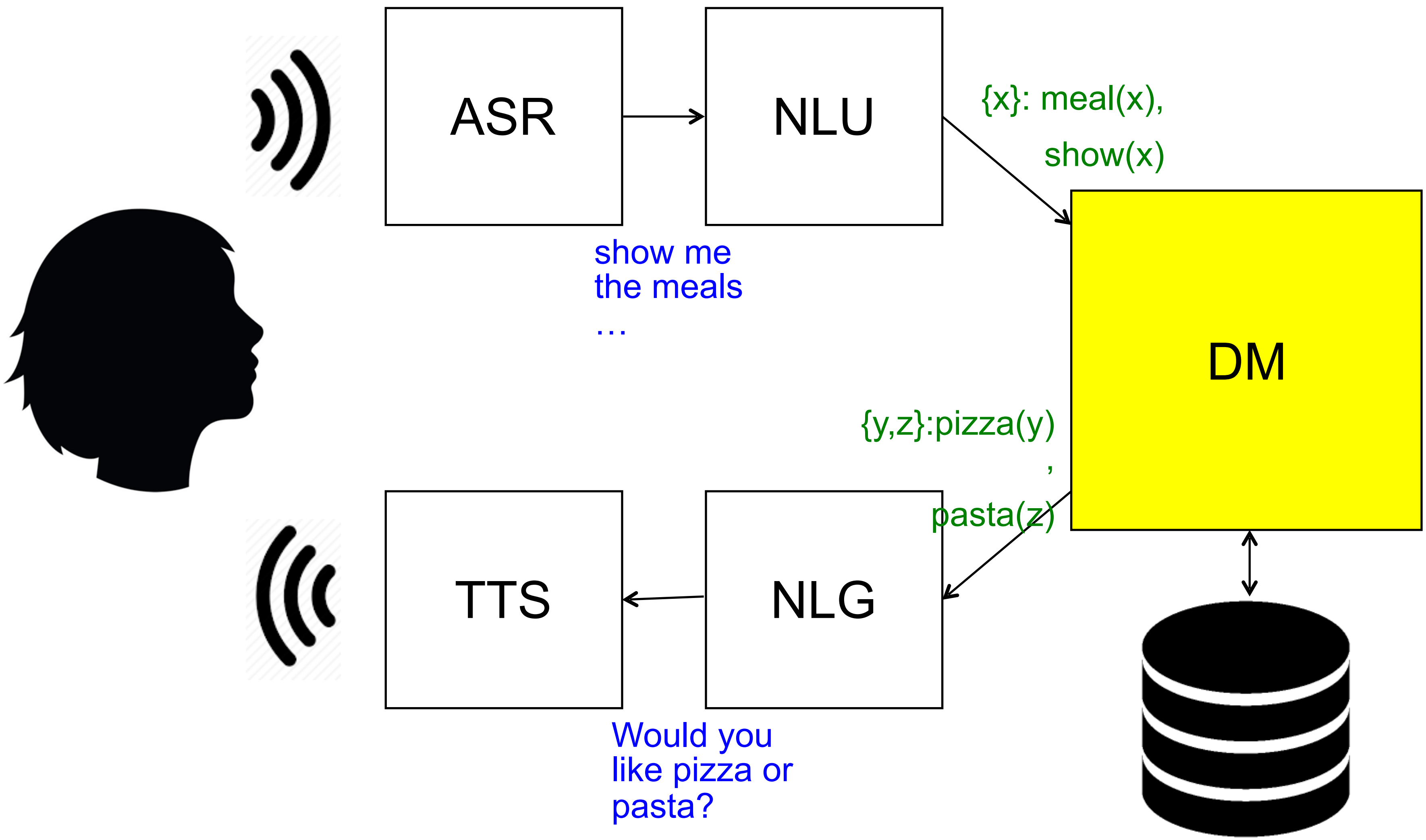
Dialogue Systems



TTS (Text-to-Speech)/ Speech Synthesis

- The opposite of ASR:
  - Input = word sequences (with markup?)
  - Output = speech signals
- In principle less difficult than ASR: no search problem
  - (we know what we're trying to say)
- Naturalness and intonation are difficult though
  - Rule-based approaches
    - Mathematical models generate each phoneme
    - e.g. DECTalk (Stephen Hawking)
  - Concatenative approaches
    - Record a sound for each phoneme (actually each **diphone**)
    - Play them back in sequence, with intonation e.g. FreeTTS
  - Fully recorded output
    - Simple, v high quality; but very expensive, inflexible
  - Best systems use a range of units & choose on-the-fly
    - Phones, diphones, words, … to whole sentences
    - e.g. Festival, Cereproc

