Aalto University

Statistical Natural Language Processing course 2024
Last lecture: Conclusion

Mikko Kurimo

Learning goals

- To learn **how statistical and adaptive methods** are used in information retrieval, machine translation, text mining, speech processing and related areas **to process natural language data**
- To learn how to apply the basic methods and techniques for clustering, classification, generation and recognition **by natural language modeling**

How to achieve the learning goals (and pass the course)?

- *Participate actively in each lecture, read the corresponding material and ask questions* to learn the basics, take part in discussions, complete the lecture exercises **DONE**
- *Solve each home assignment by yourself after each lecture* to learn how to solve the problems, in practice **DONE**
- *Participate actively in project work* to learn to apply your knowledge **Final report DL April 19**
- *Prepare well for the examination* **Course exam April 16**

Course grading and the exam

- **60% (or 40% + exam)** of the grade is from the weekly **home assignments and lecture activities.**
- **20%** of the grade comes from the **optional exam** organized in 16 April. Exam points are counted on top of the assignment points (see below) which are then cut to 2/3. Examples:
  - 40/60 assignments + 10/20 exam = 50/60 points
  - 50/60 assignments + 15/20 exam = 55/60 points
  - 50/60 assignments + 5/20 exam = (~~45/60~~) 50/60 as without exam
  - Note: The exercise points are scaled to 60 and exam points to 20 for computing the final grade
- **40%** of the grade is from the **project work:** experiments, literature study, short (video) presentation and final report.

About scores, points and grades in 2023

- Max score in home assignments was 161 => 50p
- Max score in lecture activity was 25 => 10p
- Exam points could substitute max 20p of missed points
- In 2023 the points gave non-rounded grades like this:
    - 60p gave 5.6
    - 53p gave 4.6
    - 46p gave 3.6
    - 38p gave 2.5
    - 31p gave 1.5
    - 24p gave 0.6
    - 20p or less gave 0
- **The final grade** is the average of this (60%) and the project (40%) grade

Course project grading

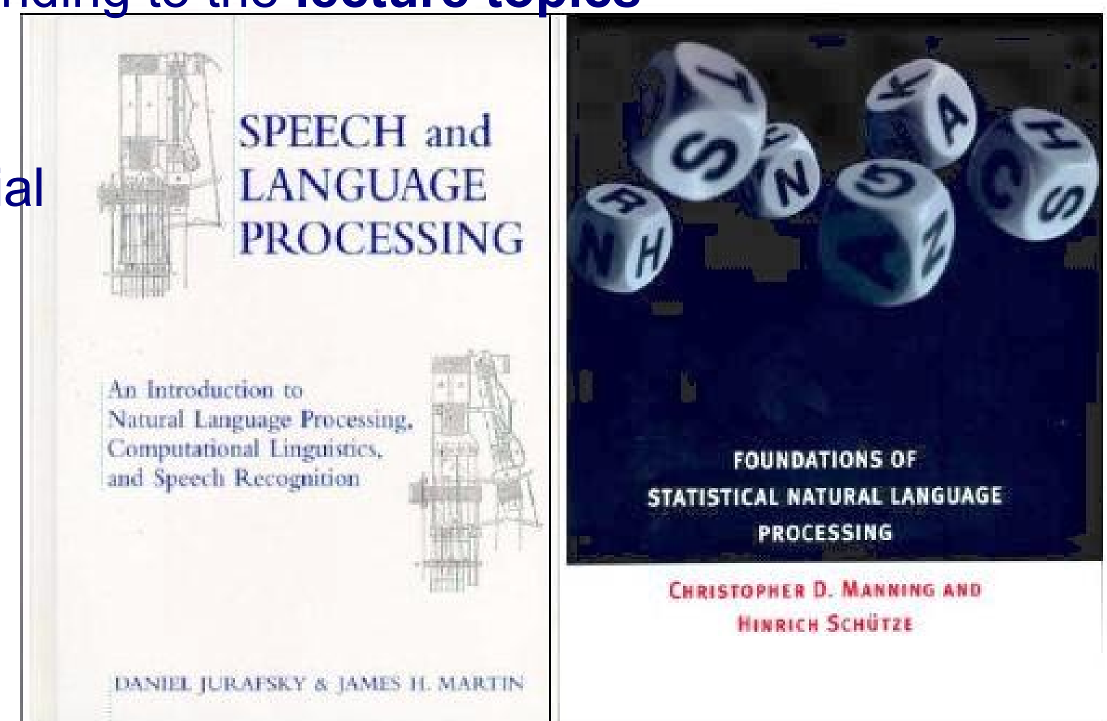
- See Mycourses for the requirements of an acceptable project report
- Excellent projects typically include additional work such as
  - Exceptional analysis of the data
  - Application of the method to a task or several
  - Algorithm development
  - Own data set(s) (with preprocessing etc to make them usable)
  - Unique and clever solutions for the Kaggle competition,

Exam grading

- The exam will be a remote digital exam
- There are 5 questions in MyCourses open in Tue 16.4. 12:00 – 15:00
- Max 6 points per question makes total 30 *exam points*. The points are scaled to 20 *assignment points* and added to the max 40 from assignments and activities
- You can use any books, course material, internet, calculators and toolkits
- **You must write in your own words**. If any copy-pasted or LLM generated texts are found, it will be an automatic reject and report to the study admins
- **You are not allowed to communicate or collaborate** with other people during the exam. If any co-operation is found, it will be an automatic reject and report to the study admins
- All exam questions and course exercise materials are copyrighted. **You are not allowed to distribute** them in any way.

