729G86/TDP030 Language Technology (VT2025)

Course Introduction

Marcel Bollmann

Department of Computer and Information Science (IDA)

LINKÖPING UNIVERSITY

Adapted from slides by Marco Kuhlmann.

Meet the teaching assistants!

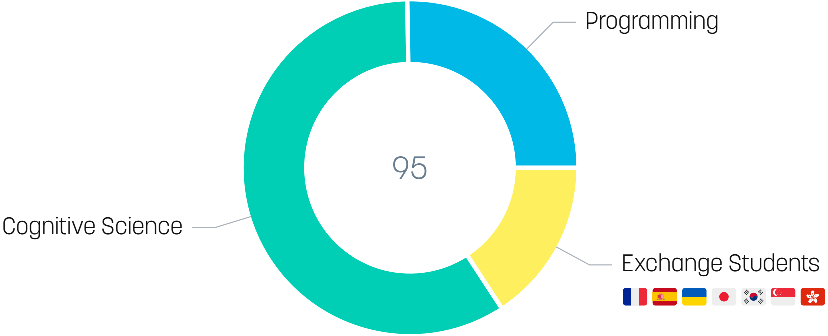
Markus Fritzsche

Kevin Glocker

Kätriin Kukk

Romina Oji

Meet your fellow students!



This introduction session

1. Language Technology

2. Logistics

3. Text processing

What is language technology?

1 ○ ○

Language technology
is technology for
the analysis and interpretation
of natural language.*

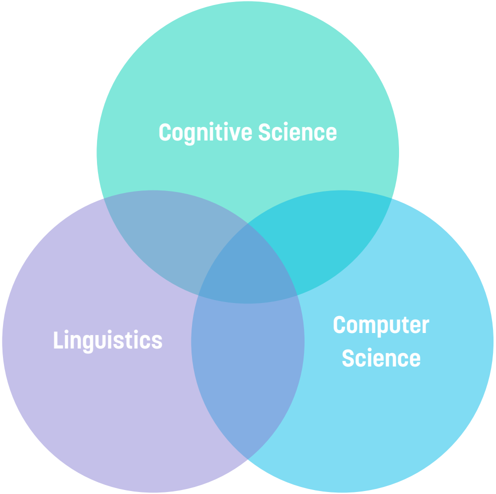
*Not formal or programming languages.

Language technology

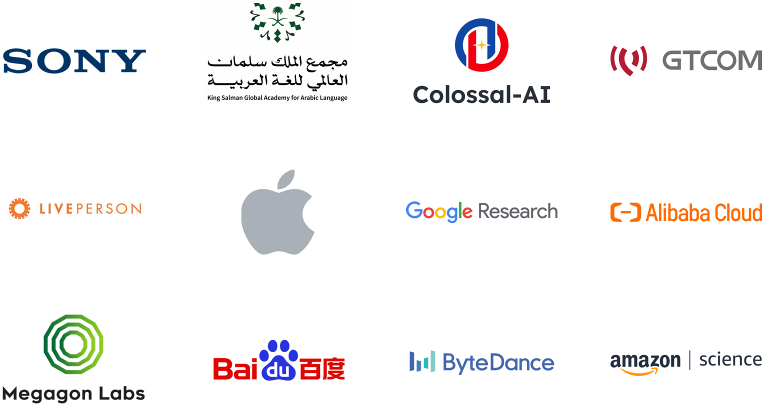
An **interdisciplinary** research area!

