Introduction to Computational Linguistics
Session 9: Dialogue Systems & Further NLP Methods

Denise Löfflad

Universität Tübingen

January 10, 2024

Exam

You need to register for the exam on Alma! Important deadlines:

- 22.01: Deadline to register for the exam (do it after the lecture)
- 02.02: Deadline to deregister from the exam. If you did not get 60% or don't want to participate, deregister! Otherwise it will count as a try, and you only have 2 tries

Plan

1. Dialogues Systems
    1. Dialogues
    2. Task-Specific DS - Digital Assistants
    3. Task-Independent DS - Chabots
2. Further NLP Methods
    1. POS Tagging
    2. Parsing

Dialogues

Different perspectives on dialogues:

- What elements do human dialogues consist of?
- What underlying rules/principles guide people's behaviour in dialogues?
- How can computers make use of these elements and principles?

Dialogues

speech act

- actions performed through language
- examples: greeting, apologising, ordering, requesting something, advising, insulting, . . .

Dialogues

common ground

- information mutually shared by participants
  1. context, e.g. the location you are in
  2. world knowledge (facts about the world)
  3. prior knowledge, e.g. earlier part of the conversation

Dialogues

turn taking

- role of the speaker and hearer(s)
- speaker: control over content, goals, formulation, style, ensuring hearer understands
- hearer: interpretation of utterances of speaker, update the common ground, plan answer, interpret conversation flow (e.g. when to switch roles)
- backchannel: " the person who has the turn receives short messages such as 'yes' and 'uh-huh' without relinquishing the turn." (Yngve, 1970, p.568)
- overlap: people participanting in an conversation and "planning to respond are already encoding their responses while the incoming turn from the other speaker is still unfinished" (Levinson and Torreira, 2015)

Dialogues

adjacency pair

- sequence of two turns with expected structure
- content and structure of second turn determined by first turn
- examples:
  - greetings elicit greetings
  - questions elicit answers
  - informational statements followed by acknowledgements
  - confusion elicits clarification
  - ...

Dialogues

Grice's Maxims: norms guiding conversations

1. **Quality:** say true things
2. **Quantity:** only as much information as necessary, not more, not less
3. **Relevance:** say only things that are relevant to the conversation
4. **Manner:** express content in a manner that can be easily understood

Dialogues

Cooperative conversations are important: allows one to make inferences

- S1: “Did you like the movie?”
- S2: “I liked the first half of it.”
- By maxims of Quality, Quantity and Relevance, we can interpret S2’s response in (at least) two ways...

Dialogues

Cooperative conversations are important: allows one to make inferences (see Textbook, p.286)

- S1: "Did you like the movie?"
- S2: "I liked the first half of it."

- **Conversational implicature:** "inference that a hearer draws from reasoning about why a speaker said one thing over another" (p.286) E.g., S2 didn't like the second half.
- **Entailment:** "an inference that follows logically from a given statement" (p.286). E.g., S2 watched the first half.
- **Presupposition:** "fact that has to be true for a given sentence to even make sense" (p. 286). E.g., S2 is talking about the same movie as S1.

Task-Specific Dialogue Systems: Digital Assistants

- dialogue system designed to help accomplish one or more specific, concrete tasks
- alternative to other ways of accomplishing a task, e.g. employee or graphical user interface
- in contrast to chatbots, designed and built for a *specific domain* and *specific task(s)*
  - good enough processing for a specific domain
  - not necessary to model a lot of world knowledge (terminology, relations between concepts, . . .) outside of application domain

Digital Assistants

- **intent**: task the user is supposed to achieve with support by the assistant
    - example: ordering a meal
- intents contain **slots**/**templates** that need to be filled
    - example: which food/drinks in which quantity
- **intent recognition**: classify user utterance as intent
    - example: ordering a meal, booking a table

Digital Assistants

Examples:

- assistants for phone-centered tasks, e.g. Siri (Apple), Alexa (Amazon)
- assistants in modern cars
- increasing number of digital assistants used for customer 'service' purposes
- . . .