Reading material

- Manning & Schütze, Foundations of Statistical Natural language processing (1999) *http://nlp.stanford.edu/fsnlp/*
- Jurafsky & Martin, Speech and Language Processing (3rd ed. Draft, 2024) *http://web.stanford.edu/~jurafsky/slp3/*
- Read the chapters corresponding to the **lecture topics**
- Do not forget to study the topic-specific reading material (mentioned in slides)



A
Mi

Lecture schedule 2024

09 Jan 1 Introduction & course organization / Mikko Kurimo

16 Jan 2 Statistical language models / Mikko Kurimo

23 Jan 3 Sentence level processing / Mikko Kurimo

30 Jan 4 Word2vec / Tiina Lindh-Knuutila

06 Feb 5 Neural language modeling and LLMs / Mittul Singh

13 Feb 6 Morpheme-level processing / Mathias Creutz

20 Feb (exam week, no lecture)

27 Feb 7 Speech recognition / Tamas Grosz

05 Mar 8 Chatbots and dialogue agents / Mikko Kurimo

12 Mar 9 Statistical machine translation / Jaakko Väyrynen

19 Mar 10 Neural machine translation / Sami Virpioja

26 Mar 11 LLMs in industry / Shantipriya Parida

02 April (spring break - no lecture)

09 Apr 12 Course conclusion / Mikko Kurimo

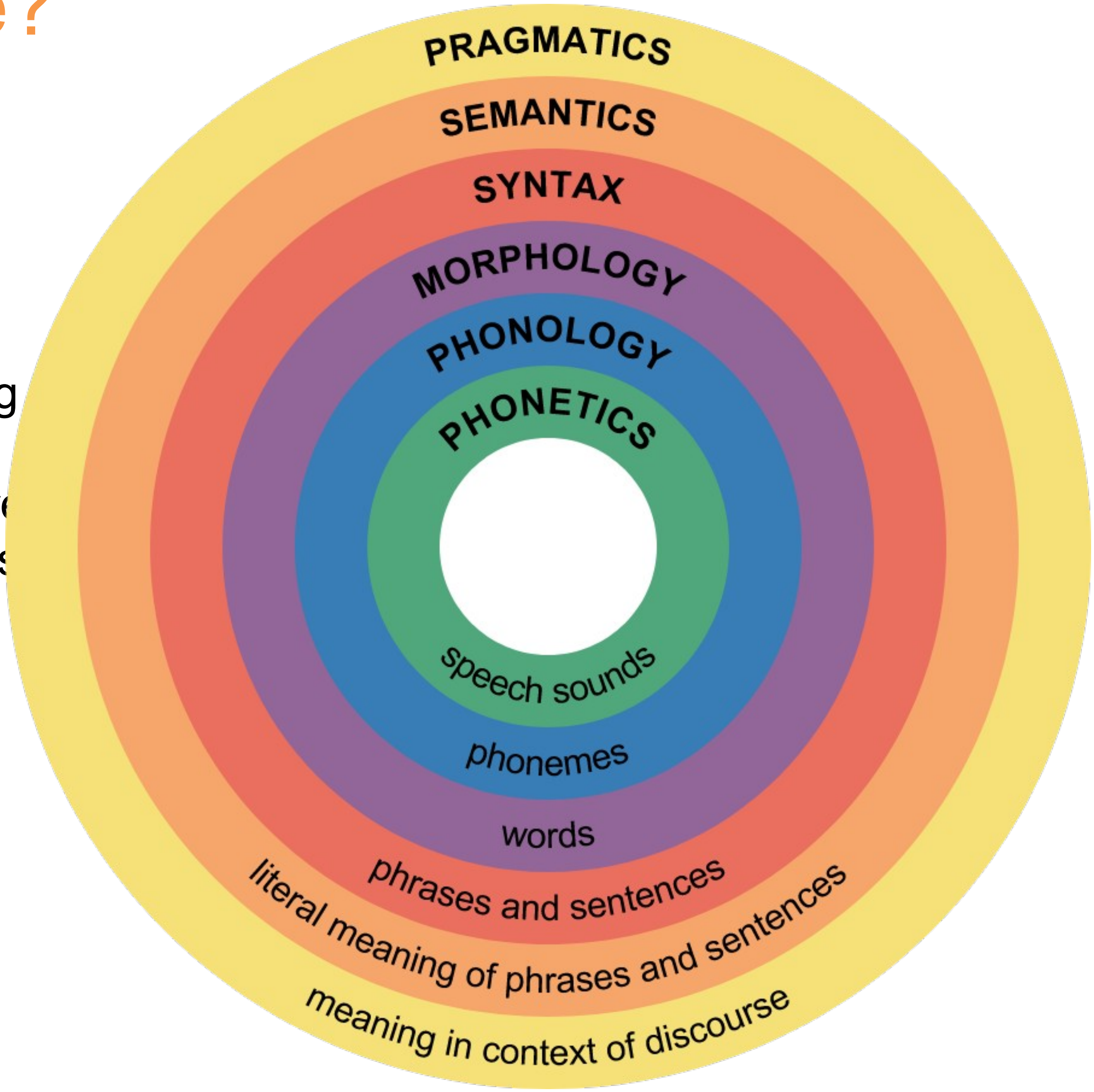
16 April Exam

1. Introduction

- What does language include?
- What makes languages so complex?
- What are the applications of statistical language modeling?

What is in a language?

- Phonetics and phonology:
  - the physical sounds
  - the patterns of sounds
- Morphology: The different building blocks of words
- Syntax: The grammatical structure
- Semantics: The meaning of words
- Pragmatics, discourse, spoken interaction...