Closely related:

- Natural language processing (NLP)
- Computational linguistics (CL)
- Speech processing



Commercial interest



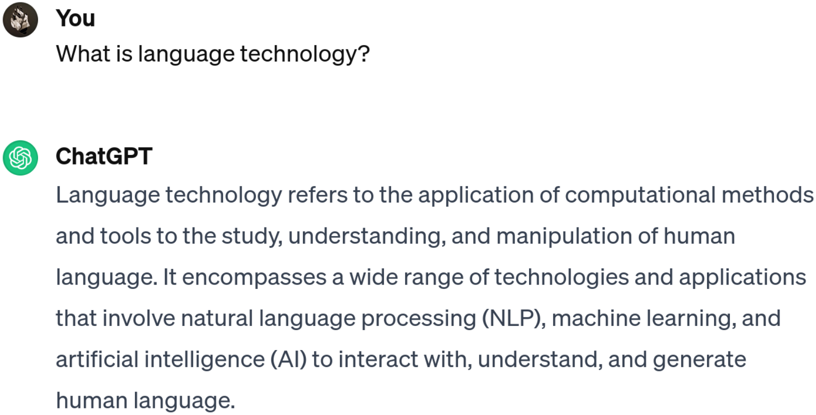
Diamond & Platinum sponsors from EMNLP 2023

Why language technology?

> *We are drowning in information but starved for knowledge.*
>
> — John Naisbitt (1982)

- We **communicate information** primarily through language.
- Language is generally **produced by & meant for humans**, rather than computers.
  - So-called unstructured data
- Language technology can help extract **structured data** from language.

Example: ChatGPT



You

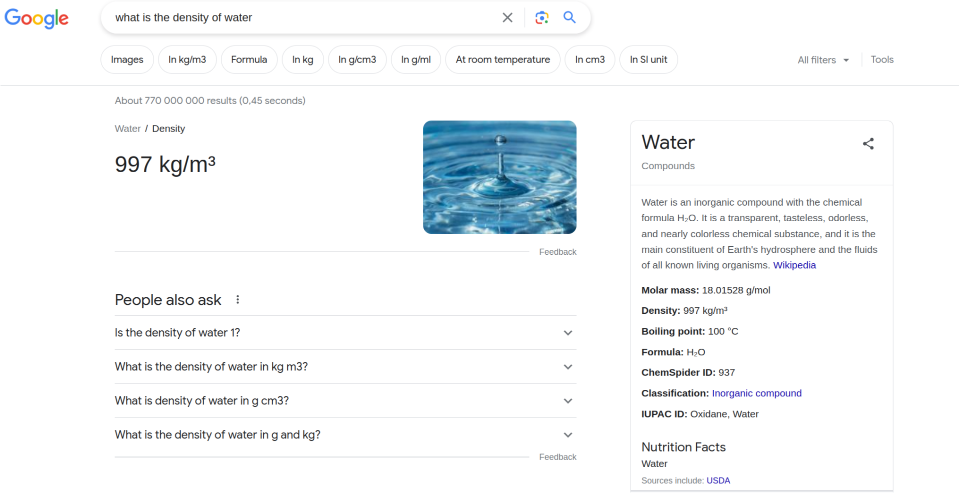
What is language technology?

ChatGPT

Language technology refers to the application of computational methods and tools to the study, understanding, and manipulation of human language. It encompasses a wide range of technologies and applications that involve natural language processing (NLP), machine learning, and artificial intelligence (AI) to interact with, understand, and generate human language.

Via ChatGPT

Example: Search engines



what is the density of water

About 770 000 000 results (0,45 seconds)

Water / Density

997 kg/m³

People also ask

Is the density of water 1?

What is the density of water in kg m3?

What is density of water in g cm3?

What is the density of water in g and kg?

