Natural Language Processing CSCI 4152/6509 — Lecture 2 Ambiguities in NLP; Course Project

Instructors: Vlado Keselj
Time and date: 14:35 – 15:55, 25-Sep-2025
Location: Studley LSC-Psychology P5260

Previous Lecture

- Syllabus and web site review
- Course Introduction

Introduction to NLP

- Definition of NLP
- Some NLP applications
- NLP as a research area
- Short history of NLP
- NLP methodology overview
- Levels of NLP

NLP is Generally Hard

- NLP problems were tackled since 1950s
  - progress has been surprisingly slow and difficult
- Some external evidence of why NLP would be hard:
  - Turing test (imitation game)
  - Evidence from neuro-science:
    "A defining difference between man and non-human primates has been found in the circuitry of brain cells involved in language, according to researchers at the Medical College of Georgia."

Some Computational Reasons that NLP is Hard

1. *highly ambiguous*
   - not easy to program disambiguation
2. *vague* (the principle of minimal effort)
   - not easy to program the context and a priori knowledge
3. *universal* (domain independent)
   - not easy to program general knowledge representation

All of these require reasoning (inference)

Ambiguities at Many Levels of NLP

- Ambiguities of different types happen at all levels of NLP
- We will look at some examples at different levels of NLP

Phonological Ambiguities

- For example, the following words sound the same:
- *two* and *too*, sometimes even *to*
- *would* and *wood*
- *there* and *their*
- *it's* and *its*
- *sea* and *see*
- *I scream* and *ice cream*

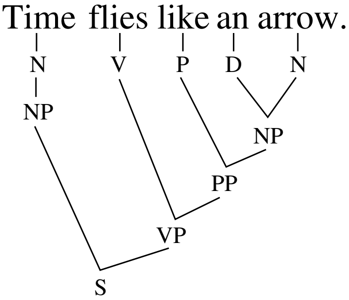
Syntactic Level Ambiguity

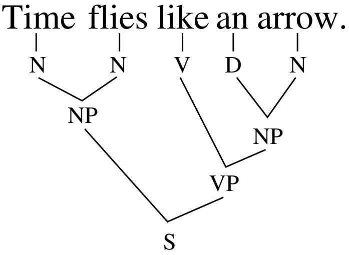
- Example: *Time flies like an arrow.*
  and consider: *Time flies like an arrow... and fruit flies like a banana.*
- Two meanings represented by two parse trees:

Time flies like an arrow.   Time flies like an arrow.

Syntactic Level Ambiguity

- Example: *Time flies like an arrow.*
  and consider: *Time flies like an arrow... and fruit flies like a banana.*
- Two meanings represented by two parse trees:





Similar Examples of Syntactic Ambiguity

- Swat flies like ants.
- I saw a man with a telescope.
- I made her duck.
- I bought a computer with a smart card.
- The cow was found by a stream by a farmer.
- Flying planes can be dangerous.
- They are hunting dogs.
- Eye Drops Off Shelf.
- I'm glad I'm a man, and so is Lola.
- Somewhere in the world a woman gives birth every nine minutes.

Semantic Ambiguities

- **semantic lexical ambiguity,** e.g. “hot” may mean: having high temperature, spicy, intense, good looking, or stolen
- Semantic ambiguity examples at the phrase level:
  1. What does “coast road” mean? Is it a road that leads to a coast, or a road that follows the coast?
  2. “carriage return” — Is it a return of a carriage, or an ASCII character?
  3. “kick the bucket”, and other idioms
- **referential ambiguity** — a kind of semantic ambiguity, or it can be considered discourse ambiguity
  Example: ‘It,’ or ‘he’ in a text – what and who does it refer to?

Pragmatic-level Ambiguity

Examples:

- 12am — is it noon or midnight?
- What date is 10/11/12. Nov 10 or Oct 11 of 2012?

About Course Project

About Course Project

- CSCI 4152:
  - Research or Implementation
  - Individual or Group Presentations
- CSCI 6509 (MCS, PhD, thesis students):
  - Research Project, Individual or Group
  - Individual Presentations
- CSCI 6509 (MACS, MDI non-thesis)
  - Research, Implementation, or Business Oriented
  - Individual or Group Presentations
- Individual projects or teams of up to 4 students
- Preference for a presentation time slot: by email
- Electronic submissions will likely be via GitLab

Course Project

- Deliverables: P0, P1, Presentation, Report
  - P0 — topic proposal,
    - ⋆ due Fri Oct 10, worth 1%, plain text by email
  - P1 — project statement,
    - ⋆ due Fri Nov 7, worth 5%, PDF,
  - P — presentation,
    - ⋆ book a time slot, submit slides, worth: 10%,
  - R — report,
    - ⋆ due Tue Dec 9, worth: 20%, PDF electronic submission.