Complexity of languages

- A large proportion of modern human activity in its different forms is based on the use of language
- Large variation:
  - morphology and
  - syntactic structures
- Complexity of natural language(s)
  - More than 6000 languages, many more dialects
  - Each language a large number of different word forms
  - Each word is understood differently by each speaker of a language at least to some degree

Application areas

- Information retrieval
- Text clustering and classification
- Automatic speech recognition
- Natural language interfaces
- Statistical machine translation
- Topic detection
- Sentiment analysis
- Word sense disambiguation
- Syntactic parsing
- Text generation
- Image, audio and video description
- Text-to-speech synthesis
- ...

2. Statistical language modeling

- N-gram LMs
  - Data sparsity problem
  - Equivalence classes
  - Back-off and interpolation
  - Smoothing methods
- Maximum entropy LMs
- Continuous space LMs
- Neural network LMs

Estimation of N-gram model

$$P(w_i \mid w_j) = \frac{c(w_j, w_i)}{c(w_j)} \qquad \frac{c(\text{“eggplant stew”})}{c(\text{“eggplant”})}$$

$$= P(w_i) b_{w_j} \qquad \text{otherwise}$$

- Bigram example:
  - Start from a **maximum likelihood estimate**
  - probability of *P(“stew” | “eggplant”)* is computed from **counts** of *“eggplant stew”* and *“eggplant”*
  - works well for frequent bigrams
    - Why not for good rare bigrams?

Picture by B.Pellom

Zero probability problem

- If an N-gram is not seen in the corpus, it will get probability = 0
- The higher N, the sparser data, and the more zero counts there will be
- 20K words => 400M 2-grams => 8000G 3-grams, so even a gigaword corpus has MANY zero counts!
- **Equivalence classes**: Cluster several similar n-grams together to reach higher counts
- **Smoothing**: Redistribute some probability mass from seen N-grams to unseen ones

16/

Smoothing methods

1. **Add-one**: Add 1 to each count and normalize => gives too much probability to unseen N-grams
2. **(Absolute) discounting**: Subtract a constant from all counts and redistribute this to unseen ones using N-1 gram probs and back-off (normalization) weights
3. **Witten-Bell smoothing**: Use the count of things seen once to help to estimate the count of unseen things
4. **Good Turing smoothing**: Estimate the rare n-grams based on counts of more frequent counts
5. Best: **Kneser-Ney smoothing**: Instead of the number of occurrences, weigh the back-offs by the **number of contexts** the word appears in
6. Instead of only back-off cases, **interpolate** all N-gram counts with N-1 counts

Weaknesses of N-grams

- Skips long-span dependencies:
  - “The **girl** that I met in the train **was** ...”
- Too dependent on word order:
  - “dog chased **cat**”: “koira jahtasi **kissaa**” ~ “**kissaa** koira jahtasi”
- Dependencies directly between words, instead of latent variables, e.g. word categories

Continuous space LMs

- Alleviates the data sparsity problem by representing words in a distributed way
- Various algorithms can be used to learn the most efficient and discriminative representations and classifiers
- The most popular family of algorithm is called (Artificial) **Neural Networks (NN)**
  - can learn very complex functions by combining simple computation units in a hierarchy of non-linear layers
  - Fast in action, but training takes a lot of time and labeled training data
- Can be seen as a non-linear multilayer generalization of the maximum entropy model

3. Sentence-level processing

- Tagging words in a sentence
  - Part-of-speech tagging
  - Named entity recognition
  - Solving ambiguities
  - Hidden Markov model, Viterbi, Baum-Welch, Forward-Backward
  - Recurrent neural networks
- Sentence parsing
  - Grammars, trees
  - Probabilistic context free grammar

Part of Speech (POS) tagging

Task: Assign tags for each word in a sentence

Applications: Tool for parsing the sentence

The reaction in the newsroom was emotional.

=> *DT NN IN DT NN VBD JJ*

(determiner)

(noun) (preposition)

(determiner) (noun)

(verb past tense) (adjective)

Named entity recognition

- Detect names of persons, organizations, locations
- Detect dates, addresses, phone numbers, etc
- Applications: Information retrieval, ontologies
-     UN official Ekeus heads for Baghdad.
- => ORG - PER - - LOC
- (organization) (person) (location)

A general approach

1. Generate candidates
2. Score the candidates
3. Select the highest scoring ones

A simple scoring method

- Count the frequency of each tagging by listing all appearances of the word in an annotated corpus
- Select the most common tag for each word
- How well would this method work?

Estimation of HMM parameters

- For corpora annotated with POS tags
  - Just count each tag observations P(y(t)|x(t)
  - And tag transitions P(y(t)|y(t-1))
- For unknown data use e.g. Viterbi to first estimate labels and then re-estimate parameters and iterate

Even better POS tags? Discriminative models

- Use previous words and tags as features
- The context is computed from a sliding window
- Train a classifier to predict the next tag
  - Jurafsky: Maximum entropy Markov model (MEMM)
  - Support vector machine (SVM)
  - Deep (feed-forward) neural network (DNN)
  - Conditional random field (CRF) is a bidirectional extension of MEMM that uses also tags on right

4. Word2vec

- Vector space models, distributional semantics
  - ~ word-document and word-word matrices
  - ~ Constructing word vectors
  - ~ stemming, weighting, dimensionality reduction
  - ~ similarity measures
  - ~ Count models vs. predictive models
  - ~ Word2vec
- Information retrieval

How to build a vector space model?

1. Preprocessing
2. Defining word-document or word-word matrix
   - choosing features
3. Dimensionality reduction
   - choosing features
   - removing noise
   - easing computation
4. Weighting and normalization
   - emphasizing the features
5. Similarity / distance measures
   - comparing the vectors

To count or predict?