Water

Compounds

Water is an inorganic compound with the chemical formula H_2O. It is a transparent, tasteless, odorless, and nearly colorless chemical substance, and it is the main constituent of Earth's hydrosphere and the fluids of all known living organisms. Wikipedia

Molar mass: 18.01528 g/mol

Density: 997 kg/m³

Boiling point: 100 °C

Formula: H_2O

ChemSpider ID: 937

Classification: Inorganic compound

IUPAC ID: Oxidane, Water

Nutrition Facts

Water

Sources include: USDA

Via Google

Example: Forensic linguistics

> *I realized the faxed copy I just received was an outline of the manifesto, using much of the same wording, definitely the same topics and themes. … I invented [the language analysis] for this case and really, forensic linguistics took off after that.*

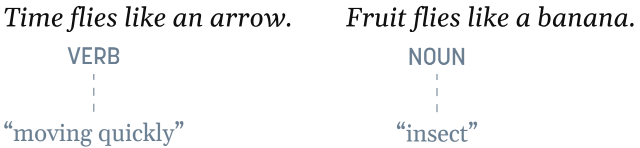
— James Fitzgerald, profiler

Sources: Wikipedia & Newsweek

A major challenge: Ambiguity

Definition

The term **ambiguity** refers to the fact that a linguistic expression can often mean several different things.



Time flies like an arrow.

VERB

“moving quickly”

Fruit flies like a banana.

NOUN

“insect”

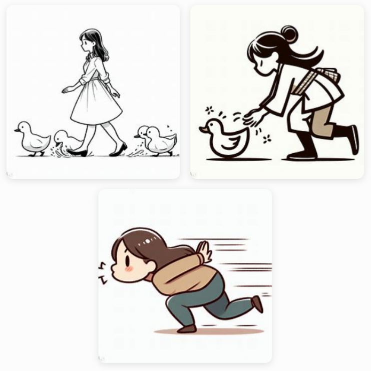
Lexical ambiguity

I saw her ***duck*** .

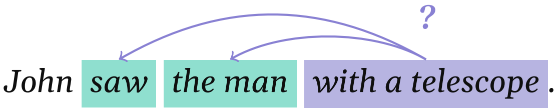
- Ambiguity poses a major challenge for computers.

- The images to the right were generated by **DALL·E 3** from the prompt:

  "A simple illustration of a woman ducking with no other objects in the scene."



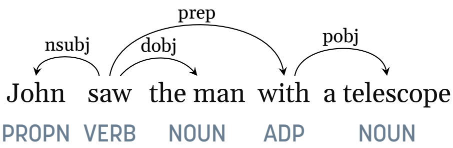
Structural ambiguity



?

John saw the man with a telescope .

- **Linguistic representations** can describe the underlying structures:

