11-751 Speech Recognition and Understanding

Spoken Language Understanding

Marsal Gavaldà
marsal@cs.cmu.edu

November 9, 1998

Outline

- Motivation
- The JANUS speech-to-speech translation system
- The ATIS task

Purpose of SLU

- Understand user (i.e. perform intended action), e.g. MT
- Help recognition by constraining possible word sequences, i.e. NL as knowledge source
  - phonetics, language model; **syntax**, **semantics**, **discourse**; general world knowledge
  - loose vs tight coupling of speech recognizer (SR) and NL module

Integration of knowledge sources (i)

Loose vs tight coupling of SR and NL components (cf. syntax-semantics dilemma in parsing)

- **Serial paradigm**:
  SR → N-best → NL
  - independent modules (easy to modify), fast, robust
  - SR may prune away true best, removal of prosody
- **Integrated architecture**
  - more complete knowledge brought to bear from the start
  - computationally expensive; word-incremental parsing

Integration of knowledge sources (ii)

- **Blackboard architecture**: multiple independent processes (e.g. acoustic, lexical, syntactical) post and retrieve intermediate results from common workspace
  - benefits of modularity without ordering constraints
  - complexity of control module
- Reference
  - Ward, K. and Novick, D. "On the need for a theory of integration of knowledge sources for spoken language understanding," *Proceedings of the AAAI-94 workshop on the integration of natural language and speech*

Carnegie Mellon

Characteristics of spoken language

Spontaneously spoken language is highly disfluent

- false starts, e.g. truncated words and phrases
- repetitions
- ill-formedness
- filled pauses, e.g. *uhm*
- unfilled pauses (i.e. silence)
- variable speaking rate
- noises

Carnegie Mellon

Noises

(From the VERBMOBIL / C-STAR conventions)

- **human noises**

  /ah/, /aw/, /cg/, /ch/, /eh/, /er/, /glottal/, /h#/, /ha/ /hm/, /huh/, /lg/, /begin_lg/, /end_lg/, /ls/, /mm/, /nn/, /noise/, /oh/, /oo/, /sneeze/, /sniff/, /throat_clear/, /uh/, /um/, /whistle/

- **non-human noises**

  #bang#, #beep#, #click#, #electronic_hum#, #headset#, #key_click#, #mechanical#, #microphone#, #mouse_click#, #noise#, #paper_rustle#, #phone_ring#, #scratching#, #slam#, #squeak#, #tap#

Carnegie Mellon

Requirements of SLU systems

And still, SLU systems supposed to

- accept input from casual users
- run in real-time
- perform intended task (i.e. understand user)

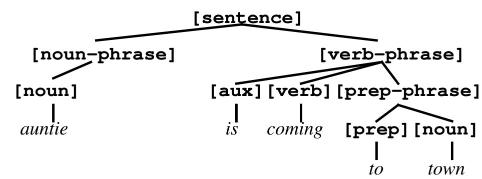
Carnegie Mellon

Analysis of spontaneous speech

- focus on information extraction (rather than detailed linguistic analysis)
  - E.g. JANUS speech-to-speech translation: focus on *transferring meaning* (elicit intended response)
- semantic grammars
- flat parsing
- hidden understanding
- robustness (e.g. skipping)

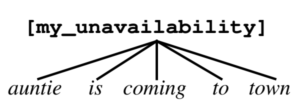
Syntactic grammars

- nonterminal symbols are grammatical categories
- domain-independent
- semantics obtained through non-trivial transformation of parse tree
- e.g. *Auntie is coming to town*



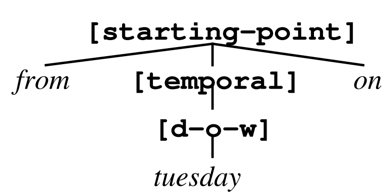
Semantic grammars (i)

- nonterminal symbols are concepts
- domain-dependent
- semantics are simply read off the parse tree
- e.g. *Auntie is coming to town*



Semantic grammars (ii)