Count-based methods

- compute the word co-occurrence statistics with its neighbor words in a large text corpus
- followed by a mapping (through weighting and dimensionality reduction) to dense vectors

Predictive models

- try to predict a word from its neighbors by directly learning a dense representation

Statistical semantics

Statistical semantics hypothesis: Statistical patterns of human word usage can be used to figure out what people mean (Weaver, 1955; Furnas et al., 1983).

Bag of words hypothesis: The frequencies of words in a document tend to indicate the relevance of the document to a query (Salton et al., 1975).

Distributional hypothesis: Words that occur in similar contexts tend to have similar meanings (Harris, 1954; Firth, 1957; Deerwester et al., 1990).

Latent relation hypothesis: Pairs of words that co-occur in similar patterns tend to have similar semantic relations (Turney et al., 2003).

Modifying the vector spaces

The basic matrix formulation offers lots of variations:

- window sizes
- word weighting, normalization, thresholding, removing stopwords
- stemming, lemmatizing, clustering, classification, sampling
- distance measures
- dimensionality reduction methods
- neural networks

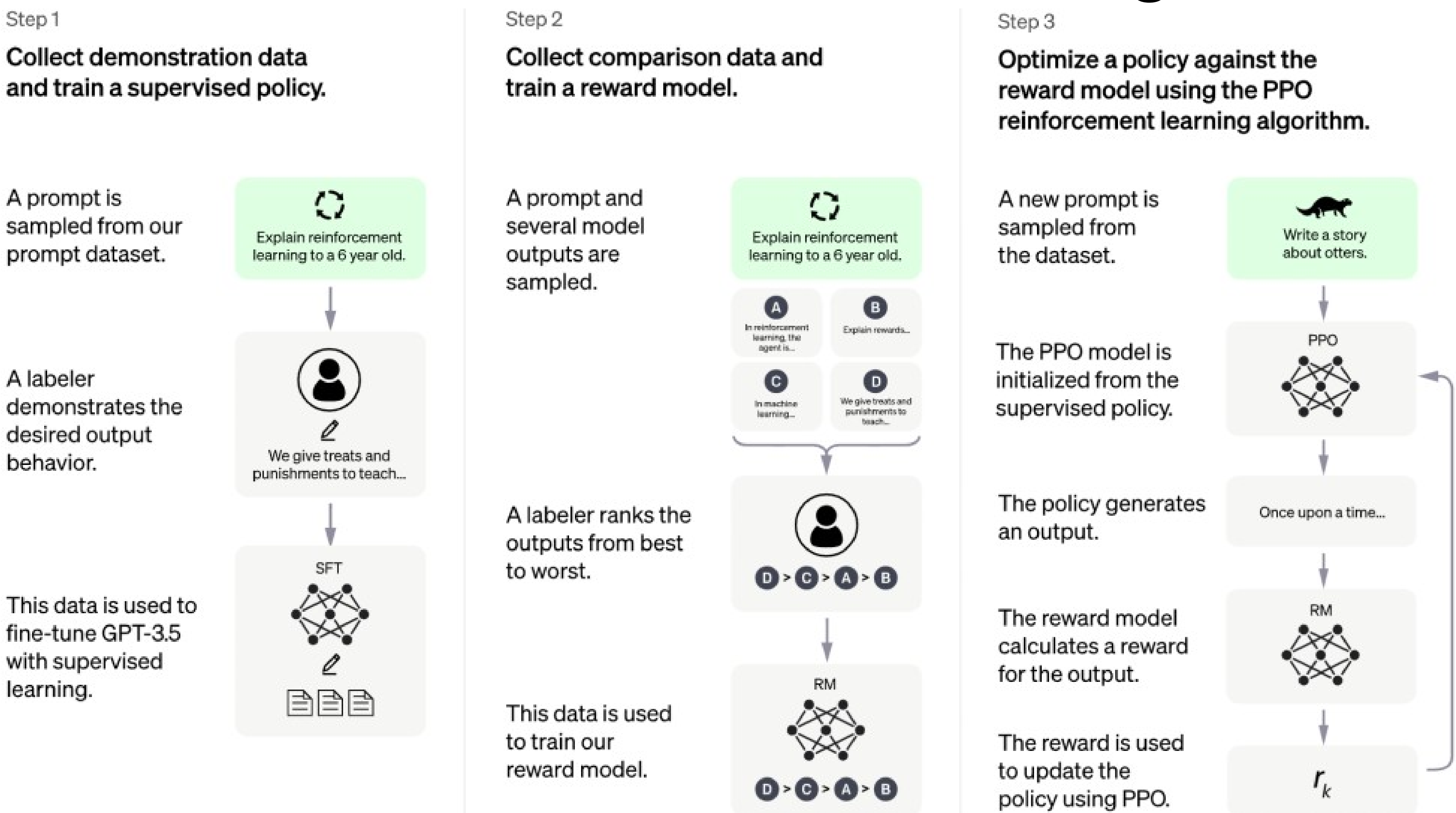
5. Neural Network LM and LLMs

- Feed-Forward and Recurrent NNLM
- LSTMs, Transformers
- Masked LMs, Bidirectional LMs, BERT
- Reinforcement Learning, ChatGPT

Transformers for Language Modelling

- RNNs: Process tokens one-by-one
  - Chain of dependencies built using a single token
- Transformers LM: Process a segment of tokens
  - Dependencies within the segment
  - Within segment position is given by the positional encoding

