Extracting Information from Spoken User Input

Piroska Lendvai
Tilburg University

19 April 2005

Outline

- Two-party communication: Dialogue
  - Pragmatics of dialogue
  - Semantics of dialogue
  - Human - machine dialogue
- A case study on the OVIS corpus
  - Detection of miscommunication in dialogue
  - Extraction of pragmatic and semantic information from dialogue
  - ML issues: Co-learning of dialogue phenomena

Aspects of dialogue

Spoken Dutch corpus

MapTask, ATIS

3 **Philip**:
{fg|Eh}, you have the missionary camp in the bottom right-hand corner, is that right?
4 **Martin**:
Yeah, I've got a start.
5 **Philip**:
Right. {fg|Eh}, if you take ... If you go about an inch exactly to the left of that and stop.
6 **Martin**:
Okay.
7 **Philip**:
{fg|Eh}, now you're heading diagonally down and to the left about two inches. {fg|eh}, {ip|til=until} you're within about three quarters of an inch of the bottom of the page. And at the bottom left-hand corner of the missionary camp.
8 **Martin**:
Okay.
9 **Philip**:
Right. {fg|Eh}, to the left have you got gorillas at the bottom left-hand corner or any sort of
10 **Martin**:
I've got a banana tree.
11 **Philip**:
Banana tree, right. Okay. {fp|Erm}, if you head left about four inches {ip|til=until} you're at the bottom left-hand side of that marker, that should leave you within an inch of the left-hand side of the page and about three quarters of an inch up from the bottom of the page.

AirTravel Information System

- User: I WANT TO GO TO SAN FRANCISCO.
- ***ATIS****: Where from?*
- User: BOSTON.
- ***ATIS****: What date will you be travelling on?*
- User: I'LL BE LEAVING BOSTON NEXT SUNDAY AND RETURNING THE FOLLOWING TUESDAY.
- ***ATIS****: These are the flights from Boston to San Francisco on Sunday January 6. (…)*
- User: WHERE DOES THE THIRD ONE STOP?
- ***ATIS****: American flight 813 from Boston to San Francisco on Sunday January 6 stops in the following places. (…)*
- User: WHAT IS THE CHEAPEST FARE FOR THE EARLIEST NONSTOP FLIGHT THAT SERVES DINNER?
- ***ATIS****: This is the cheapest round-trip fare for the earliest non-stop flights from Boston to San Francisco serving dinner on Sunday January 6. (…)*
- User: BOOK IT.

Properties of dialogue (i)

- Communicative function (purpose): small talk, information exchange, job interview, task solving
- Form: how is the purpose expressed? Dependent on
  - (Artificial) intelligence: human-human vs human-machine, expert vs naive
  - Modality (visual instruments): eye contact yes/no, facial expression, gestures, spoken vs written style, telephone, keyboard
  - Situation: spontaneous/relaxed vs directed/formal
  - Structure: Turn taking, multi-party

Properties of dialogue (ii)

- What information can we draw on when understanding the other party?
  - Language (verbal instruments):
    - Syntax: well/ill-formed
    - Prosody: loudness, duration, pitch
    - Word usage: large vocabulary vs restricted vs jargon
  - Cognitive processes involved: world knowledge, reasoning
  - Multi-modality (face, eyes, hands, pointing)

Pragmatics of dialogue

- Speaker's intentions are manifested by his utterance
- Intentions are formed by and dependent on the situation
- Intentions are referred to by the term "speech act" or "dialogue act"
  - "My name is Piroska." *inform*
  - "My name is Bond." *inform + threat/joke*
- Computational pragmatics: detection and processing of dialogue acts
  - Discovery of underlying mechanisms in dialogue
  - Interpretation of dialogues
- Annotation frameworks (DAMSL, Switchboard, MATE)
- Hierarchy and granularity of dialogue acts
  - ("I won't go tomorrow.": *answer, inform: statement, repeat_statement, influence_addressee_future_action*)

Semantics of dialogue

- Utterance content
- Highly relevant for task-oriented dialogue: factual values that exist in the world independent of the context of the dialogue
- Need to be extracted to reach the dialgoue's goal
- In task-oriented dialogue may be referred to as "slots"
  - "go to the left of the missionary camp": *go(left_of,missionary_camp)*
  - "I'd like to book a trip to Northern Italy": *interest(trip,Northern_Italy)*
  - Semantic parser: a tool that detects such information units
- Involves both segmentation and value extraction
  [I'd like to book] [a trip] [to Northern Italy]
- Hierarchy and granularity of slots
  [I'd like to book] - *dialogue act*
  [a trip] - (vs accomodation vs transport) *slot supercategory, "topic"*
  [to Northern Italy] - *slot value*