- terminals (words) and nonterminals (concepts) freely mixed in rules, e.g.
  - `[starting-point] → from [temporal] on`
  - `[temporal] → *on [d-o-w]`
  - *from Tuesday on*



JANUS grammars

- semantic
- grammar represented as recursive transition networks (RTNs)
- SOUP stochastic parser (probabilistic RTNs)
- parse = path through RTNs that consumes input words

Grammar formalism

Context-free grammar rules of the form

head → body

where

- head ::= nonterminal
- body ::= body-elt body | body-elt
- body-elt ::= token | *token
  | +token | *+token
- token ::= nonterminal | word
- nonterminal ::= concept
  | aux-nonterminal

Grammar fragment

{**E/G/S/J**}**SST** grammars: about **150** concepts, **600** auxiliary nonterminals, **800-1600** terminals. E.g. concept **[reject]** for ESST:

```
[reject]
   (*[neg_babble] +NO *I_NEG_AUX)
   ([temporal] *just BE *just NO)
   ([temporal] *just NEG_BE looking good)
   (not [temporal])

NO
   (nope)
   (no *good)
   (not good)
   (*I_BE *very sorry)
   (*THERE_BE no way)
   (MAYBE not)
   (*THAT_BE out of the question)

MAYBE
   (maybe)
   (perhaps)
```

(...)

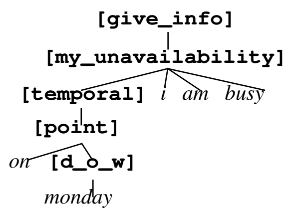
Languages covered

- **Analysis**
  - English
  - German
  - Spanish
  - (Japanese)
- **Generation**
  - English
  - German
  - Spanish
  - Italian
  - Japanese
  - Korean
  - Chinese

Example

Input: *On Monday I'm busy*

Parse:



Generations:

English = >I'm busy Monday <
German = >Leider kann ich montag nicht <
Spanish = >Yo estoy ocupado lunes <
Italian = >Sono occupato luned <
Japanese = >Getsuyoobi wa tsugoo ga tsukanai n desu ga <
Korean = >Jeo_neun ueol_yo_il an_doib_ni_da. <
Chinese = >Xing1 q y wo3 mei2 kong4 <
Hanzi = >星期一 我没空 <

Carnegie Mellon

Some results

Table 1: Performance on transcriptions

language pairs	*translation accuracy*
English → German	88.3%
Spanish → English	81.4%
German → English	75.5%
Korean → Korean	80.6%

Table 2: Performance on speech (WA ~70%)

language pairs	*translation accuracy*
English → German	60.5%
Spanish → English	73.3%
German → English	66.4%
Korean → Korean	50.0%

Carnegie Mellon

The ATIS task

Air Travel Information Service

- ARPA funded, 1989-1995
- speech → text → semantic representation → SQL → info
- funded: BBN, CMU, MIT, SRI
- volunteer: AT&T, MITRE, UNISYS
- standardized training and test sets
  - set A: context-independent queries
  - set D: context-dependent queries
  - set X: unanswerable, e.g. info not in database
- automatic evaluation
  - *principles of interpretation* set minimal and maximal reference answers (e.g. *late evening*)

Carnegie Mellon

Final results (i)

Performance on transcriptions
(set A, set D, **total**)

system	*query accuracy*
AT&T	96.2% 90.9% **94.1%**
BBN	90.6% 79.1% **86.1%**
CMU	96.2% 89.9% **93.7%**
MIT	95.5% 88.9% **92.9%**
MITRE	69.4% 41.1% **58.3%**
SRI	93.0% 83.6% **89.3%**
UNISYS	76.4% 51.2% **66.5%**

Carnegie Mellon

Final results (ii)