Reinforcement Learning



6. Morpheme-level processing

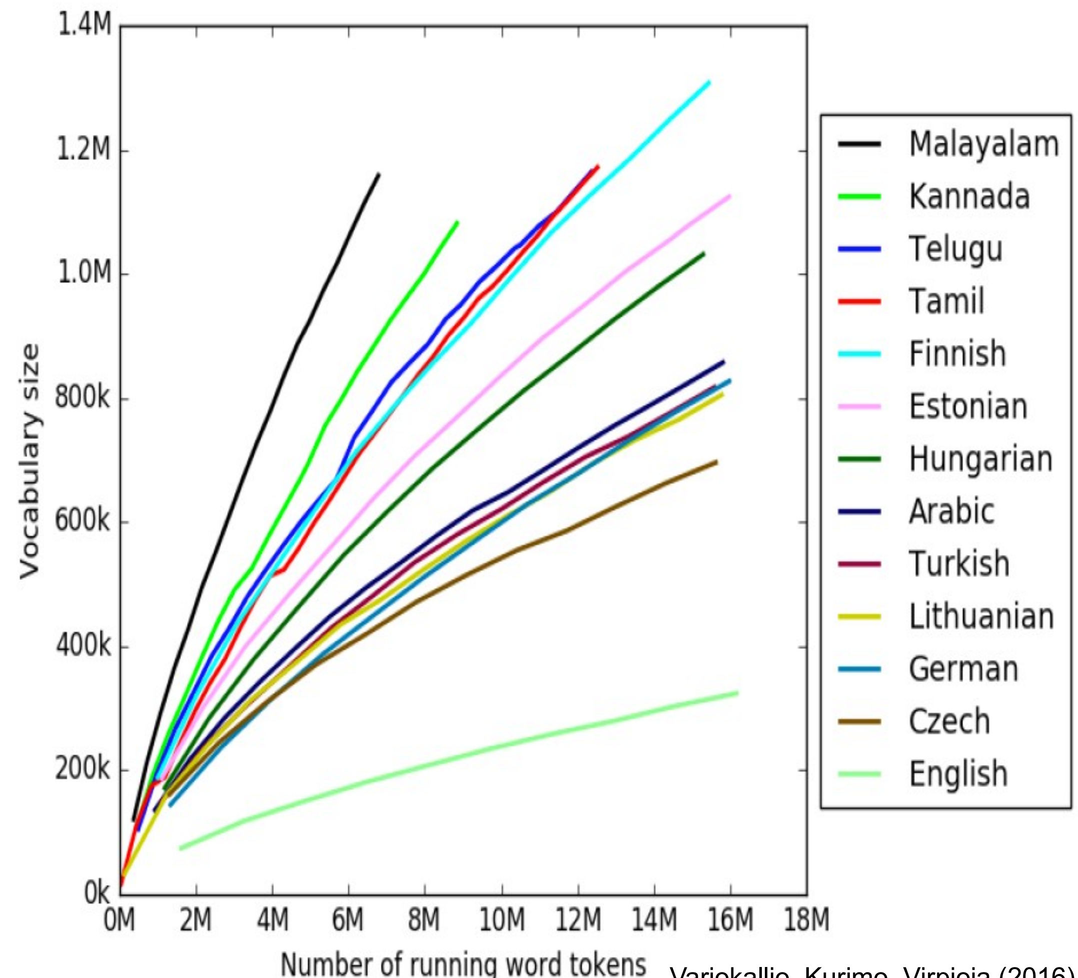
- Morphemes
  - morphological complexity, processes, models
- Morphological clustering
  - stemming, lemmatization
- Morphological analysis and generation
  - Finite-state methods, transducers
- Morphological segmentation
  - Morfessor

Types of morphemes

- **Root:** a portion of word without any affixes; carries the principle portion of meaning (buildings ℗ build)
- **Stem:** a root, or compound of roots together with derivational affixes (buildings ℗ building)
- **Affix:** a bound morpheme (does not occur by itself) that is attached before, after, or inside a root or stem
  - **Prefix** (**un**-happy)
  - **Suffix** (build-**ing**, happi-**er**)
  - **Infix** (abso-**bloody**-lutely)
  - …

Morphology affects the vocabulary size

Vocabulary size as a function of corpus size



Varjokallio, Kurimo, Virpioja (2016)

Morphological processes

Inflection:

- cat – cats
- slow – slower
- find – found

Derivation:

- build (V) – building (N)
- do (V) – doable (ADJ)
- short (ADJ) – shorten (V)
- write – rewrite
- do – undo

Compounding:

- fireman (fire + man)
- hardware (hard + ware)

3 main approaches to deal with rich morphology

1. **"Canonical" forms of a word**
   - Stemming is relatively simple and implementations are available, for English: e.g., Porter (1980), Snowball: http://snowball.tartarus.org
   - Lemmatization is more complex and needs morphological analysis
   - Applications: Information retrieval etc.
2. **Morphological analysis**
   - Hand-crafted morphological analyzers/generators exist for many languages, e.g. Tähtien => tähti N Gen Pl
   - Applications: Spell checking, syntactic parsers, machine translation, etc.
3. **Segmentation into morphs**
   - Pragmatic approaches that work well enough in practice.
   - Applications: Speech recognition, language modeling etc.