Human - machine dialogue: Practical use

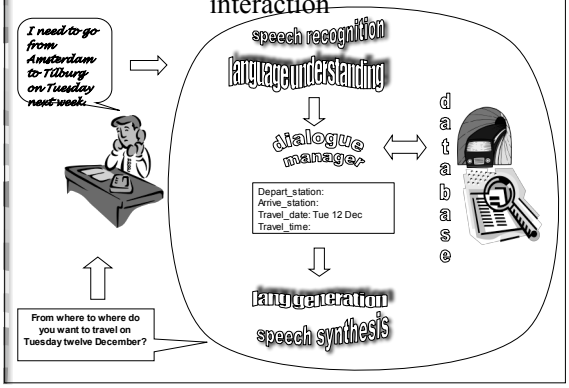
- Spoken dialogue systems (SDS): software that communicates with a user in order to perform some task (e.g. book / inform about flights in database). AKA "information systems", "conversational agents"
- Save money with automated telephone interfaces that provide info (calling, transport, weather, booking, banking)

  Ook na twaalven reisinformatie?
  Bel met de sprekende computer van 9292 via **0900-1475 (€ 0,35 p/m)**, voor het plannen van uw treinreis en voor informatie over werkzaamheden en vertragingen.
- Create automated help systems/manuals
- Voice control in smart household appliances / industrial robots / research
- (Provide support for customers in) using e- commerce
- E-mail, voice mail access

Human - machine dialogue: Technology

- Create systems that enable interaction with an application (eg. software, TV, database) using natural language through a voice interface
- From 60s: communication with a machine in natural language. Advances in speech technology facilitate development of SDS
- Ever-going progress in NLP: spoken language understanding, reasoning
- Dialogue is task-oriented: restricted vocabulary and limited amount of moves (dialogue acts)
- Popular technology: slot-filling

A typical human - machine interaction



Human - machine dialogue: Challenges

- Man-machine communication demands combining techniques of speech analysis and generation + linguistic analysis/generation (syntax, semantics) + task planning (dynamically changing context)
- Must interpret subtle and implicit dialogue acts (does user request / provide / acknowledge info, correct misunderstanding, etc.)
- Must recognise problematic situations and recover from those
- Must extract semantic values from the user's utterance
- Future (?): Must meet discourse requirements: satisfy broad social obligations, handle world knowledge

The OVIS system

- Developed in 1995-2000, NWO project
- Openbaar Vervoer Informatiesysteem
- 80 test users
- Noisy real-data Dutch corpus
- 441 full dialogues
- 3,738 turns of system prompt - user reply
- Slots to fill: DepartStation, ArriveStation, Day_of_Travel, Time_of_Travel
- System always verifies received info
  - explicitly ("So you want to leave on Thursday.") or
  - implicitly (" At what time do you want to leave on Thursday?")

OVIS example

S1: Good evening. From which station to which station do you want to travel?
U1: I need to go from Schiphol to Nijmegen on Tuesday next week.
S2: From where to where do you want to travel on Tuesday twelve December?
U2: From Schiphol to Nijmegen.
S3: At what time do you want to travel from Schiphol to Nijmegen?
U3: Around quarter past eleven in the evening.
(…)
S5: I have found the following connections: (…). Do you want me to repeat the connection?
U5: Please do.
…

Miscommunication in human-machine spoken dialogue

Communication problems

- Frequent occurrence of communication problems btw system and user in SDSs
- User input is erroneously processed
  - ASR: hyperarticulated speech, dialect, noise, OOV word
  - NLU: ungrammatical user input, self-corrections
  - DM: incorrect assumptions or grounding
- Prompt generated is improper
- User gets distracted or frustrated

Automatic detection of miscommunication

- Statistical approaches are widely applied for automatic problem detection
- Case study: Machine learning of miscommunication with a train timetable SDS
- Achieving robustness of method:
  - Automatically extractable features
  - Algorithm choice
  - Class engineering

Assessment of SDS performance

- Word accuracy
  - Percentage of words correctly recognised by SDS
- Concept accuracy
  - Percentage of semantic concepts (e.g. departure station) correctly recognised
- In our study: Miscommunication (Problem) =
  Lack of *full* concept accuracy
- Defined on a dual time line:
  - Problem origin
  - Awareness of problems