Performance on speech
(set A, set D, **total**)

system	*word accuracy*			*query accuracy*		
AT&T	97.4%	96.3%	**97.0%**	93.0%	88.2%	**91.1%**
BBN	97.2%	96.5%	**97.0%**	88.1%	76.3%	**83.5%**
CMU	97.0%	96.2%	**96.7%**	92.6%	86.8%	**90.3%**
CMU*	97.3%	96.3%	**96.9%**	93.5%	88.2%	**91.4%**
MIT	95.9%	94.7%	**95.6%**	89.7%	82.6%	**86.9%**
MITRE	86.6%	84.5%	**85.9%**	55.1%	28.6%	**44.7%**
SRI	97.7%	97.8%	**97.7%**	89.4%	81.5%	**86.3%**
SRI*	98.1%	98.0%	**98.1%**	90.3%	82.6%	**87.3%**
UNISYS	96.9%	95.3%	**96.4%**	72.6%	49.8%	**63.7%**

*: with NL rescoring of N-best

Carnegie Mellon

CMU

Philosophy

- SLU as information extraction
- robustness, graceful decay

Modus operandi

- speech input → decoder + trigram LM → best hypo
- parser + semantic grammar → interpretation
- backend looks at current context (i.e. history of interpretations) → SQL query or clarification dialogue

Carnegie Mellon

Speech decoder (i)

- named ‘SPHINX-II’
- semi-continuous HMMs to model context-dependent phones
- 4 codebooks (Mel-scale cepstra, energy, 1st and 2nd derivatives)
- 3-step search:
  - forward Viterbi + bigram LM → word lattice multiple end times
  - backward pass → word lattice multiple start times
  - A* extracts N(=100) best hypos

Carnegie Mellon

Speech decoder (ii)

- male and female acoustic models run in parallel → two N-best lists merged → best scoring utterance selected
- ATIS (22k utts) and WSJ (36k utts) data mixed in 9:2 ratio for training acoustic models
- recognition dictionary
  - 3207 words (283 alternate pronunciations)
  - 10 non-speech events
- class-based back-off trigram LM (1573 classes)

Carnegie Mellon

NL system

- named ‘PHOENIX’; written in C
- semantic grammars
- about 70 concepts
- RTNs
- Viterbi beam search
- parsing score (number of content words accounted for) efficiently (only if 0 < content errors < 4) used to re-rank N-best list

Carnegie Mellon

Backend ‘discourse’ module

- recognizes unanswerable queries
  1. unparsed input → clarification dialogue (not fully implemented), e.g.
     - user: *I want to go from Pittsburgh to New York *and *Boston*
     - system: *Please specify whether Boston is an arrival, departure or stop city*
  2. meta-knowledge, e.g.
     - user: *I want to go from Boston to Tokyo*
     - system: *Sorry, I don’t know about Tokyo*

Carnegie Mellon

Performance

- NL rescoring of N-best list
  - no change in word accuracy (sets A+D+X)
  - brings query accuracy from **90.3%** to **91.4%**
  - 11% reduction of total error
  - 32% reduction of errors introduced by misrecognitions
- NL error breakdown

Table 1: Error types in NL system (sets A+D)

type	*proportion*
grammar (wrong parse)	42%
context handler (this utt)	28%
context handler (previous utts)	9%
backend (wrong SQL query)	14%
semantics (inexpressible, e.g. *next earlier flight*)	7%

Carnegie Mellon

AT&T

Philosophy

- SLU as concept decoding
- flat parsing through conceptual HMM

Components

- speech decoder
  - variable n-gram stochastic automata
- NL system
  - lexical analyzer
  - conceptual decoder
  - template generator
  - dialogue manager / interpreter
  - relational database interface

Carnegie Mellon

Speech decoder (i)

- named ‘CHRONUS’
- 39 features (12 mel-cepstra, energy, 1st and 2nd derivatives)
- 20ms window shifted at 10ms intervals
- gender-dependent, continuous tri-state HMMs
- state tying through ‘ensemble merging’ (sharing of HMM state parameters)
- gender classifier (98.5% accuracy)
- lexicon of 1530 words, no alternate pronunciations
- training with ATIS (20k utts) and WSJ (8k utts) data

Carnegie Mellon

Speech decoder (ii)