Morfessor

- Instead of sending over the vocabulary as it is, we split it into two parts:
    1. a fairly compact lexicon of morphs: “aamu”, “aurinko”, “ksi”, “lla”, …
    2. the word vocabulary expressed as sequences of morphs

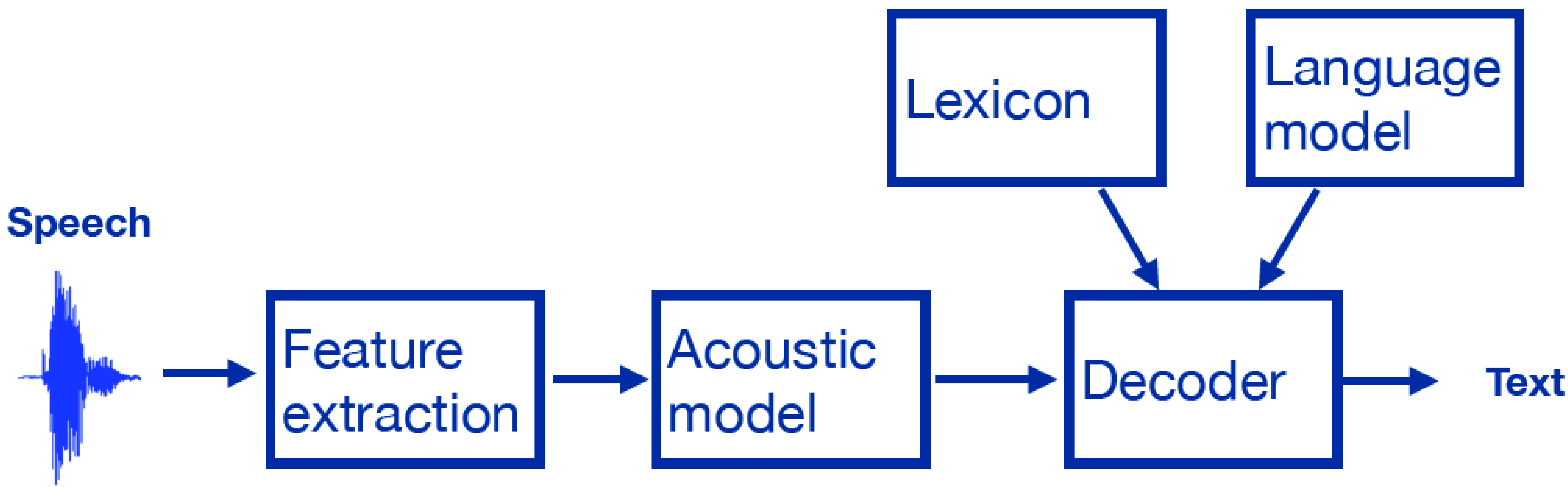
- Since we are doing unsupervised learning, we do not know the correct answer.

- Our target is to **minimize the combined code length** of:
    1. the code length of the morph lexicon
    2. plus the code length of the word vocabulary expressed using the morph lexicon.

7. Speech recognition

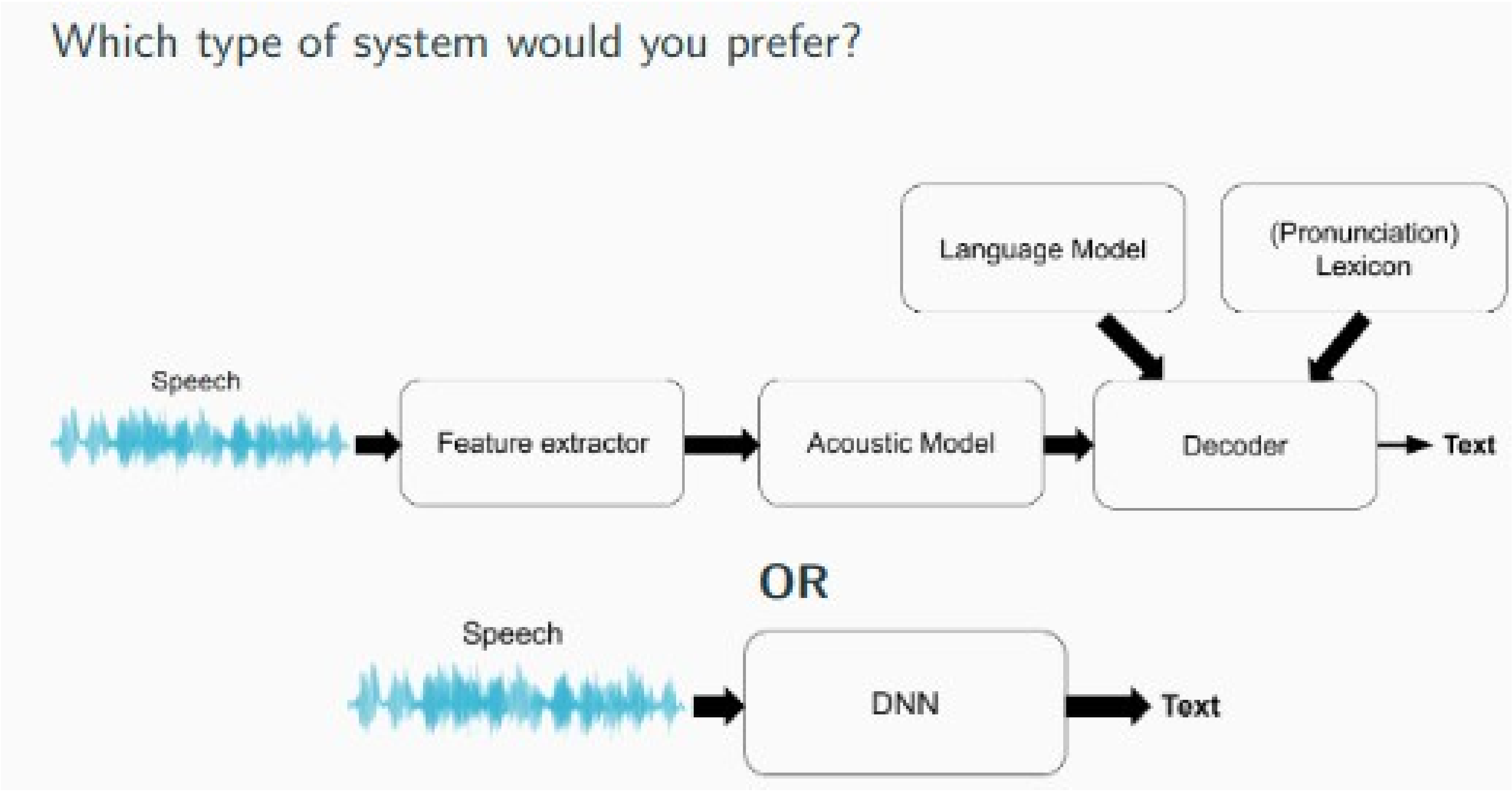
- Acoustic features
- Gaussian mixture models
- Hidden Markov models
- Deep neural networks for acoustic modeling
- Phonemes, pronunciation of words
- Decoding with language models
- End-to-end neural networks
- Encoder, Attention, Decoder