Task-Independent Dialogue Systems: Chatbots

- dialogue system designed to chat with a human in natural language (sociable chit-chat)
- conversations might be carried out in written or spoken form
- designed and built task- and domain-independent with the purpose of entertaining or impressing users (e.g. customers) and/or serve as digital companion
- Systems try to:
    - respond appropriately to the human's turn
    - be polite and friendly
    - model a reasonable amount of world, prior and context knowledge (Common Ground)
    - comply with Grice's Maxims

Chatbots: Implementation

- **brute force**: using an enormous amount of hand-written question-answer pairs
- **rule-based**: using fixed rules to derive response utterances given the preceding turn
- **corpus-trained chatbots**: based on a huge set of data the bot derives the answer that is most suitable for a given turn
- **language generation**: based on corpus data machine learning models are built; at production time they take previous turn(s) as input and construct and output a response that has a high probability of being a reasonable reply (the output could be something that does not exist in the corpus)

Most modern systems use a mixture of different approaches.
It is also important to keep ethical aspects in mind, especially when using corpus based methods for implementation.

Chatbots: ELIZA

- implemented by Joseph Weizenbaum in 1964
- mimics the behavior of a psychiatrist, who is asking questions rather than uttering statements
- rule-based chatbot
  - “If the input contains a noun, reply: TELL ME MORE ABOUT [NOUN]”
  - “If the input contains the phrase I AM [X], reply: HOW LONG HAVE YOU BEEN [X]?”

Chatbots: ELIZA

ELIZA: Hello, I am Eliza
USER: Hi
ELIZA: How are you today . . . What would you like to discuss?
USER: I am studying chatbots
ELIZA: Did you come to me because you are studying chatbots?
USER: Yes!
ELIZA: I see.
USER: Can an alligator ride a bicycle?
ELIZA: Can you elaborate on that?
USER: I am trying to see how smart you are
ELIZA: Perhaps you are trying to be?
USER: What animal did I just ask you about?
ELIZA: We were discussing you, not me.
USER: Let me ask about you though!
ELIZA: We were discussing you, not me.

Chatbots: Kuki

- implemented by Steve Worswick, published 2005
- using blend of rules and hand-written responses
- initial implementation has been extended over the years based on questions from users
- different answer options for common questions have been implemented to avoid repetitions
- contains reference to a database that contains facts about world (ConceptNet)
- used by humans to practice their english, thus some rules have been adapted to learner language

Chatbots: Kuki

USER: good who else have you been talking to?

KUKI: I am talking to a bunch of people all over the world

USER: Like who?

KUKI: Joe

USER: What is he saying?

KUKI: Are you asking about Joe?

USER: Yeah, what is he saying?

KUKI: I don't want to talk about that now. Are you asking about Joe?

USER: Yes but you said you don't want to talk about that

KUKI: I don't want to talk about that now. What I said earlier is irrelevant.

USER: Okay what do you want to talk about?

KUKI: I like to talk about movies, robots, and history. I hate one-word answers.

Evaluating Chatbots

- Turing test
- Winograd Schemas: A pronoun is used with two possible antecedents, world knowledge is needed to correctly resolve the pronoun.
  In the sentence: "The city council members refused the demonstrators a permit because they advocated violence", who does "they" refer to?

And finally, a cautionary note

Microsoft's Tay

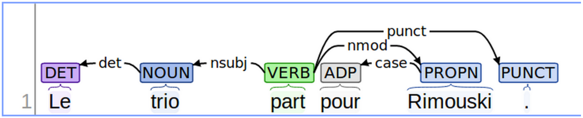
- released in 2016, Tay was programmed to keep itself up to date by learning from conversations
- within 24h hours, it had to be taken down
- didn't include a moral filter
- → be cautious and test well

For next time - Generative AI

"As one of humanity's unique tools, language is often considered key to the success of artificial general intelligence" Dickinson et al. [2012]

Further into NLP

- Beyond segmenting and tokenizing character streams, NLP also investigates methods for annotating strings with linguistic information **automatically**



- Today we introduce two common NLP functionalities that provide linguistic annotations:
  - POS Tagging
  - Parsing

What are Part of Speech (POS) Tags?

- Annotate tokens with information related to word type
- Information takes form of “tags”:
  - Short labels describing a token’s part of speech
  - Examples: Noun, verb, adjective, adverbs, . . .
- Different corpora / NLP tools use different tag sets, each with different categories, e.g.
  - Penn Treebank tag set: 45 tags
  - Brown corpus tag set: 87 tags
  - C7 tag set: 146 tags

Why POS Tags?