- **variable n-gram stochastic automata** (VNSA) as LM
  - decoder algorithm independent of LM (order and type)
  - unified representation of n-gram model
  - $P(w_1...w_n)$ derived from VNSA state sequence probabilities
  - efficient search (Viterbi, forward) since best VNSA path corresponds to MAP criterion (i.e. $w_1...w_n$ as argmax $P(o_1...o_m|w_1...w_n)P(w_1...w_n)$)

Carnegie Mellon

Speech decoder (iii)

- **sketch**: each state corresponds to a sequence of m words observed in training (m=history window size<n) two kinds of transitions: given current word sequence and state
  - *type 1 transition* (not always possible): consume input word and {shift history window or increase history window (m++)}
  - *type 2 transition* (always possible): decrease history window (m--) i.e. backoff
- reference
  - Riccardi, G., Pieraccini, R. and Bocchieri, E. ‘‘Stochastic automata for language modeling’’, *Computer and Language*, 1996, **10**, 265-293
- VNSA used: n=3, 13 classes, about 100 compounds (e.g. *thank you*)

Carnegie Mellon

NL system (i)

- named ‘CHRONUS’, too
- lexical analyzer
  - word sequence → category lattice
  - e.g.
    - *flight*, *flights* → **`flight(s)`**
    - *dallas fort worth* →

      `(<city>DDFW)`

      `(<city>FDFW)`

      `(<airport>DFW)`
- conceptual decoder
  - conceptual HMM
  - segmentation into concepts
  - e.g. question, subject, origin, otime

Carnegie Mellon

NL system (ii)

- template generator
  - fills in feature-value pairs , e.g.
    - question {**show(s) me all**} →
      QUESTION = displa y_all
    - otime {**leaving** ... **before (<num>9.30)**} →
      DEP ARTURE_TIME = < {9.30 | 21.30}
- dialogue manager / interpreter
  - keeps context template to specialize queries; hierarc hy of concepts
    - if concept named in new sentence with different value → drop lower concepts from context
  - most human labor intensive module
- relational database interface
  - current context template → SQL statement

Carnegie Mellon

HMM (i)

Discrete **hidden Markov model**
$\lambda = (A, B, \Pi)$

- states: $s_1, ..., s_N$
- observation symbols: $v_1, ..., v_M$
- $A=(a_{ij})_{NxN}$: state transition probability distribution
  - $a_{ij} = P(q_{t+1} = s_j \mid q_t = s_i)$
  - $\forall_{i,j}\ a_{ij} > 0$ → ergodic model
- $B=(b_i(k))_{NxM}$: observation probability distribution
  - $b_i(k) = P(v_k \text{ at } t \mid q_t = s_i)$
- $\Pi=(\pi_i)_N$: initial state probability distribution
  - $\pi_i = P(q_0 = s_i)$

Carnegie Mellon

HMM (ii)

Three basic problems

- find probability of given observation sequence
  - forw ard-bac kw ard
- find most likely state sequence given observation sequence (i.e . recognition)
  - Viterbi
- find model parameters λ that maximize probability of observation sequence (i.e . training)
  - Baum-W elch (EM), gradient
- reference
  - Rabiner , L. ‘‘A tutorial on hidden Markov models and selected applications in speec h recognition, ” *Proceedings of the IEEE* , vol. 77, no. 2 257-286, F eb. 1989

Carnegie Mellon

Conceptual HMM (i)

- SLU as a decoding problem
  - speec h decoding

$$P(W|A) = \frac{P(A|W) \cdot P(W)}{P(A)}$$

find ***W*** as **argmax{ *P*(*A*| *W*)*P*(*W*)}**

  - meaning decoding

$$P(W, C|A) = \frac{P(A|W,C) \cdot P(W|C) \cdot P(C)}{P(A)}$$

$$P(A|W,C) = P(A|W)$$

find ***W***, ***C*** as
argmax{ *P*(*A*| *W*)*P*(*W*| *C*)*P*(*C*)}

acoustics , syntax, semantics

Carnegie Mellon

Conceptual HMM (ii)