Emails and Project Web Page

- Use course number in email subject lines, ideally '`CSCI4152/6509`'
- For deliverables, follow the requirements, but the course number is always required in the subject line if delivered by e-mail
- Check the project web page at: `https://web.cs.dal.ca/~vlado/csci6509/project.html`
- The web page contains additional information and will be updated during the term

P0 — Project Topic Proposal

- Worth: 1% of the final mark
- If you choose topic earlier, send it earlier
- If topics overlap too much, later submission may be required to change it
- Plain-text email submission (no attachments) with
  - tentative title
  - list of team members
  - one-paragraph description

P1 — Project Statement

- Worth 5% of the final mark
- Through GitLab (will be clarified later) (text or PDF), about 2 pages
- It must include:
  - Project title,
  - Names of the member(s) of the group,
  - Problem statement,
  - List of possible approaches with citations to relevant work,
  - Project plan for the rest of the term, and
  - List of references.

P — Oral Presentation

- Worth: 10% of the final mark
- Send me preference about time slot by email
- Submit slides at least 24h before presentation
- 8min presentation + 4min for questions (total 12min)
- Use your computer
- Content: related to project, but in a wide sense
- Evaluation:
  - content: interesting, appropriate
  - presentation: vivid, interesting
  - slides: organization, use of text and figures
  - question-answering: to the point

R — Project Report

- Worth: 20% of the final mark
- Submitted electronically
- Typical project report structure:
  - Title, author, course name, date
  - Abstract
  - 1. Introduction, 2. Related work
  - 3. Problem description, Methodology
  - 4. Experiment design, implementation
  - 5. Evaluation
  - 6. Conclusion
  - References, Appendices

How to Choose Project Topic

- Some more information in lecture notes
- A typical approach to a research project
- Alternative project types:
    - theoretical project
    - implementation-oriented
    - software evaluation
    - survey

Resources

- NLP Research Links on the course web page
- `http://acl.ldc.upenn.edu/` — ACL Anthology
- Google scholar and other scientific Internet resources
- Dalhousie library

Example Themes

- These are some themes related to current research at Dal CS
- However, you are encouraged to think about other, different areas
- Themes:
  - Analysis of social media data (e.g., Twitter)
  - Author attribution and profiling
  - Sentiment analysis
  - Processing of email data
  - Language, dialect detection; demographic analysis using NLP, etc.

Topics of Some Previous Course Projects

- The Effects of Sentence Simplification as a Preprocessing Step in Text Summarization
- An Analysis of Predictive Text Software and Algorithms
- Extraction of Topics and Clustering of Documents using Topic Modeling Algorithm
- Role of Emoticons for Sentiment Analysis
- Author Profiling for Keyboard Layouts to Understanding User Typing Pattern
- Natural Language Math Problem Assistance Tool
- Canadian Happiness Level Mapping by Using Twitter Data
- Detection of Emotion and Emotion Stimuli in Text
- *and many more are included in the notes.*

Part II: Stream-based Text Processing

- Considering text as a stream of characters, words, and lines of text
- Review of Finite Automata and Regular Expressions
- Review of Unix-style text processing
- Introduction to Perl
- Morphology fundamentals
- N-grams
- Reading: Chapter 2, Jurafsky and Martin

Finite-State Automata

- Regular Expressions and Regular Languages
- Regular Languages can be described using
  - Regular Expressions
  - Regular Grammars
  - Finite-State Automata (DFA and NFA)
- DFA = Deterministic Finite Automaton
- NFA = Non-deterministic Finite Automaton
- also referred to as Finite-State Machines

Typical Low-level NLP Tasks

- Pre-processing text
- Tokenization
- Sentence Segmentation
- Morphological Processing (e.g., Stemming)
- “Vectorizing” Text
- Information Extraction (simpler cases)
- *and so on*

Example Task: Removing HTML Tags

Deterministic Finite Automaton

- Formally defined as a 5-tuple: $(Q, \Sigma, \delta, q_0, F)$
  - Q is a set of states
  - Σ is an input alphabet
  - $\delta : Q \times \Sigma \to Q$ is a transition function
  - $q_0 \in Q$ is the start state
  - $F \subset Q$ is a set of final or accepting states
- Graph representation is frequently used
- Consider finite automata for sets of strings:

  `baaa...a!`   `ha-ha-...-ha`

  `up-up-down-up-down-up-up-...down`

DFA for language baa...a! using a graph