Dialogue Systems



Dialogue Management

- A Dialogue System must decide what to say and how to say it:
  - NLG dealt with the 'how to say it' party
  - Dialogue Management (DM) deals with the '**what to say**' part (**content selection**).
- DM also has the role of maintaining the system's **state/context** as the interaction progresses- what does the semantics of an utterance from NLU do?
- DM (or sometimes called the '**action management**' part of it), manages the non-linguistic actions the system. It communicates with underlying application and triggers what it needs to do- e.g. database look-up, ordering train tickets, play music etc.

Dialogue Management

Traum and Larsson (2003) definition of DM. All parts of a DS which:

- Update the dialogue context on the basis of interpreted communication (both that produced by the system and by other communicating agents, be they human ‘user’ or other software agents)
- Provide **context-dependent expectations** for interpretation of observed signals as communicative behaviour
- **Interface** with task/domain processing (e.g. database planner, execution module, other back-end system), to coordinate dialogue and non-dialogue behaviour and reasoning
- **Decide what content to express next** and when to express it

Dialogue Management

- One of its key roles is to manage communication and **error**
- There will be a lot of error/ambiguity!
- DM lets the user know what the system can understand
  - Helpful prompts
- DM lets the user know what the system did understand
  - informative & timely responses “searching the flight database ...”
- DM allows the user to **correct errors**
  - Telling them when the system didn't understand

Dialogue Management

- "**Grounding**": management of coordination/uncertainty

- How do humans do this? **Backchannels**:
  - "uh-huh", "I see", "OK".
  - "Wow!", "really?", "no!"
  - "Eh?", "what do you mean?", "did you say 'pizza'?"
  - Head nods, eyebrow raising, gaze, gesture (→ screen?)

Backchannels & Clarification

- Show **positive/negative** understanding at critical points
  - After user input
    - ASR & NLU can have very high error rates
  - When there's other processing to do (avoid silence)
    - “OK. Searching the flight database …”
- **Explicitly** indicate problem & level of the problem:
    - “What did you say?”
    - “Did you say “Avatar”?”
    - “I think you said “Avatar”, is that correct?”
    - “Which John do you want?”
- **Implicitly** check information
    - “What time do you want to see “Avatar”?”
    - “I found no cinemas showing “Avatar” after 9pm”
    - “The next showing of “Avatar” is at 8pm”

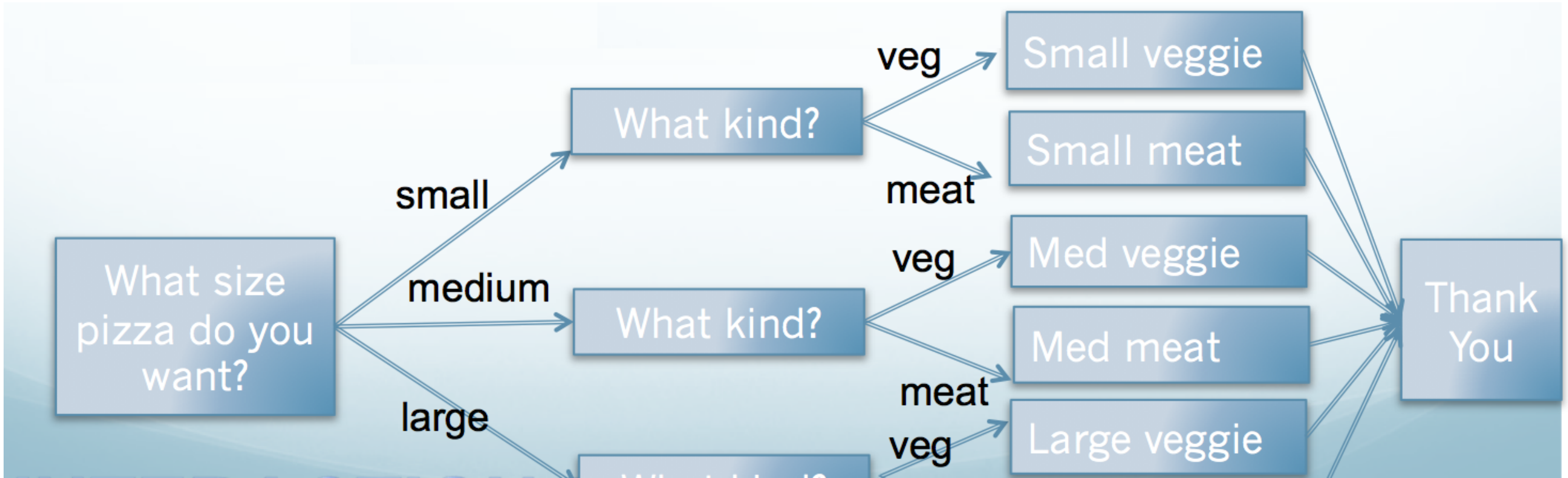
– Common strategy: drive from ASR model confidence
  - Confidence < threshold1: explicit rejections
  - Confidence < threshold2: explicit clarification
  - Otherwise: implicit confirmation in next action