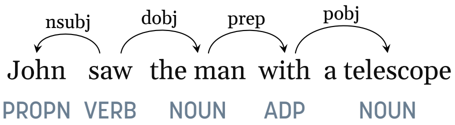


prep

nsubj dobj pobj

John saw the man with a telescope

PROPN VERB NOUN ADP NOUN

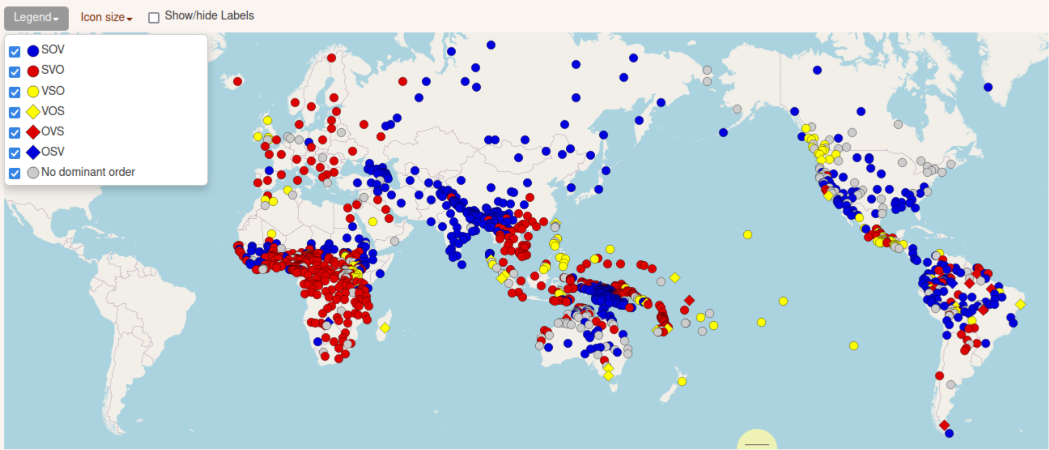


nsubj dobj prep pobj

John saw the man with a telescope

PROPN VERB NOUN ADP NOUN

Each language is different



Source: The World Atlas of Language Structures

Recurring questions

- How does this method work?
  - algorithm, mathematical formula, …
- How can we evaluate this method?
  - accuracy, precision/recall, …
- How does this method use data?
  - estimate probabilities, learn weights of a neural network, …

Course logistics

○ 2 ○

Week	Topic	Scheduled
4	Introduction	2 Lectures + Lab zero
5	Text Classification	Lecture + 2 Lab sessions
6	Language Modelling	Lecture + 2 Lab sessions
7	Sequence Labelling	Lecture + 2 Lab sessions
8	Word Embeddings	Lecture + 2 Lab sessions
9	Syntactic Analysis	Lecture + 2 Lab sessions
10	Project Work	—
11	Project Work	—
12	Project Work	Project presentations
12	—	Written exam

Course website

https://liu-nlp.ai/lang-tech/

Examination

Practical assignments

- 2 credits
- 6 labs
- Pairs of two

Project assignments

- 2 credits
- 6 deliverables
- Groups of ≈ 6

Digital written exam

- 2 credits
- Written exam
- Individually

Grading scales

729G86	Fx	E	D	C	B	A
TDP030	U	3		4		5

💡 In the CogSci programme, this roughly corresponds to:

U	G			VG	

Sign up for a lab group!

729G86

TDP030

- Please sign up by **Thursday, 16:00**! I will transfer the groups to Lisam after that.
  - You won’t be able to see the lab submissions in Lisam before that.

Working on the labs

- Labs come in form of **Jupyter Notebooks**.
  - All required libraries are installed on the lab computers.
- Each lab *(except L0)* has a **basic** and an **advanced ('X')** part.
  - To **pass the labs**, you must pass all basic labs.
  - You can **earn a higher grade** by passing the advanced labs.

Assignment due dates

Tuesday, 23:59,
the week after the labs

2025-03-28
(last exam date)

- First due date: **timely, formative feedback**

⚠ Important

We will ***not*** grade (re-)submissions between the deadlines!

Project assignments

- You will work on a **project** in groups of ≈ 6 students.
  - Groups will be **mixed** to have **both** 729G86 and TDP030 students.
  - You may form smaller groups *within your course code* if you want, but you will be paired up with students from the other course code.
    - 729G86: ca. 3–4 students
    - TDP030: ca. 2–3 students

Info

We will discuss the project module in detail on Friday!

Optional: Form a project group!

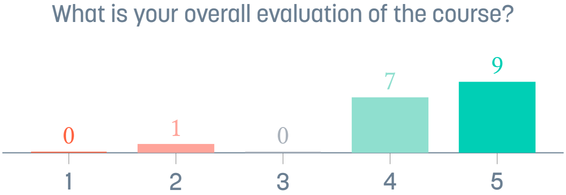
- You can give your preferences by **Thursday, 16:00**!
- **I will assign the project groups** after that.

Digital written exam

- The course ends with a **digital written exam**.
  - You must register for the exam at least 10 days before.
- The format of the exam is changing this year.
  - There will be **examples of possible exam questions** throughout the course.
  - I will share a **sample exam** a few weeks before the first examination.

Previous course evaluation

- 73 students took the course in VT2024.
- 17 students submitted a course evaluation (→ 23% 😟).



- A detailed **description of changes** is on the course website!

Questions?