- states model concepts, N=50
- observation symbols are words (more exactly, categories from lexical analyzer), M=750
- state transition probabilities given by concept bigram
  - $a_{ij} = P(c_i|c_j)$
- observation probabilities given by concept-dependent word bigram
  - $b_i(k, w_{r-1}) = P(w_k|w_{r-1}, c_i)$ where w_{r-1} is word previous to current word

Conceptual HMM (iii)

- supervised smoothing of bigrams
  - increased performance if concept-dependent synonym classes
  - e.g. under concept origin categories **`depart(s)`**, **`leave(s)`**, **`arrive(s)`**
  - manual compilation of such classes
- iterated training on correct segmentations (as automatically compared to reference answer)
- problem sentences corrected by hand

Conceptual HMM (iv)

- segmentation (i.e. flat parsing) does not always work
  - *Is the earliest flight going to Boston on a 747?*
  - *Is the earliest flight on a 747 going to Boston?*
- but good results for ATIS task
- NL error breakdown

Table 1: Error types in NL system (sets A+D)

type	*proportion*
limitations of flat semantic representation	37%
interpreter dropping current context too soon	5%
"general imperfectness of the system"	58%

BBN

Philosophy

- NLU system based on statistical machine learning techniques
- "nearly all the knowledge required for NLU is learned from training examples"

Modus operandi

- sentence → initial semantic structure → normalized semantic structure → complete meaning
- for initial search: hidden understanding model (HUM) (nested HMMs)
- for normalization and discourse search: non-statistical, rule-based procedures (due to time constraints)

Hidden understanding model (i)

- language inherently nested → embedded HMMs
- requirements for a HUM system
  - **semantic representation**: concept labels
  - **statistical model relating semantics and words**: nested HMMs
  - **training of model parameters given sentence-meaning pairs**: counts, back-off smoothing
  - **search of most likely meaning given sentence**: Viterbi

Carnegie Mellon

HUM (ii)

- replaces POS tagger, parser and semantic interpreter of a classical system
- again

$$P(C|W) = \frac{P(W|C) \cdot P(C)}{P(W)}$$

find C as **argmax{** $P(W|C)\,P(C)$**}**

- P(C): *what to say*, semantic LM
  - for each *concept*: fully-connected subnetwork of *component concepts*
  - probability on each arc: P(state$_n$ | state$_{n-1}$, context), e.g. P(dest | origin, flight) > P(origin | dest, flight)
  - i.e. any ordering of component concepts allowed, but most common orderings are most likely

Carnegie Mellon

HUM (iii)

- P(W|C): *how to say it*, lexical realization model
  - concept-dependent word bigram: P(w$_n$ | w$_{n-1}$, context)
- training: 10k hand-annotated ATIS sentences
  - bootstrapping
  - 4 undergrads 10 weeks part-time
  - double check

Carnegie Mellon

Rest of NL components

- *normalization search*: semantic parse tree (synchronous alignment to input sentence) → normalized semantic structure (NSS) (asynchronous alignment)
- if class A sentence, NSS = complete meaning
- if class D sentence, *discourse search*: discourse state (previous NSSs) + new NSS → complete meaning
- discourse state: triggers (e.g. NSS contains both origin and destination) + actions (e.g. clear discourse state)

Carnegie Mellon

Performance

- NL error breakdown

Table 1: Error types in NL system (sets A+D)

type	*proportion*
hidden understanding model	40%
other components	60%

MIT

- speec h decoder
- NL system

Speech decoder

- named 'SUMMIT'
- 2460 words
- 22k utts for acoustic training
- A* searc h: input + context-independent phone model + class bigram → word-lattice
- A* searc h: word-lattice + class 4-gram → N(=50) best hypos
- N-best re-ranking through gender - and context-dependent boundary models → 5-best
- NL re-ranking through parsability and satisfaction of pragmatic constraints

NL system (i)

