IN2110: Språkteknologiske metoder

Introduksjon

Eivind A. Bergem, Fredrik Jørgensen, Stephan Oepen, Erik Velldal

Språkteknologigruppen (LTG)

15. Januar, 2019

Topics for today

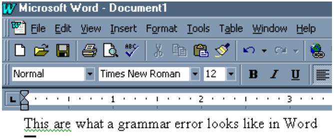
- AI, NLP, ML — What are they?
    - Definitions
    - Applications
    - Historical review
- Outline of lectures and learning goals
- Practical details
    - Syllabus
    - Obligatory assignments
    - Programming
    - Communication

What is Natural Language Processing?

- Making computers ‘understand’ human language
- Aka language technology or computational linguistics
- Young and interdisciplinary field:
- Computer science + linguistics
- (+ cognitive science, statistics, machine learning . . .)
- Sub-field of AI.

NLP applications

- Grammar and/or spell checkers, auto-completion
- Machine translation
- Q&A systems, dialog systems, and chatbots
- Speech recognition and synthesis
- Intelligent information extraction
- Summarization
- Sentiment analysis
- Any application requiring an understanding of language. . .


Microsoft Word - Document1
File Edit View Insert Format Tools Table Window Help
Normal
Times New Roman
12
B I U

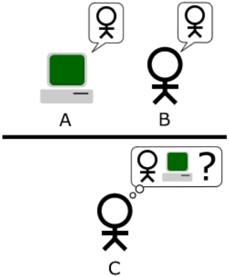
This are what a grammar error looks like in Word

Cortana. Siri amazon echo "Ok Google"

What is AI?

- The term ‘AI’ coined by John McCarthy (Dartmouth Workshop, 1956).
  - *The science and engineering of making intelligent machines.*
  - *Every aspect of learning or any other feature of intelligence can be so precisely described that a machine can be made to simulate it.*

- Alan Turing, 1950:
  - *I propose to consider the question, ’Can machines think?’*
- The Turing Test, based on the imitation game.
- Language understanding has always been central to AI.



AI in this course

- For our purposes: AI is a toolkit of methods for representation and problem solving, a bag of tricks.

Paradigm shifts in NLP (and AI at large)

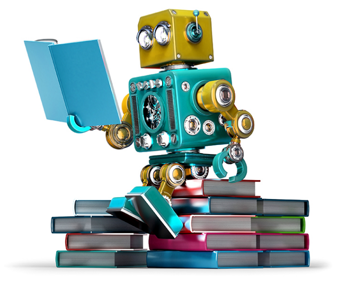
- 50s–80s: mostly rule-based (symbolic / rationalist) approaches.
- Hand-crafted formal rules and manually encoded knowledge.
- (Though some AI research on neural networks in the 40s and 50s).
- Late 80s: success with statistical ('empirical') methods in the fields of speech recognition and machine translation.
- Late 90s: NLP (and AI at large) sees a massive shift towards statistical methods and machine-learning.
- Based on automatically inferring statistical patterns from data.
- 00s: Machine-learning methods dominant.
- 2010–: neural methods and deep learning.
- Today, in the popular media, AI is mostly synonymous with ML.

The basis of empirical methods

Machine Learning

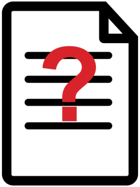
- *the study of computer algorithms that improve automatically through experience* (Tom Mitchell 1997).

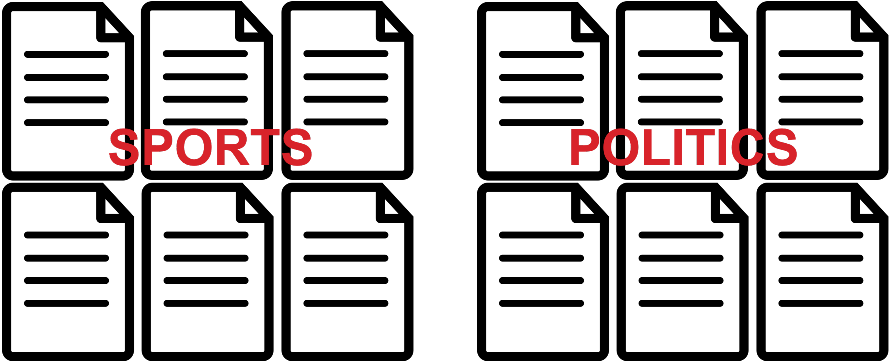
- Similar to statistical data analysis, but the models are applied to solve a practical tasks rather than to describe the data.
- Goal: to learn from data.
- Not interested in simply learning by rote; want to generalize.
- Used in many data-intensive fields besides NLP, e.g. bio-informatics, physics, robotics, image processing, market analytics, law, etc.
- A core element in the emerging field of *data science*.