- Tags encode morphological and syntactic information of tokens
- Allows to draw conclusions about properties of tokens and - to a certain extent - their neighbors
    - E.g., pronunciation can change depending on part-of-speech: **con**tent vs. con**tent**
- Pre-requisite for other NLP tasks (e.g., parsing, keyword extraction, language learning exercise generation, etc.)

POS Tags - Different Granularities

- main categories: noun, verb, adverb, . . .
- fine-grained categories:
  - Noun singular or mass NN (bird, water, . . .)
    vs. noun plural NNS (birds, geese, . . .)
  - adverbs comparative RBR (better, faster)
    vs. superlative RBS (best, fastest)

POS Tags - Closed and Open Classes

- closed class part of speech tags
    - fixed set of tokens/words
    - often function words
    - shared across corpora
    - example: prepositions (on, since, from, with, . . .)
- open class part of speech tags
    - changing (language change, borrowing/coining/inventing new words)
    - depending on domain of corpus
    - example: nouns, verbs

See also https://universaldependencies.org/u/pos/

Annotating tokens with POS Tags

- different approaches towards part-of-speech tagging
- typically requires pre-processed text
    - sentence-segmented
    - tokenized
- goal: assign each token single most probable tag

POS Tagging - Ambiguity

- most words unambiguous,
  straightforward mapping from word -¿ POS tag
- a small number of common word forms are ambiguous (2 - 7 tags)
- Example:
  I can can a can of tuna.
- Brown corpus: 11.5% of types ambiguous, but 40% of tokens ambiguous

POS Tagging - Ambiguity cont.

- many words have more than one possible part-of-speech tag
- example: book
  - book a flight/book a suspect (verb)
  - I read a book (noun)
- example: that
  - Is that a good idea? (determiner)
  - I like that the movie begins (complementizer)

→ requires context to disambiguate

POS Tagging - Approaches

Approaches:

- rule-based tagging
- statistical tagging

POS Tagging - Rule-based tagging

- Example: Constraint grammar
- Requires lexicon with rich information, e.g.
  - part-of-speech
  - morphological information
  - subcategorization pattern
    - with what types of arguments does this word occur
    - what are required arguments
    - e.g. animated subject plus object for a verb
- Procedure:
  1. Lookup word in lexicon to retrieve all candidate POS tags, along with their associated rules (constraints)
  2. Try each rule on input sentence and exclude POS tag if rule does not fit

POS Tagging - Statistical Approaches

- Input: a token sequence (e.g., I fish tuna)
- Task: assign a POS tag to each token (e.g., I/PRP fish/VB tuna/NN)
- Problem of ambiguity: certain tokens are associated with multiple parts of speech (e.g., “fish” could be a noun or a verb)
- Handle ambiguity by exploring different possible sequences
- Approach similar to minimal edit distance search:
    - Generate list of possible sequences
        - Candidate 1: I/PRP fish/NN tuna/NN
        - Candidate 2: I/PRP fish/VB tuna/NN
    - Rank each sequence by its statistical probability

Ranking in Statistical POS Tagging Approaches

- Can calculate a sequence's probability using a supervised machine learning approach
- Humans annotate corpus with POS tags
- Algorithm processes corpus to pre-compute different statistical measures
  1. For each part of speech tag, store likelihood of each word being observed
     - E.g. P(fish — NN) = 0.12, P(fish — VB) = 0.04
  2. For each observed POS tag sequence (n-gram), store probability
     - E.g. P(PRP NN) = 0.1, P(PRP VB) = 0.7
- To compute probability of sequence, multiply 1) with 2) for each word
- Choose sequence with highest probability
- Example of Bayesian modelling

A Second Major NLP Task: Parsing

- Today we introduce two common NLP functionalities that provide linguistic annotations:
  - POS Tagging
  - =¿ Parsing

Parsing - What is it?

- Annotate natural language with linguistic information to represent grammatical structure or relationships
- Often a step towards higher-level tasks, e.g. semantic analyses
- Different types of parsing, e.g.
  - constituency parsing
  - dependency parsing