- named 'TINA'; written in LISP
- probabilistic RTN: for eac h RTN
  - $S = \{s_1,...,s_N\}$: set of distinct symbols appearing in any RHS
  - $A = (a_{ij})_{NxN}$: transition probability matrix $a_{ij} = P(s_i \mid s_j)$
- trained by correct parses
- effect of merging of rules , e.g. X → A B C , X → B C D becomes X → (A) B C (D)
- *cross-pollination effect* : different rules contribute to same transition probabilities

NL system (ii)

- top-down search
- handling of constraints (e.g. agreement constraints, semantic restrictions, long-distance movement) through single-variable unification
- (generation of synthetic data for LM)
- (displayless ATIS: PEGASUS)
- potential incorporation of NL component in second A* search (problem: how to estimate score of *total* parse (i.e. h' in A*'s f' = g + h', h' should never overestimate h (cost)))

Carnegie Mellon

Performance

- NL error breakdown

Table 1: Error types in NL system (sets A+D)

type	*proportion*
lack of grammar coverage	23.1%
bugs in the code	5.8%
illegal SQL query	9.6%
discourse	36.5%
inexpressible in chosen semantics	25.0%

Carnegie Mellon

SRI

- speech decoder
- NL system

Carnegie Mellon

Speech decoder

- named 'DECIPHER'
- 39 features extracted every 10ms: 12 cepstra, energy, 1st and 2nd derivatives
- gender-dependent, context-dependent HMM
- 4-pass search
  1. gender classification (99% accuracy)
  2. forward-backward + bigram → word lattice
  3. forward-backward constrained by lattice → N(=1300) best hypos
  4. N-best list rescoring with 5 knowledge sources
     1. acoustic model (ATIS)
     2. acoustic model (ATIS+WSJ, 10:1 ratio)
     3. bigram
     4. class-based trigram (35 classes)
     5. word count (to prevent excessive insertions)

Carnegie Mellon

NL system

- named 'GEMINI'; written in PROLOG
- unification-based, semantic rules, bottom-up chart parser
- NL as knowledge source to constrain decoder search
- robustness
  - combination of fragmentary parses
  - repair spotter
  - template matcher as fall-back
- 'discourse' module looks at context and constructs model of the user's itinerary

Carnegie Mellon

Constraining recognition (i)

- integration of linguistic and statistical factors

  1. parser analyzes hypo as sequence of semantic chunks (flat parsing)
  2. probability of hypo under that semantic sequence estimated from n-gram model

  - trigram of the 5 top-level semantic chunks (TLSC) (*sentence*, *noun phrase*, *modifier phrase*, *filler* and *skipped*)
  - single 4-gram model at the word and word-class levels (intra-TLSC model too sparse) but begin-TLSC tokens inserted, e.g. `begin_sentence`

Carnegie Mellon

Constraining recognition (ii)

- cannot replace traditional LM
- but combination is better
  - word accuracy from **97.7%** to **98.1%** (i.e. 17% error reduction)
  - utterance accuracy from **84.8%** to **87.2%** (i.e. 16% error reduction)
  - query accuracy from **86.3%** to **87.3%** (i.e. 10% error reduction)

Carnegie Mellon

Robustness

- if fail to find complete analysis of utt, find sequence of TLSCs
- Viterbi search to find analysis with fewest skipped words
- if still tie, use score from linguistic and statistical LM
- combine resulting TLSCs
  - find minimal set of relations from 'database graph' (ontology) that allows combination of TLSCs present, e.g.
    - **`[what types of aircraft] [can i get a first class ticket from philadelphia to dallas]`**
    - frame **flight** introduced to relate *aircraft* with *first class ticket*

Carnegie Mellon

‘Discourse’ module

- context-handling mechanism
- attempts to recognize user’s intended itinerary
- associates constraints with each itinerary leg
- useful in understanding implicit info, e.g. *the return flights*
- can resolve some anaphoric references
- query specialization

MITRE

Toolkit architecture, 3 modules:

- speech recognition system
- NL system
- discourse module

But written in different languages and running on different machines ... how to combine them?

