Putting things together

- Spoken dialogue systems
  - Types of SDS
  - Dialogue management
- Email query answering
  - Processing stages with sample email
  - Email response vs spoken dialogue
- Wrapping up

Spoken dialogue systems

1. Single initiative systems (also known as system initiative systems): system controls what happens when.

   System: Do you have your customer reference number? Please say yes or no.
   User: Yes

   Limited mixed-initiative:

   System: When do you want to leave?
   User: the twenty-third
   OR
   User: the morning of the twenty-third

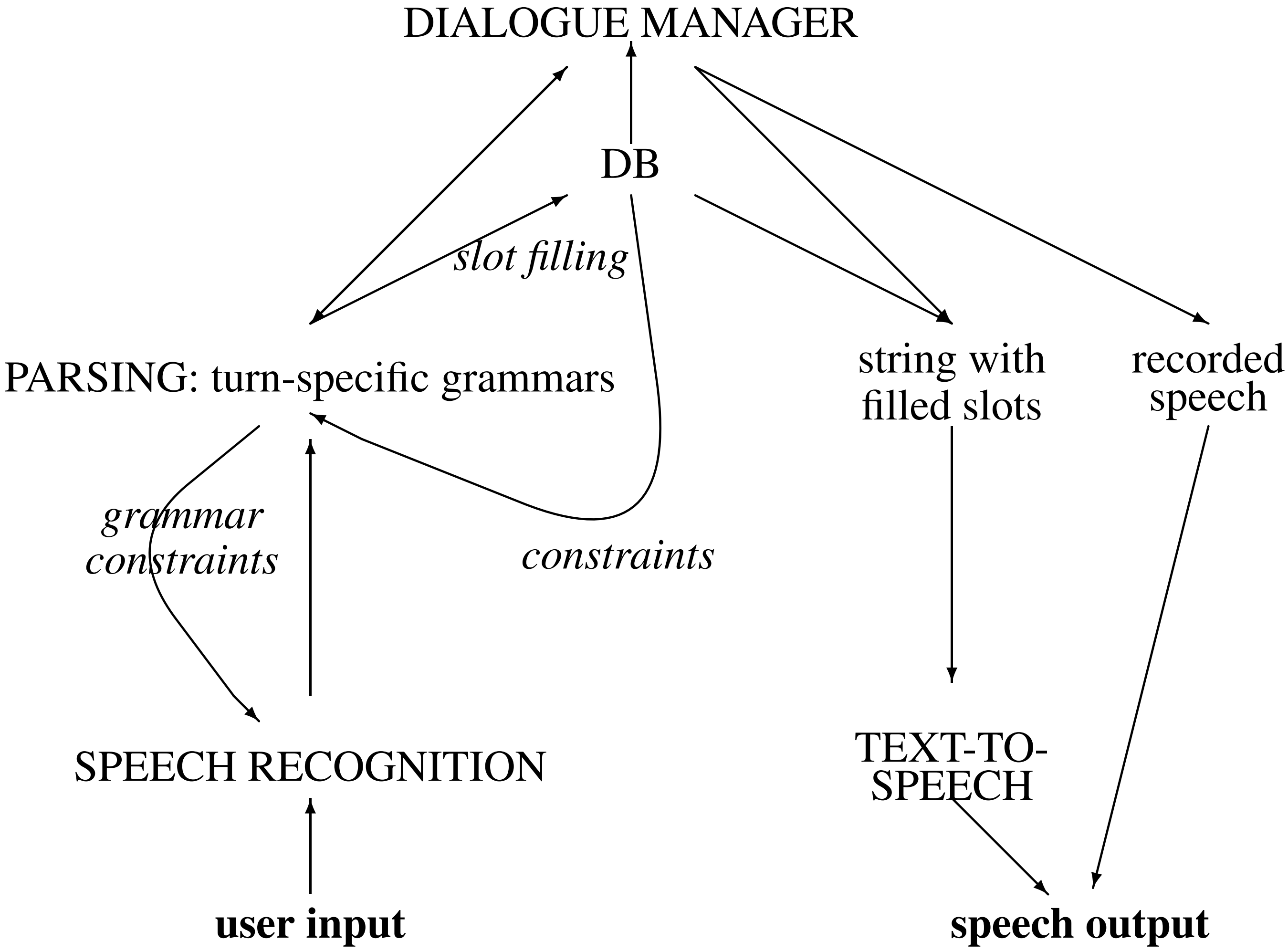
2. Mixed initiative dialogue. Both participants can control the dialogue to some extent.