DM Example

- DUDE system [1, 2]
  - Grounding via backchannels
  - Explicit vs implicit confirmation
  - Clarification
- SIRI with errors [3]

Rule-Based DM

- Dialogue as a graph (i.e. flowchart/script)
  - **Finite state machine**
  - Path per possible dialogue (including clarification etc)
  - Simple, controllable
  - Supported by standards
  - Only suitable for quite limited interactions
- Still the most common in commercial use
  - e.g. VoiceXML



Information-State-Based DM

- Moving beyond semi-scripted finite state approaches, Traum and Larsson (2003) and others proposed the idea of maintaining and updating an **information state** in dialogue.

- There was a need to try to develop a way of testing different **dialogue theories** in working systems.

- **TrindiKit** an IS-based toolkit was widely used in the 00's, and elements of its insight is still used.

Information-State-Based DM

- IS approaches assume a **BDI (Beliefs, Desires and Intentions)** model of agents, such that each agent has their private beliefs and agenda.
- It also contains what they believe to be **shared information/common ground** with their conversation participants.

Information-State-Based DM

Example of BDI:

Sys: ‘Where are you flying from?’

User: ‘from Paris to London’

User: ‘on Saturday’.

- The BDI inference helps resolve **non-sentential utterances** like ‘on Saturday’ and over-answering
  - ‘to London’ must be an answer to ‘where are you flying to?’ because getting an answer to that question is part of the system’s intentions, without the question being raised explicitly.

Information-State-Based DM

- The information state maintains what is stored by the agent at the present time (like a blackboard architecture) in terms of what is **private** to the system and what is **shared**, in a record data structure:

$$
\begin{bmatrix}
\text{PRIVATE} & : & \begin{bmatrix} \text{BEL} & : & \textsc{Set}(\textsc{Prop}) \\ \text{AGENDA} & : & \textsc{Stack}(\textsc{Action}) \end{bmatrix} \\
\text{SHARED} & : & \begin{bmatrix} \text{BEL} & : & \textsc{Set}(\textsc{Prop}) \\ \text{QUD} & : & \textsc{Stack}(\textsc{Question}) \\ \text{LM} & : & \textsc{Move} \end{bmatrix}
\end{bmatrix}
$$

(Traum and Larsson, 2003)

Information-State-Based DM

$$
\begin{bmatrix}
\text{PRIVATE} & : & \begin{bmatrix} \text{BEL} & : & \textsc{Set}(\textsc{Prop}) \\ \text{AGENDA} & : & \textsc{Stack}(\textsc{Action}) \end{bmatrix} \\
\text{SHARED} & : & \begin{bmatrix} \text{BEL} & : & \textsc{Set}(\textsc{Prop}) \\ \text{QUD} & : & \textsc{Stack}(\textsc{Question}) \\ \text{LM} & : & \textsc{Move} \end{bmatrix}
\end{bmatrix}
$$

- BEL: Beliefs (set of propositions)
- AGENDA: Intention of what info to get/actions to implement (stack)
- QUD: Questions Under Discussion/issues to be resolved (stack)
- LM: Latest Move (a dialogue move/act)

(Traum and Larsson, 2003)

Information-State-Based DM

- **Update rules** update the information state have preconditions and effects:

$$\text{U-RULE: } \mathbf{accommodateQuestion}(Q, A)$$

$$\text{PRE: } \begin{cases} \text{in}(\text{SHARED.LU, answer(usr,}A)), \\ \text{in}(\text{PRIVATE.PLAN, findout}(Q)) \\ \text{domain :: relevant}(\ A, Q\ ) \end{cases}$$

$$\text{EFF: } \begin{cases} \text{del}(\text{PRIVATE.PLAN, findout}(Q)) \\ \text{push}(\text{SHARED.QUD, } Q) \end{cases}$$

(Traum and Larsson, 2003)