Constituency Parsing

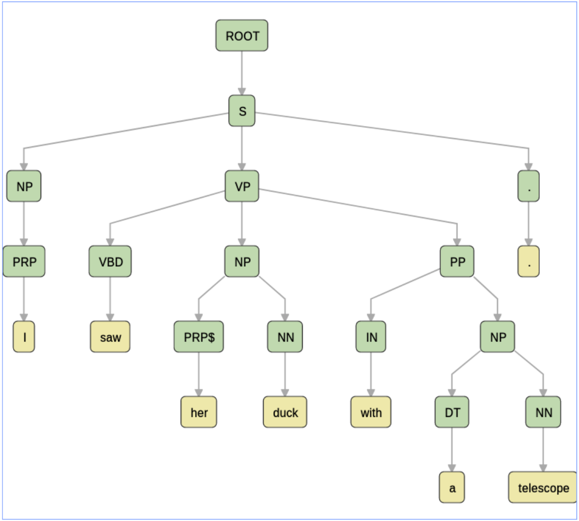
- Constituents: group of words that form syntactic units
  - E.g., Noun phrases (NP): “The/DET big/ADJ dog/NN”
  - E.g., Verb phrases (VP): “walked/VBD quickly/ADVP”
- Assigns syntactic structure
  → Given a sentence, what structure does it have?
- Tasks:
  - identify multi-word syntactic units (eg. noun phrases)
  - capture and represent potentially nested syntactic units (e.g. noun phrases embedded in verb phrases)
- Two common visual representations:
  - Trees
  - Notation with parentheses

Constituency Parsing - Example

Roughly parse the following sentence: "I saw her duck with the telescope."

Constituency Parsing - Example Parse Tree

Input: *I saw her duck with a telescope.*



Constituency Parsing - Example Notation with Parentheses

Input: *I saw her duck with a telescope.*

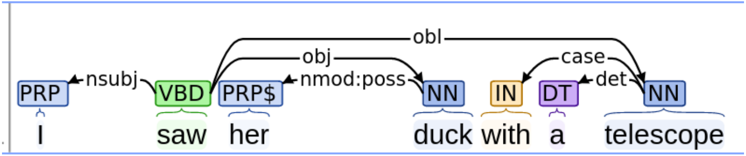
```
(S
(NP (PRP I))
(VP (VBD saw)
(S
(NP (PRP her))
(VP (VB duck)
(PP (IN with)
(NP (DT a)
(NN telescope)
)
)
)
)
)
)
```

Dependency Parsing

- Aim: capture the syntactic (and semantic) relations between words in a sentence
- Consists exclusively of directed grammatical relations between pairs of words
- Each relation has a governor (or head) and a dependent; arcs point from governor to dependent
- Fixed set of dependency relation types (e.g. https://universaldependencies.org/u/dep/index.html)
- Widely used in NLP processing (easier processing due to less nesting, ability to handle languages with different word orders)

Dependency Parsing - Example

Input: *I saw her duck with a telescope.*



Parsing - Ambiguity as Major Challenge

- Attachment ambiguity
  I saw her duck with a telescope
- Coordination ambiguity
  old men and women

Parsing - Methods

Classical parsing paradigms:

- top-down parsing
- bottom-up parsing
    - chart parsing

Before going into more detail, need to look at a tool parsers use called *context-free grammar*

Parsing and Grammars

- Term *grammar* has special meaning in parsing context
- Can think of it as a set of rules that express valid word sequences

```
//Simple Grammar Example

SENTENCE -> NOUN_PHRASE VERB

NOUN_PHRASE -> DET NOUN

DET -> the | a

NOUN -> dog | cat

VERB -> walks | sleeps | eats
```

- Strings in all-capitals denote classes of words, and are called nonterminals (e.g., NOUN).
- Strings in lowercase denote instances of real words in the language,

Uses for Grammars

- A grammar can be used to check if an utterance is grammatical (i.e., satisfies one of the rules)

```
//Simple Grammar Example

SENTENCE -> NOUN_PHRASE VERB

NOUN_PHRASE -> DET NOUN

DET -> the | a

NOUN -> dog | cat

VERB -> walks | sleeps | eats
```

- The sentence “the dog walks” is grammatical
- The sentence “the dog walk” is ungrammatical
- The sentence “the fish sleeps” is ungrammatical

Top-down Parsing

- Uses a grammar to parse a sentence
- Can be seen as a 'rationalist' approach: start with definition of "good" sentences
- Start with the grammar's start symbol
- Expand the start symbol to right-side rules
- Continue expanding until encountering a terminal
- Search string to see if terminal exists in sentence
    - If yes, register rule match, remove terminal from string, and continue

Top-Down Parsing Example

Example: Parse the sentence *the cat sleeps* using the grammar above

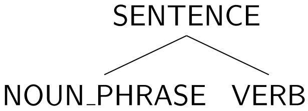
Top-Down Parsing Example

Input: *the cat sleeps*

SENTENCE

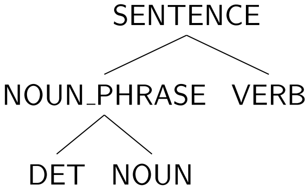
Top-Down Parsing Example cont.