ML of miscommunication

- Supervised learning: a data-driven AI method
  - Learners are able to extract knowledge from examples, and to improve with experience
- Annotated corpus is required for training learners
- Training examples: human-machine dialogue turns converted into a fixed-length vector of dialogue features + class (to be assigned)

Class design (1)

- What do we want to learn?
  - Granularity issue: Research often models some subclass of miscommunication:
    - Poor speech recognition
    - Mismatch/partial match btw user input and ASR output
    - End task success/failure
    - Erroneous system grounding
    - Types of user reactions to errors
  - Encoding such classes is often not trivial

Class design (2)

- Some dialogue act taxonomies (e.g. DAMSL) partially cover miscommunication:
  - REJECT ('Well, no.')
  - SIGNAL NON-UNDERSTANDING ('Excuse me?')
  - APOLOGY ('I'm sorry.')
- Merge all such cases in binary class
- Assigned class per training example: Problem/OK

Problem origin (i)

- Predict problems originating in *current dialogue turn* (having consequences in next turn)
- Example: OVIS train travel SDS:
  User t1: 'I need to go from Schiphol to Nijmegen on Tuesday next week.'
  System t2: 'From where to where do you want to travel on Tuesday twelve December?'
- Miscommunication can only be inferred from next prompt, but this is not yet known

Problem origin (ii)

- Usefulness of detecting PROB ORIGIN:
- ASR has more confidence in accepting/rejecting recognition hypothesis
- DM can adapt strategy to a more optimal one
  - Re-prompt for same input
  - Launch a differently trained ASR
  - Switch to explicit prompting strategy

Problem awareness (i)

- Information received is grounded
  - System:
    - 'Go on' (implicit prompt, ask for next info piece)
    - 'Go back' (explicit prompt, meta-prompt, apology)
  - User feedback
    - 'Go on' ("yes", give info that was asked for)
    - 'Go back' ("no", "incorrect", "not to Amsterdam")
- Goal: Detect incorrect grounding by system,
- Based on user's awareness in this

Problem awareness (ii)

- 'Awareness site': user becomes aware of miscommunication
- Often signalled by user's negative feedback
  - Wording
    - Correction, Rejection, Info Repetition *('Not this Tuesday but Tuesday next week')*
  - Prosody (high pitch, slower tempo, longer input)
- Usefulness: DM can launch error recovery

Annotation: *ORIGIN; AWARENESS*

S1: Good evening. From which station to which station do you want to travel?
U1: I need to go from Schiphol to Nijmegen on Tuesday next week.
ORIGIN >> Prob **AWARENESS >> OK**

S2: From where to where do you want to travel on Tuesday twelve December?
U2: From Schiphol to Nijmegen.
ORIGIN >> OK **AWARENESS >> Prob**

S3: At what time do you want to travel from Schiphol to Nijmegen?
U3: Around quarter past eleven in the evening.
ORIGIN >> OK **AWARENESS >> OK**

S5: I have found the following connections: (…). Do you want me to repeat the connection?
U5: Please do.
ORIGIN >> OK **AWARENESS >> OK**

How to interpret automatically?

- Baseline strategy: predict that user input will be the one which is most likely to be triggered by the system prompt

```
If system_input is "Q_DepArr" ('Good evening. From
  which station to which station do you want to
  travel?')
Then always predict: ProbOrigin_OK (ProbAwareness_OK)
```

- Real annotation: *ProbOrigin_Prob* *(ProbAwareness_OK)*

('I need to go from Schiphol to Nijmegen on Tuesday next week')

Assessment: Testing

- Baseline scoring in evaluative test

	Acc	*Prec*	*Rec*	$F_{\beta=1}$
ProbOrigin	64.8	61.3	50.6	55.3
ProbAwareness	86.2	96.2	70.7	81.3

ML experiments

Classification carried out by two orthogonally different learning algorithms:

- Memory-based learning (MBL):
  - classifies new input based on similarity with stored examples
  - *k*-nearest neighbour classifier implemented in TiMBL
- Rule induction (RI): classifies new input by rules that were induced from examples, implemented in RIPPER

Experimental set-up

- ProbOrigin and ProbAwareness are learnt in separate experiments
- 10-fold cross-validation
- Algorithm parameter settings optimised with heuristics (wrapped progressive sampling; Antal Van den Bosch 2004)