- ‘COBRA’ common object request broker architecture, each module is both client and server

Speech decoder

- named ‘GLACIER’
- single-pass Viterbi beam search, constrained by LM (smoothed class bigram)
- 3-state context-independent HMM (48 phone-like units + 1 for silence)
- high-pass filter, 8th order PLP analysis every 10ms, 16ms window
- 18-D feature vector
- output: word lattice

NL system

Composed of 5 submodules

- lexicon, parser, generator
- knowledge representation
- discourse processor (resolves anaphora, tracks task)
- backend application query and command module
- presentation planner

Parsing module

- PHOENIX parser, very much undeveloped grammar
- augmented by semantic interpretation rules in LISP
- **Interpretation** (meaning representation) defined as the standard interchange object among NL submodules

Discourse module(i)

- does not handle full range of context-dependent NP or VP ellipsis, only those that
  - occur in the data
  - are expressible in the **interpretation** formalism
- does not build new interpretation, can only *specialize* referent, i.e. tell SQL query generator to restrict search to tuples returned by previous query
- e.g.
  - *Show me all QUICKWING flights from Barcelona to Shekou*
  - *Arriving on a Tuesday*

Discourse module (ii)

- each constraint of each new utterance classified as
  - *novel*, *repeating* or *conflicting*

Table 1: Discourse types in set D

type	*proportion*
specialization	41%
conflicting	24%
other	35%

- some conflicting constraints are specializations if enough world knowledge, e.g.
  - *Show me all QUICKWING flights from Barcelona to Shekou*
  - *Arriving after 4pm*
  - *Arriving after 5pm*

Conclusions

- exercise in connecting multiple, independent components (toolkit architecture)
- poor results due to
  - inadequate development of each component
  - error propagation in serial architecture, e.g. discourse depends on parser, parser on speech decoder

References (i)

- **JANUS, PHOENIX**
  - Lavie, A. et al. "JANUS-III: speech-to-speech translation in multiple languages," *Proceedings of ICASSP-97*
  - Waibel, A. et al. "JANUS-II: translation of conversational speech," *Proceedings of ICASSP-96*
  - Mayfield, L., Gavaldà, M., Ward, W. and Waibel, A. "Concept-based speech translation," *Proceedings of ICASSP-95*
  - Issar, S. and Ward, W. "Unanswerable queries in a spontaneous speech task," *Proceedings of ICASSP-94*

References (ii)

- **ATIS** (from *Proceedings of the spoken language systems technology workshop*, ARPA, Morgan Kaufmann, 1995, unless noted otherwise)
  - **AT&T**:
    - Pieraccini, R. and Levin, E. "A learning approach to natural language understanding," *New advances and trends in speech recognition and coding*, NATO ASI Series, Springer-Verlag, 1993
    - Bocchieri, E., Riccardi, G. and Anantharaman, J. "The 1994 AT&T CHRONUS recognizer"
    - Levin, E. and Pieraccini, R. "CHRONUS, the next generation"

References (iii)

- **BBN**:
  - Miller, S. et al. "Recent progress in hidden understanding models"
  - Miller, S., Bobrow, R., Ingria, R. and Schwartz, R. "Hidden understanding models of natural language," *Proceedings of ACL-94*
- **CMU**: Ward, W. and Issar, S. "The CMU ATIS system"
- **MIT**:
  - Glass, J. et al. "The MIT ATIS system: December 1994 progress report"
  - Seneff, S. "TINA: a natural language system for spoken language applications," *Computational Linguistics*, 1992, vol. 18, no. 1, pp. 61-83

References (iv)

- **MITRE**: Bayer, S. et al. "Spoken language understanding: Report on the MITRE spoken language system"
- **SRI**:
  - Cohen, M., Rivlin, Z. and Bratt, H. "Speech recognition in the ATIS domain using multiple knowledge sources"
  - Moore, R. et al. "Combining linguistic and statistical knowledge sources in natural-language processing for ATIS"
- **UNISYS**: Dahl, D. et al. "Weakly supervised training for spoken language understanding systems"
- **http://xxx.lanl.gov/cmp-lg/**