Traditional Components of an ASR system



- **Task of an automatic speech recognition: Find the most likely word sequence given the observations (speech) and the models for acoustics and language**
- **Speech acoustics are matched with a statistical model**
- **Language model is also typically a statistical model (n-gram, RNN), but in simple tasks it can be a fixed grammar or just a vocabulary**

End-to-end ASR



End-to-end ASR systems

- CTC
- RNN transducer
- Attention-based Encoder-Decoder
- Transformers
- Self-supervised pre-training
  - Wav2vec
  - HuBERT

8. Chatbots and dialogue agents

- True chatbots vs task-oriented dialogue agents
- Text processing steps in chatbots
- Chatbot architectures
- Evaluation of chatbots

Definitions

Chatbot:

- A system that **you can chat** with
- Discussion topics can be fixed, but there is **no specific goal** except for fun and keeping company

Dialogue agent:

- A system that helps you to **reach a specific goal** by giving and collecting information by **answering and asking questions**

In popular media both are often called chatbots, but here only the first one.

Comparison of chatbots and dialogue agents: required operations

Chatbot

- Detect the discussion topic
- Ask typical questions
- React to human input, be coherent with previous turns
- World knowledge, persona

Dialogue agent

- Detect the user's intent
- Ask the required questions
- Parse and use human input

Chatbot architectures

Rule-based

- Pattern-action rules: Eliza (1966)
- Mental model: Parry (1971)

Corpus-based

- IR: Cleverbot
- DNN encoder-decoders etc

Turing's test (1950) for machine intelligence: *Can you judge between a real human and a chatbot?*

Evaluation of chatbots

Automatic evaluation

- Lack of proper evaluation data and metrics
- N-gram matching evaluations such as BLEU correlate poorly with human evaluation
  - Too many correct answers
  - Common words give a good score
- Perplexity measures predictability using a language model
  - Favours short, boring and repetitive answers
- ADEM classifier trained by human judgements
- Adversarial evaluation trained to distinguish human and machine responses

Human evaluation

e.g. research challenges (competitions):

- ConvAI (NeurIPS)
- Dialog Systems Technology Challenge (DSTC7)
- Amazon Alexa prize
- Loebner Prize

9. Statistical machine translation

- Sentence, word and phrase alignment methods
- Re-ordering models
- Translation methods
- Full SMT systems

Phrase-based SMT system

- Training data and data preprocessing
- Word aligment, phrase aligment
- Estimation of translation model scores
- Estimation of reordering model scores
- Estimation of language model scores
- Decoding algorithm and optimization of the model weights
- Translation, recasing, detokenization
- Evaluation, quality estimation
- Operational management

Translation methods

- Phrase-based beam-search decoder (e.g. Moses)
- Weighted finite state transducer (WFST) based translation models
- Extended word-level representations, e.g., hierarchical phrase-based models and factored translation models with words augmented with POS tags, lemmas, etc.
- Syntax-based translation models, which take syntax parse trees as input.
- Feature-based models, where translation is performed between features. E.g., discriminative training or exponential models over feature vectors

Evaluation of machine translation

- Human evaluation
  - Assessment, ranking, agreement, efficiency
- Automatic evaluation
  - Challenges?
  - Edit distance metrics
  - Precision & recall, BLEU
  - Beyond word-based metrics
- Meta-evaluation
  - Evaluation of evaluation metrics

10. Neural machine translation

- Why NMT is the mainstream * approach?
- How are the current NMT systems build?
- What are the challenges and limitations for the systems?
- Transfer learning
- Massively multilingual machine translation
- Using monolingual corpora
- Large language models as translators

Why NMT?