Features of ML examples (i)

- Other studies: Information sources used are sometimes very high level
  - inconsistency btw system prompt and user answer, topic shifts, age, gender, input word order
- Here: MBL and RI utilise linguistically unsophisticated information, automatically extracted from SDS
- Speech signal measurements: Acoustics, Prosody
- Recognition hypotheses: Bag-of-words (BoW), recognition confidence
- Dialogue context in terms of prompt type history: 10 system prompts represented as structured symbols
  - *"From where to where do you want to travel on Tuesday twelve December?"* >> ImplVerif_Day; Q-DepArr
  - _, _, _, _, _, _,Q-DepArr, RepQ-DepArr, ImplVerDep;Q-Arr, ImplVerArr;Q-Day

Features of ML examples (ii)

- Features of the user input, coming from ASR
  - Acoustics/prosody (voice pitch, loudness, duration, tempo, pausing)
  - Recognised hypotheses (bag-of-words, confidence-based scores)
- *"ik moet volgende week dinsdag van Schiphol naar Nijmegen"*

```
> ik moet volgende week dinsdag van schiphol maar nijmegen
> ik moet op dinsdag van schiphol naar nijmegen
```

BoW: >> ik, dinsdag, maar, moet, naar, nijmegen, op, schiphol, van, volgende, week

Experimental Results

		Acc	*Prec*	*Rec*	$F_{\beta=1}$
ProbOrigin	*baseline*	*64.8*	*61.3*	*50.6*	*55.3*
	MBL	68.1	67.7	49.4	57.0
	RI	64.8	63.9	54.9	54.8
ProbAwareness	*bline*	*86.2*	*96.2*	*70.7*	*81.3*
	MBL	89.9	95.0	80.7	87.2
	RI	90.5	92.4	85.1	88.5

ProbOrigin - induced rules

- Specific and complex situations
- All feature types are used

- If 'naar' in_SysBoW and BranchFact > 3 and 'ik' in_UserBoW and 'herhalen' not_in_prevSysBoW then ProbOrigin.
- If 'verbinding' not_in_SysBoW and BranchFact > 2 and 'een' not_in_prevSysBoW and 'naar' in_UserBoW and loudness < 285 then ProbOrigin.
- If recognisedStringLength > 4 and 'van' in_prevSysBoW and 'herhalen' not_in_SysBoW and tempo < 2.057 then ProbOrigin.
- If 'verbinding' not_in_SysBoW and 'maar' not_in_SysBoW and 'ja' in_UserBoW and 'uur' in_UserBoW then ProbOrigin.
- Else OK.

ProbAwareness - induced rules

- All feature types are used
- First two rules cover the 'system knows' baseline

- If 'niet' in_SysBoW and 'ik' in_SysBoW then ProbAwareness.
- If 'waar' in_SysBoW then ProbAwareness.
- If 'uur' in_prevSysBoW and topConfidence > 772 then ProbAwareness.
- If 'naar' in_prevSysBoW and 'naar' in_SysBoW and prevBranchFact > 2 and '@m' not_in_prevUserBoW then ProbAwareness.
- Else OK.

ML methodological issues

- What if both problem aspects (origin; awareness) are co-learnt simultaneously
- What if miscommunication aspects are co-learnt with yet other dialogue phenomena (dialogue act; filled slots)
- During classification certain interpretation components may correlate, license, or disturb each other in classifier

Co-learning components of spoken input

ProbOrigin
ProbAwareness
DialogueAct
FilledSlot

Pragmatic and semantic classes

- Task-related dialogue act: basic action in user's utterance:
  - Slot-filling *('I need to go from Schiphol to Nijmegen on Tuesday next week.')*
  - Affirmative *('Please do.', 'Yes, indeed.', ...)*
  - Negation *('No, it's not necessary.', 'Incorrect.', ...)*
  - Acceptance of error *('Yes.', ...)*
  - Non-standard input *(silence, irrelevant info,...)*
- Slot(s) being filled by user:
  - Departure station *('from Schiphol')*
  - Arrival station *('to Nijmegen')*
  - Day *('on Tuesday next week')*
  - Time of day *('in the evening')*
  - Hour *('Around quarter past eleven')*

Co-Learning

- Example Question: If we simultaneously classify what the user is doing and whether he is aware of communication problems (task: TRA +Slot + ProbAwareness), do we get better scores than when we only have to detect that user is aware/unaware of problems (task: ProbAwareness)?
- Solution: Exhaustive search by ML experiments:

for each user input aspect {dialogue act; filled slots; prob origin; prob awareness}
select the optimal task component combination by maximising learner performance (F)