In person

- During the session
- In the break
- In the lab

Asynchronously

- Email

Project-related

- Email
- Schedule a meeting via the booking link on the website

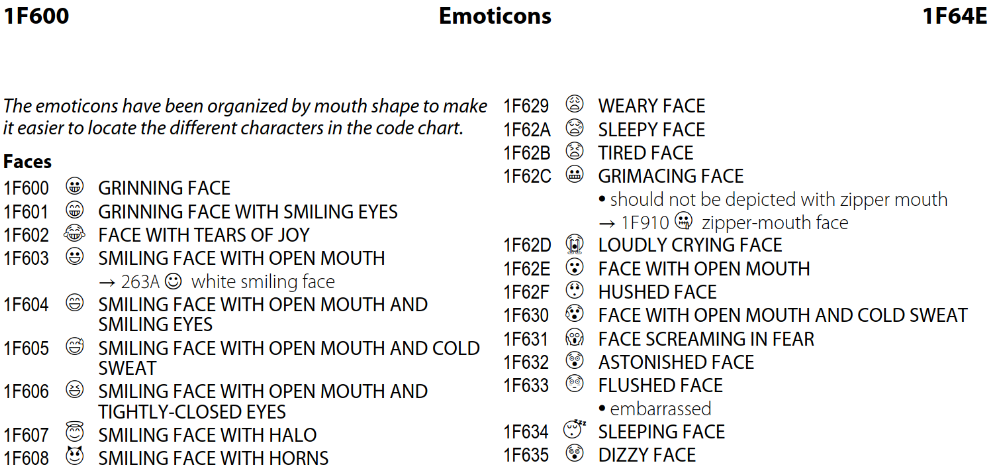
✉ `marcel.bollmann@liu.se` — marbo59

Text Processing

○ ○ 3

How text is stored on a computer

- Text is stored as a **sequence of bytes**.
  - 1 byte = 8 bits = 256 possible values
- An **encoding scheme** defines how bytes map to characters.
  - Usually UTF-8, but sometimes still ISO-8859, or others
- **Unicode** is an initiative to define code points for all naturally occurring characters.
  - Natural languages, mathematical symbols, emoji, …



1F600 **Emoticons** 1F64E

The emoticons have been organized by mouth shape to make it easier to locate the different characters in the code chart.

Faces

1F600	😀	GRINNING FACE
1F601	😁	GRINNING FACE WITH SMILING EYES
1F602	😂	FACE WITH TEARS OF JOY
1F603	😃	SMILING FACE WITH OPEN MOUTH → 263A ☺ white smiling face
1F604	😄	SMILING FACE WITH OPEN MOUTH AND SMILING EYES
1F605	😅	SMILING FACE WITH OPEN MOUTH AND COLD SWEAT
1F606	😆	SMILING FACE WITH OPEN MOUTH AND TIGHTLY-CLOSED EYES
1F607	😇	SMILING FACE WITH HALO
1F608	😈	SMILING FACE WITH HORNS
1F629	😩	WEARY FACE
1F62A	😪	SLEEPY FACE
1F62B	😫	TIRED FACE
1F62C	😬	GRIMACING FACE • should not be depicted with zipper mouth → 1F910 🤐 zipper-mouth face
1F62D	😭	LOUDLY CRYING FACE
1F62E	😮	FACE WITH OPEN MOUTH
1F62F	😯	HUSHED FACE
1F630	😰	FACE WITH OPEN MOUTH AND COLD SWEAT
1F631	😱	FACE SCREAMING IN FEAR
1F632	😲	ASTONISHED FACE
1F633	😳	FLUSHED FACE • embarrassed
1F634	😴	SLEEPING FACE
1F635	😵	DIZZY FACE

via Unicode 15.1 Character Code Charts

UTF-8

- Unicode can represent 2^{32} = 4,294,967,296 different characters.
  - But a single byte can only represent 256 values.

- UTF-8 is the most widely used scheme to encode Unicode in bytes.
  - 8-bit Unicode Transformation Format
  - Unicode characters 0–127 = 1 byte, 128–2,047 = 2 bytes, ...