Input: *the cat sleeps*



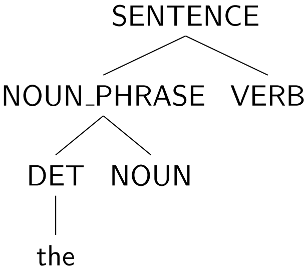
Example: Top-Down Parsing

Input: *the cat sleeps*



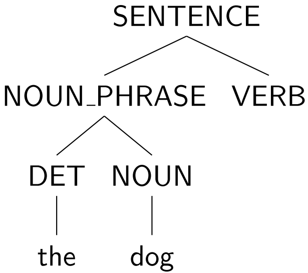
Example: Top-Down Parsing

Input: *the cat sleeps*



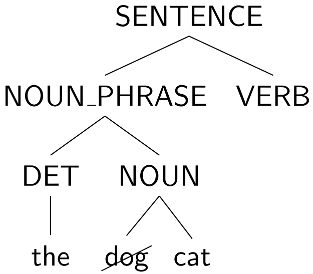
Example: Top-Down Parsing

Input: *the cat sleeps*



Example: Top-Down Parsing

Input: *the cat sleeps*



SENTENCE

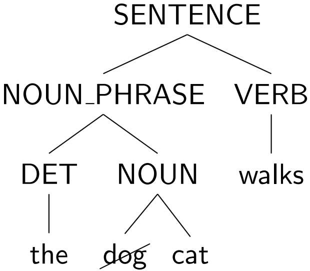
NOUN_PHRASE VERB

DET NOUN

the ~~dog~~ cat

Example: Top-Down Parsing

Input: *the cat sleeps*



SENTENCE

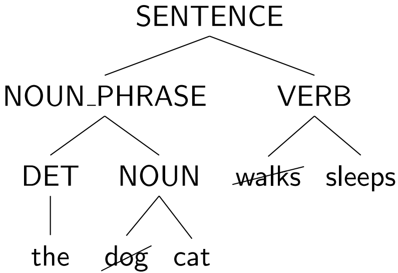
NOUN_PHRASE VERB

DET NOUN walks

the ~~dog~~ cat

Example: Top-Down Parsing

Input: *the cat sleeps*



SENTENCE

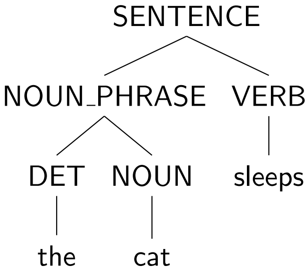
NOUN_PHRASE VERB

DET NOUN ~~walks~~ sleeps

the ~~dog~~ cat

Example: Top-Down Parsing

Input: *the cat sleeps*



In parentheses notation:

```
(S
(NOUN_PHRASE (DET the)
(NOUN cat)
)
(VERB sleeps)
)
```

Top-Down Parsing: Method

- continue until all input words have been integrated into the tree
    - final goal: all words matched
    - if not possible: sentence is ungrammatical
- ambiguity:
    - in case of multiple applicable rules, either create a copy of the tree or use data structures that support parse trees

Further into Parsing

We've only scratched the surface

- Besides top-down, other methods for constituency parsing:
    - bottom-up parsing
        - chart parsing
- Also methods for dependency parsing
- Beyond the scope of our intro
- Topic of ISCL-BA-06 (4th semester)

References and Acknowledgments

These slides are based on slides for Intro to CL, 2022 by Dr. Stephen Bodnar, which were based on slides by Dr. Björn Rudzewitz.

Markus Dickinson, Chris Brew, and Detmar Meurers. *Language and computers*. John Wiley & Sons, 2012.

Levinson, S.C. and Torreira, F. (2015) "Timing in turn-taking and its implications for processing models of language," Frontiers in Psychology, 6. Available at: https://doi.org/10.3389/fpsyg.2015.00731.

Yngve, V. H. (1970). On getting a word in edgewise. CLS-70 (p./pp. 567–577), : University of Chicago.