Co-learning: Tasks

- Detect four information components:
  1. Task-related act in the input (query slot filling / affirmative / negative / error acceptance)
  2. Query slot filled by input (departure / arrival station / time / hour of travel)
  3. Origin of communication problem (input will cause communication problem / OK)
  4. Awareness of communication problem (user knows there is a communication problem / OK)
- These provide a partial interpretation of the user's utterance

Co-learning: Classes

S1: Good evening. From which station to which station do you want to travel?
U1: I need to go from Amsterdam to Tilburg on Tuesday next week.
>> S_Dep-Arr-TravelDay_Prob_Ok
S2: From where to where do you want to travel on Tuesday twelve December?
U2: From Amsterdam to Tilburg.
>> S_Dep-Arr_Ok_Prob
S3: At what time do you want to travel from Amsterdam to Tilburg ?
U3: Around quarter past eleven in the evening.
>> S_TimeofDay-Hour_Ok_Ok
S5: I have found the following connections: (…). Do you want me to repeat the connection?
U5: Please do.
>> Y_void_Ok_Ok

Class label combinations

Component	*nr of classes*	*Component*	*nr of classes*
TRA	8	TRA + Slot + ProbOrigin	104
Slot	30	TRA + Slot + ProbAware	90
ProbOrigin	2	TRA + ProbOrigin + ProbAware	29
ProbAware	2	Slot + ProbOrigin + ProbAware	81
TRA + slot	63	TRA + Slot + ProbOrigin + ProbAware	148
TRA + ProbOrigin	16		
TRA + ProbAware	15		
Slot + ProbOrigin	48		
Slot + ProbAware	47		
ProbOrigin + ProbAware	4		

Co-learning: Findings

- Optimal component combinations might differ per learner (MBL: TRA + *slot*; RI: *slot)*
- After finding the optimal combination
  - Classifier performance enhances compared to "traditional" combination (MBL: TRA + *slot*, TRA + ProbAware; RI: *slot*, TRA + ProbAware)
  - The differently biased classifiers achieve same performance
- Optimal component combinations result from algorithm bias (eg., RI sometimes prefers learning less classes) AND the components' correlation
  - TRA + ProbOrigin (16), TRA + ProbAware (15) but TRA + ProbOrigin classifies much worse
- TRA is in general beneficial for all components: main, decisive action of utterance

Results

	DA	Filled Slots	ProbOrig	ProbAware
	F	F	F	F
MBL				
isolated	*91.7*	*86.7*	*57.0*	*87.7*
co-learnt	91.7	87.7	59.4	90.8
RI				
isolated	*90.5*	*85.5*	*54.8*	*88.5*
co-learnt	90.5	82.0	62.6	88.5

- Magnitude of difficulty (performance): TRA > ProbAware > Filled Slots > ProbOrigin
- ProbOrigin combines unoptimally with anything, and performance is generally low on it
- ProbOrigin has nothing to do with dialogue pragmatics or semantics

Classifying TRAs

- TRA is best learnt in isolation, or with one other component (stat. insign.)
- Performance ordering (both learners):
  Slot-filling > Affirmative > Negation > Nonstd > Acceptance
- Affirmative input much better classified than Negative
- TRA and ProbAwareness combine optimally: describe same dimension by same properties

Classifying Filled Slots

- MBL learns slots best in combination with TRA
- RI learns slots best in isolation (class amount)
- Performance ordering (both learners):
  DepartStat > NoSlot > ArrivStat > Hour > Day > TimeOfDay

Co-learning: Conclusions

- Class engineering (via co-learning different language phenomena) is an important issue in ML for NLP
- Can provide explanation about the nature of designed tasks
- Might enable improvement over traditionally established task design/order
- Enables identical performance of the two, differently biased algorithms on all four user input aspects
- Pragmatic-semantic processing tasks should sometimes be differently formulated depending on the classifier's bias
- ProbOrigin should be decomposed into more meaningful classes
- New shallow features can be introduced to the same method

Evaluation of ML performance

- Automatic extraction of high-level dialogue phenomena based on cheap info is possible with good performance
- Useful to search for optimal class component combination, possibly yielding improvement
- If ASR in OVIS would get better, interpretation scores would overall improve
  - Up to 24% error reduction if simulating perfect ASR.