Information-State-Based DM

- A typical interaction begins with the system intending to ask the user where they want to go.

$$\left[\begin{array}{lcl} \text{PRIVATE} & = & \left[\begin{array}{lcl} \text{BEL} & = & \{\} \\ \text{AGENDA} & = & \langle \text{raise}(?x.\text{dest-city}(x)), \text{raise}(\ldots), \ldots \rangle \end{array}\right] \\ \text{SHARED} & = & \left[\begin{array}{lcl} \text{BEL} & = & \{\} \\ \text{QUD} & = & \langle\rangle \\ \text{LM} & = & \ldots \end{array}\right] \end{array}\right]$$

(Traum and Larsson, 2003)

Information-State-Based DM

- A typical interaction (continued)

U-RULE: **integrateSysAsk**

$$\text{EFF: } \begin{cases} \text{push}(\text{SHARED.QUD}, ?x.\text{dest-city}(x)) \\ \text{pop}(\text{PRIVATE.AGENDA}) \end{cases}$$

(Traum and Larsson, 2003)

Information-State-Based DM

- A typical interaction (continued)

```
Usr:  Malvern
```

(Traum and Larsson, 2003)

Information-State-Based DM

```
Sys:  Welcome to the travel agency!
Usr:  flights to paris
Sys:  Okay, you want to know about price.  A flight.
To Paris.  Let's see.  What city do you want to go
from?
Usr:  london
Sys:  Okay, from London.  When do you want to travel?
Usr:  in april, as cheap as possible
Sys:  Okay, in April, economy class.  What day do you
want to travel?
Usr:  the fifth
Sys:  Okay.  The price is $300
```

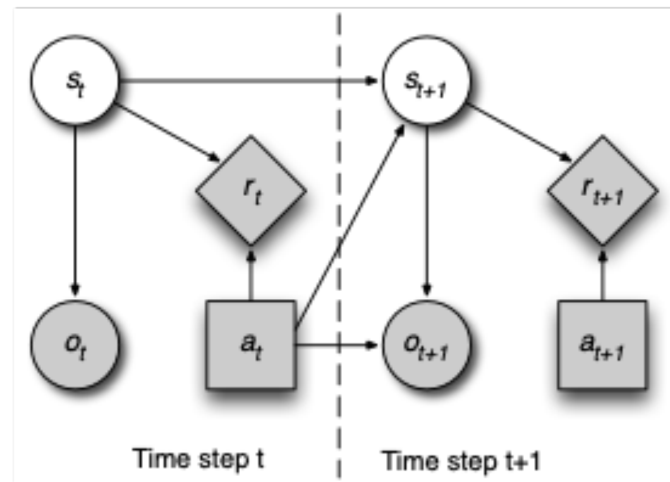
(Traum and Larsson, 2003)

Information-State-Based DM

- Information-state update:
  - Used in many research systems
  - Many advantages over pre-defined finite state systems.
  - Allows short answers, over-answering, several types of non-sentential utterances, and clarification interactions.
- Dialogue state is structured as sets of facts, questions, plans etc
  - Still rule-based, but more complex (deep) representations
  - Use of semantic LFs, ellipsis resolution, inference, planning
  - More flexible behaviour
  - BUT… ore complex to design & maintain
- Probabilistic versions (e.g. Lison, 2015)