   System: Which station do you want to leave from?
   User: I don't know, tell me which station I need for Cambridge.

Approaches to SDS

- Custom grammars: e.g. Nuance toolkit. FSAs or simple CFGs (compiled to FSAs) at each point in a dialogue controlled by an FSA. VoiceXML
- Statistical language modelling plus robust customised grammars or keyword spotting
- Statistical language modelling plus grammar induction
- Statistical language modelling plus general purpose grammar

Spoken dialogue system architecture



DIALOGUE MANAGER

DB

slot filling

PARSING: turn-specific grammars

string with filled slots

recorded speech

grammar constraints

constraints

SPEECH RECOGNITION

TEXT-TO-SPEECH

user input

speech output

Dialogue management in single-initiative SDS

- Finite-state dialogue manager
- Tightly controls the dialogue: prompts user for specific information
- Separate recognition grammar for every state
- DB may help specify the grammars: e.g.,
  1. prompt for post code
  2. get 100 items on n-best list from recogniser
  3. use first line of addresses from these to build a FS grammar
  4. prompt for 'first line' of address
  5. disambiguate post code
- Confirmation strategy is important

Email response

- more difficult to make limitations apparent to user in email response than in spoken dialogue systems
- no immediate feedback
- no speech recognition problem
- human fallback without delay or obvious transfer
- high precision needed

So:

- Broad coverage required
- Broad coverage possible, because no speech
- Deep grammars to get precision and meaning representation
- Relatively domain-independent grammars, otherwise too expensive

Ordering electronic goods

```
From amel  Mon Jan 27 22:37:34 PST 2003
From: amel@yy.com
Subject: OrderCancellation # 53000
MIME-Version: 1.0
Content-Type: text/plain; charset=iso-8859-1
Content-Transfer-Encoding: 8bit
Message-ID: <1043735854@foo.yy.com>

I talked with one of your CSR’s and she said
that she would cancel my order. I am writing
to check if it has in fact been cancelled.
```

0. Data collection.
1. Process email headers, tokenise email body. (i.e., separate words, detect non-words), run automatic spelling correction (maybe).
2. Morphology and lexical lookup
3. Parse with general typed feature structure grammar. Stochastic parse selection (trained on treebank). Semantics in TFS.
4. DB query extracted from semantics

Morphology and lexical lookup

```
I talk+PAST_VERB with one of your
CSR+SING_NOUN ’s and she say+PAST_VERB
that she would cancel+BSE_VERB my
order+SING_NOUN. I am writing+PRP_VERB
to check+BSE_VERB if it has
in fact been cancelled+PASSIVE.
```

spelling rule

```
prp_verb_infl_rule :=
%suffix (!t!v!c !t!v!c!cing) (* ing) (e ing)
                        (ee eeing) (ie ying)
lex_rule_infl_affixed &
[ ND-AFF +,
  SYNSEM.LOCAL prp_verb ].
```

irregular entry

```
said PAST_VERB say
```

multiword entry

```
in_fact_adv1 := adv_vp_aux_le &
 [ STEM < "in", "fact" >,
   SYNSEM [ LKEYS.KEYREL.PRED "_in+fact_a_rel",
            PHON.ONSET voc ] ].
```