Two main types of machine learning (1)

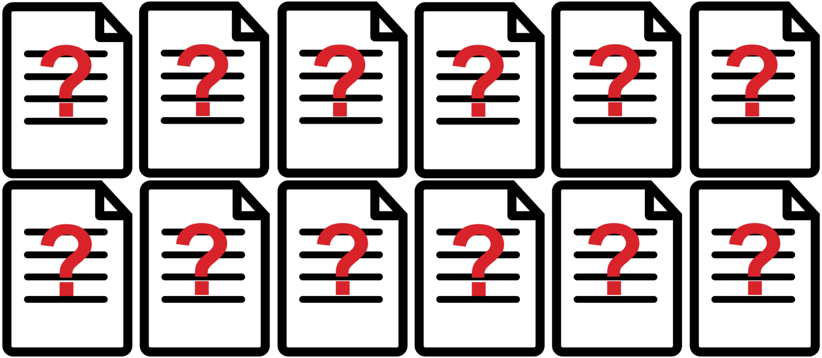
- Supervised learning ('Veiledet læring')
- Requires training data; pre-defined examples of what we want the algorithm to learn.
- Labeled data.





Two main types of machine learning (2)

- Unsupervised learning ('Ikke-veiledet læring')
- Unlabeled data: no pre-defined examples.

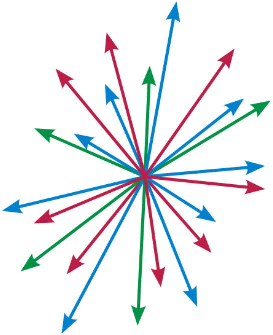


Supervised vs unsupervised learning

- Currently we get most precise results with supervised learning.
- Typically requires manually labeled training data (= costly).
- A lot of research directed at making better use of unsupervised methods; we have much more unlabeled data available.
- A lot of fuzz about Big Data: great for training unsupervised methods or when applying a pre-trained supervised model.
- But for supervised methods, the need for labeled data typically limits the size.
- ML is no free lunch:
- The data is often more important than the algorithm.
- And related to this; how we choose to represent the data.

Overview of lectures

- `https://www.uio.no/studier/emner/matnat/ifi/IN2110/v19/`
- Vector space models (non-probabilistic ML)
- Representing documents
- Representing word meaning
- Classification (supervised learning)
- Sequence classification
- Statistical parsing



Reading list

- Selected chapters of the following books.
- Both are freely available online.
- Jurafsky & Martin (2008):
  Speech and Language Processing (3rd ed. draft of 2018):
  `https://web.stanford.edu/~jurafsky/slp3/`
- Manning, Raghavan, & Schütze (2008):
  Introduction to Information Retrieval:
  `https://nlp.stanford.edu/IR-book/information-retrieval-book.html`

Something about ourselves

`https://skjema.uio.no/110010`

Screencasting

- Hope to screencast all lecture sessions (audio and slides).
- Will link to IN2110 YouTube channel from course page soon.

Obligatory exercises

- Two obligatory exercises, each in two parts; four submissions:
- 1a+b and 2a+b.
- Possible to earn maximum of 10 points for each submission.
- In order to pass and qualify for the exam you need to collect at least 60% of the points across all exercises, i.e. 12 points across a+b.
- Extensions can only be given in case of illness, and re-submissions will not be possible.
- See course page for the schedule:

  `https://www.uio.no/studier/emner/matnat/ifi/IN2110/v19/innleveringer.html`

Course Communication

- Questions?
  - Piazza: on-line discussion board linked from course page.
  - `in2110-hjelp@ifi.uio.no` reaches all course staff:
    - Eivind Alexander Bergem (`eivinabe`);
    - Fredrik Jørgensen (`fredrijo`);
    - Stephan Oepen (`oe`);
    - Erik Velldal (`erikve`);
    - Henrik Askjer (`henraskj`).
- Messages:
  - Check your UiO email regularly;
  - Check the course pages regularly;
  - Participate in the on-line discussion board.

Language technology in Python

- Python is a simplified Lisp dialect (with an idiosyncratic syntax) with great popularity for all things ‘data science’;
- it provides a very convenient, high-level scripting language with a gentle learning curve; works easily across different platforms;
- comprehensive standard library; ecosystem of community-maintained add-on modules with specialized (and optimized) functionality;
- pretty much everything open-source; we provide reference environment on IFI Linux machines; in principle possible to install ‘at home’.

A menagerie of interoperable modules

- The Python add-ons ecosystem is vast (and can be confusing to navigate);
- NumPy for efficient multi-dimensional arrays and linear algebra;
- SciKit-Learn for machine learning (and data preparation);
- MatPlotLib for visualization and data analysis;
- JuPyter as an integrated development environment and authoring tool;
- NLTK and spaCy for text pre-processing (from tokenization to parsing).