VarfÃ¶r blir det sÃ¥ hÃ¤r?

	s	å		[spc]	h	ä		r
Unicode	115	229		32	104	228		114

↑ UTF-8 ↑

Bytes	115	195	182	32	104	195	165	114

↓ ISO-8859 ↓

	s	Ã	¥	[spc]	h	Ã	¤	r

Example by Per Starbäck

Tokenization

- **Segmenting a text** into meaningful units is a common preprocessing task.
  - e.g. sentences, words

- **Tokenization** is the task of segmenting a text into words or “word-like” units.

🤔 Problem

What are words? What are the “best” units?

A simple tokenizer based on whitespace

- In Python, we can easily **split on whitespace** to tokenize:

```python
def tokenize(lines):
    for line in lines:
        for token in line.split():
            yield token
```

- Now we can **print all tokens** in a text file:

```python
with open("example.txt", "r") as f:
    for token in tokenize(f):
        print(token)
```

Tokenization is harder than one might think

Whitespace tokenization	*“The␣food␣wasn’t␣great,”␣said␣Mr.␣James.*
A more useful tokenization	*“␣The␣food␣was␣n’t␣great␣,␣”␣said␣Mr.␣James␣.*

- **Undersegmentation**: tokens aren’t split up when they should be
  - punctuation marks
  - *wasn’t* → *was␣n’t*
- **Oversegmentation**: tokens are split up when they shouldn’t be
  - The period in *Mr.* is part of the abbreviation

A simple tokenizer based on regular expressions

- We can implement more sophisticated tokenization rules with **regular expressions**:

```python
def tokenize(regex, lines):
    for line in lines:
        for match in re.finditer(regex, line):
            yield match.group(0)
```

Types versus tokens

Rose	*is*	*a*	*rose*	*is*	*a*	*rose*	*is*	*a*	*rose*	*.*
1	2	3	4	5	6	7	8	9	10	11

— Gertrude Stein, 1913

- We distinguish between **tokens** and **types**.
  - types ≈ “unique tokens”
- The example above has 11 tokens, but only 5 types.
  - types: *a, is, rose, Rose, .*

Normalization

- **Lowercasing** all tokens
  - *rose* vs. *Rose*; **but:** *apple* vs. *Apple*
- Harmonization of **spelling variants**
  - *color* ↔ *colour*; *recognise* ↔ *recognize*; *through* ↔ *thru*
- **Stemming** (suffix removal)
  - *wanted, wanting, wants* → *want*

Stop words

- A **stop word** is a frequent word that does not contribute much value to the application in question.
  - Example: function words like *a*, *the*, *and* when performing search
- Stop words are **application-specific**.
  - There is no single universal list of stop words!
  - Not all applications use stop word lists.

Example of stop words in English

a about above across after afterwards again against all almost alone along already also although always am among amongst amount an and another any anyhow anyone anything anyway anywhere are around as at back be became because become becomes becoming been before beforehand behind being below beside besides between beyond both bottom but by ca call can cannot could did do does doing done down due during each eight either eleven else elsewhere empty enough even ever every everyone everything everywhere except few fifteen fifty first five for former formerly forty four from front full further get give go had has have he hence her here hereafter hereby herein hereupon hers herself him himself his how however hundred i if in indeed into is it its itself just keep last latter latterly least less made make many may me meanwhile might mine more moreover most mostly move much must my myself n't …

Taken from spaCy

Other segmentation problems

- Sometimes we also want to perform **sentence segmentation**.

- This is not as simple as splitting on periods!
  - *We first visited the U.S. Later, we travelled from the U.S. to Canada.*

    ↑ sentence-final

    ↑ **not** sentence-final

- In some languages, even **word segmentation** can be much more difficult.
  - No whitespace between words in e.g. Chinese, Thai

LINKÖPING
UNIVERSITY