Semantics

Your order was cancelled.

```
<mrs>
<var vid=’h1’/>
<ep><pred>prop_m_rel</pred><var vid=’h1’/>
<fvpair><rargname>MARG</rargname><var vid=’h3’/></fvpair>
<fvpair><rargname>PSV</rargname><var vid=’x4’/></fvpair>
<fvpair><rargname>TPC</rargname><var vid=’u5’/></fvpair></ep>
<ep><pred>def_explicit_q_rel</pred><var vid=’h6’/>
<fvpair><rargname>ARG0</rargname><var vid=’x4’/></fvpair>
<fvpair><rargname>RSTR</rargname><var vid=’h8’/></fvpair>
<fvpair><rargname>BODY</rargname><var vid=’h7’/></fvpair></ep>
<ep><pred>pro_poss_rel</pred><var vid=’h9’/>
<fvpair><rargname>ARG0</rargname><var vid=’i11’/></fvpair>
<fvpair><rargname>ARG1</rargname><var vid=’x10’/></fvpair>
<fvpair><rargname>ARG2</rargname><var vid=’x4’/></fvpair></ep>
<ep><pred>pronoun_q_rel</pred><var vid=’h12’/>
<fvpair><rargname>ARG0</rargname><var vid=’x10’/></fvpair>
<fvpair><rargname>RSTR</rargname><var vid=’h13’/></fvpair>
<fvpair><rargname>BODY</rargname><var vid=’h14’/></fvpair></ep>
<ep><pred>pron_rel</pred><var vid=’h15’/>
<fvpair><rargname>ARG0</rargname><var vid=’x10’/></fvpair></ep>
<ep><pred>_order_n_rel</pred><var vid=’h9’/>
<fvpair><rargname>ARG0</rargname><var vid=’x4’/></fvpair>
<fvpair><rargname>ARG1</rargname><var vid=’i16’/></fvpair></ep>
<ep><pred>_cancel_v_rel</pred><var vid=’h17’/>
<fvpair><rargname>ARG0</rargname><var vid=’e2’/></fvpair>
<fvpair><rargname>ARG1</rargname><var vid=’i18’/></fvpair>
<fvpair><rargname>ARG2</rargname><var vid=’x4’/></fvpair></ep>
<hcons hreln=’qeq’><hi><var vid=’h3’/></hi>
                   <lo><var vid=’h17’/></lo></hcons>
<hcons hreln=’qeq’><hi><var vid=’h8’/></hi>
                   <lo><var vid=’h9’/></lo></hcons>
<hcons hreln=’qeq’><hi><var vid=’h13’/></hi>
                   <lo><var vid=’h15’/></lo></hcons>
</mrs>
```

Generation

Generating from this semantics gives:

```
your order was cancelled.
your order was canceled.
```

- Bidirectional grammar, but want to recognise multiple dialects and generate consistently in an appropriate one
- Full generation not actually used in e-commerce application
- Needs further work on speed, selection of realisation (i.e., the generated string) and implementation in a runtime system

Porting to new domains

- new lexical entries
- new uses, senses
- adjusting preferences
- connecting semantic representation to underlying KB/DB

Conclusion

- different processing modules
- different applications blend modules differently
- many different styles of algorithm:
  1. FSAa and FSTs
  2. Markov models and HMMs
  3. CFG (and probabilistic CFGs)
  4. constraint-based frameworks
  5. inheritance hierarchies (WordNet), decision trees (WSD)
  6. mixing hard and soft constraints (Lappin and Leass)
- evaluation

CSTIT MPhil course

http://www.cl.cam.ac.uk/Teaching/CSTIT/

Natural Language and Information Processing (NLIP) group in the Lab and Speech, Vision and Robotics (SVR) in Engineering.

Taught component (Term 1 and 2):

- Speech Processing (1 and 2)
- Language Processing (1 and 2)
- Computing and the Web
- Internet Applications
- Dialogue Systems
- Speech or Language Reading Club
- Speech and Language practicals

Project:

- 12-week research project (May – end July)

Application deadline: March 31st