Statistical DM

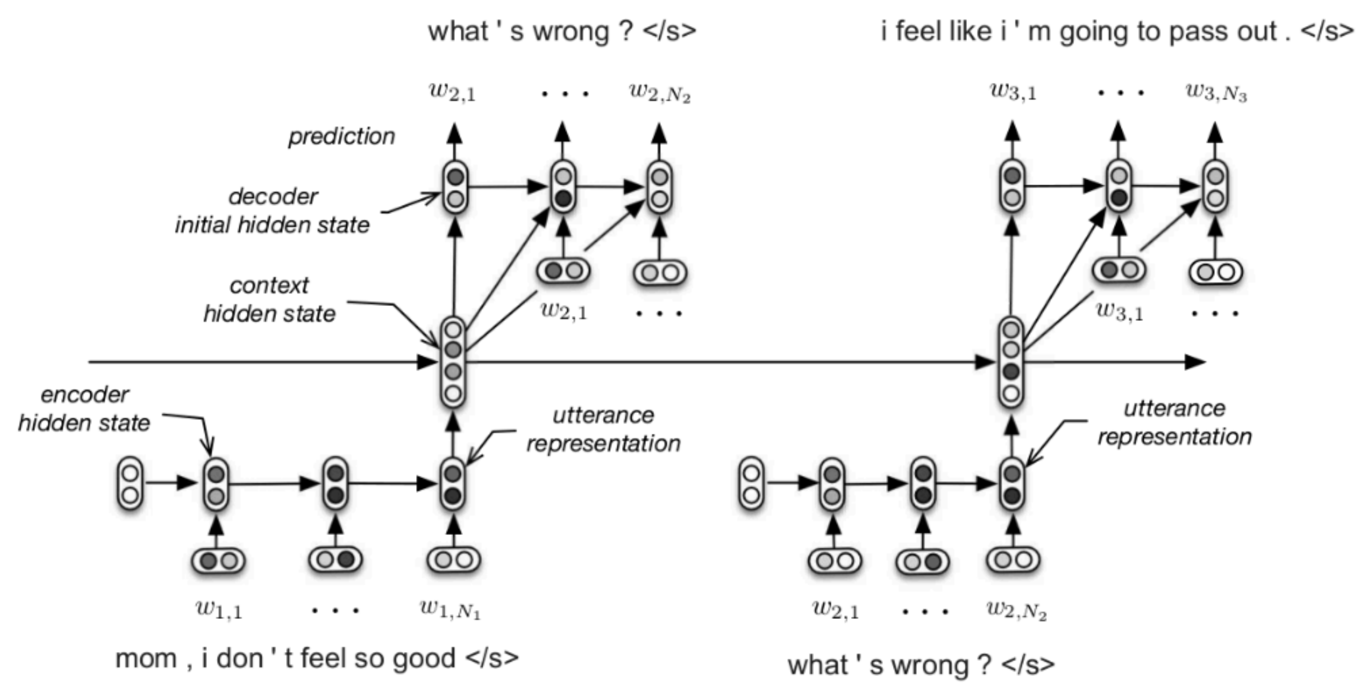
- Probabilistic models
  - Partially Observable Markov Decision Processes (POMDPs)
  - Used in many research systems, some commercial
  - e.g. VocalIQ (Apple)
- Sequence models (extension of HMMs)
  - Observed user moves *o*
  - State represents dialogue “belief” state *s*
    - e.g. destination = Paris, date = 2017-01-03
- Probabilistic decision process
  - Distribution over belief states
  - Emission probabilities $p(o|s)$
  - Transition probabilities $p(s_{t+1}|s_t)$
  - Take optimal system action *a* given expected reward *r*
- Trained from data **interactively**
  - Reinforcement learning
  - Explore possible system decision paths
  - Learn which led to good outcomes



Young et al (2013)

End-to-End NN Systems

- Increasing work in “end-to-end” (non-modular) systems
  - e.g. hierarchical recurrent NNs (Serban et al, 2015)
- Entirely data-driven
  - Robust; but data-hungry and non-modular
- More next semester!



CONTENTS

1) The challenge of dialogue
2) Dialogue act tagging
3) Dialogue System anatomy
    3.1) Focus: Automatic Speech Recognition
    3.2) Focus Information State Update (ISU) Dialogue Management
4) Training systems and evaluation

CONTENTS

4) Training systems and evaluation

Training

- Where do we get data from?
- Annotated existing dialogues
    - e.g. Switchboard corpus
    - Good for general dialogue act tagging
    - But limited:
        - We often need domain/system-specific data
        - No use for POMDP training
        - (Dialogues can go in many different directions)
- Wizard-of-Oz studies
    - Gather data using humans as simulated systems
    - Good for small datasets, and for system prototyping/evaluation
- Reinforcement learning needs thousands/millions of interactions
    - User simulations
    - Train simulated user (e.g. DA n-gram model)
    - Use in probabilistic training

Evaluation

- Task-level evaluation metrics
  - **Efficiency:** elapsed time, system turns, user turns
  - **Quality:** mean recognition/understanding scores, timeouts, rejections, helps, cancels etc.
  - **Task success:** database query completion rate etc.
- User satisfaction metrics
  - **Survey-based** e.g. 5-point Likert scale questionnaire
  - Harder to get, harder to pinpoint individual components
  - But this is what we really want to know …
- PARADISE method (Walker et al, 1998)
  - Measure:
    - (a) module/task-level metrics
    - (b) User surveys on same data
  - Train linear regression model to predict (b) from (a)

Reading

- Jurafsky and Martin (3rd Ed) Chapter 24. “Dialogue Systems and Chatbots”
- Traum, D. and Larsson, S. (2003). “The information state approach to dialogue management”