- Generalization
- Flexibility
- Integration

Building NMT

- Encoding variable-length sequences
- Sequence decoding
- Sequence-to-sequence models
- Recurrent neural networks
- Attention model, Transformer model
- Modeling units

Building NMT

- Encoding variable-length sequences
- Sequence decoding
- Sequence-to-sequence models
- Recurrent neural networks
- Attention model, Transformer model
- Modeling units

Future of NMT

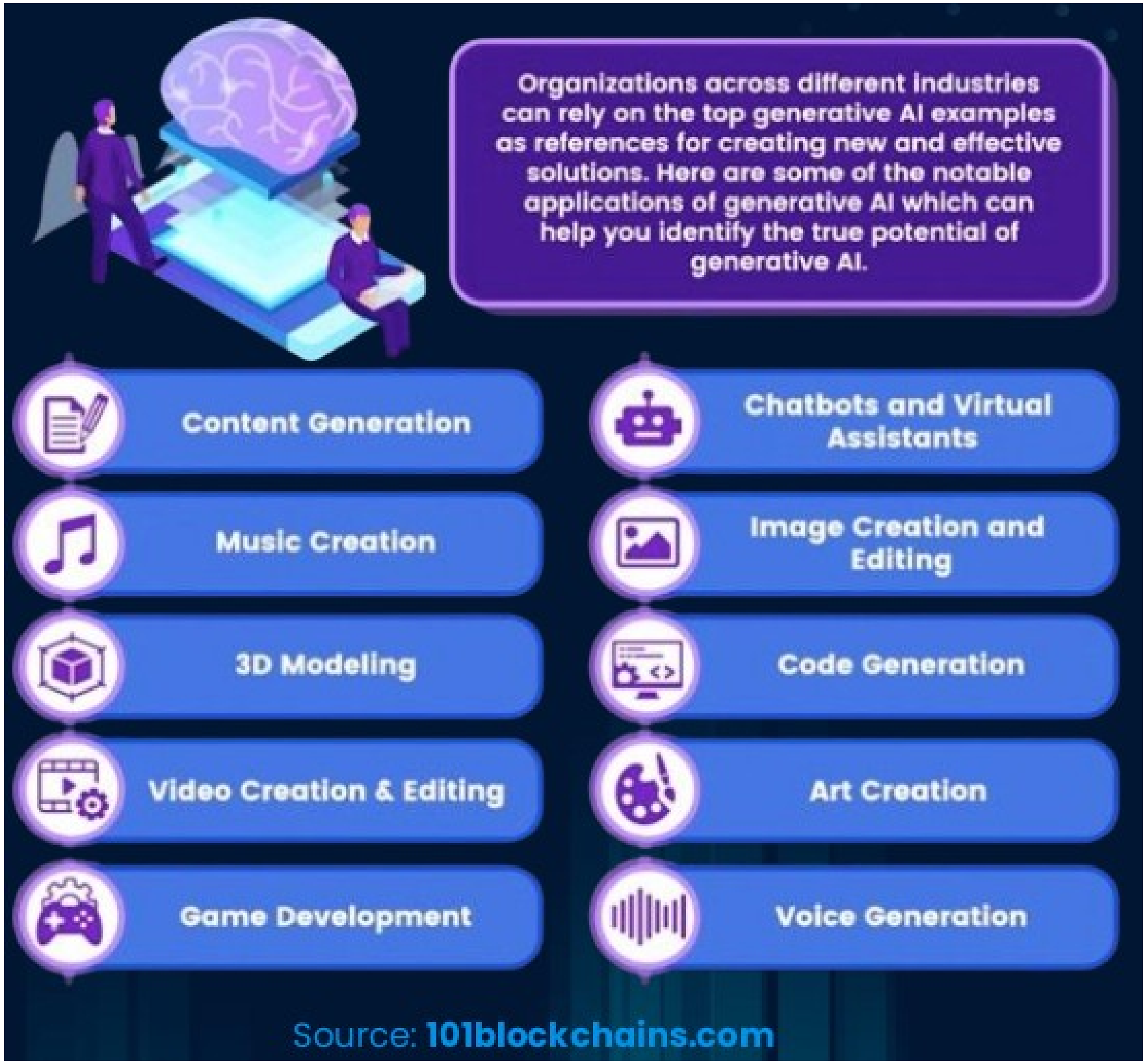
If LLMs are already so good translators, do we need separate NMT models?

- Multiple questions: Architecture, size, training.
- Size: “GPT-scale” LLMs have NMT challenges multiplied: computation, memory, ...
  - Only large corporations have resources for training.
  - Not possible to run the models locally.
  - Environmental impact.
- Training: Adapting conversational LLMs to specific tasks and domains is not simple (prompt engineering)

11.Large Language Models (LLMs) In Industries

- Overview
  - Generative AI
  - Language Model
  - Large Language Models
- LLMs in Industries
- Use Cases
- Case Study

Capability of Generative AI



LLMs Use Cases Across Industries

- Customer Experience and Support
  - Chatbots
  - Personalized recommendation
- Banking and Finance
  - Financial analysis and research
  - Fraud detection
  - Risk assessment
- E-commerce and Retail
  - Product description and reviews
  - Inventory management and demand forecasting
- Healthcare
  - Clinical documentation automation
  - Patient assistant
  - Compliance management
- Cybersecurity
  - Threat detection and analysis
  - Incident response
- Marketing and advertising
  - Personalized marketing
  - Generating creative text

Can large language models replace humans?

People in industries are going to lose their jobs ?

- adopt a human-in-the-loop approach rather than replacing everyone with an LLM
- LLMs often referred to as black boxes and important to continually evaluate and test the results of LLMs.
- subject matter experts are the best candidates to judge the quality of an LLM's output.

Lecture schedule 2024

09 Jan 1 Introduction & course organization / Mikko Kurimo

16 Jan 2 Statistical language models / Mikko Kurimo

23 Jan 3 Sentence level processing / Mikko Kurimo

30 Jan 4 Word2vec / Tiina Lindh-Knuutila

06 Feb 5 Neural language modeling and LLMs / Mittul Singh

13 Feb 6 Morpheme-level processing / Mathias Creutz

20 Feb (exam week, no lecture)

27 Feb 7 Speech recognition / Tamas Grosz

05 Mar 8 Chatbots and dialogue agents / Mikko Kurimo

12 Mar 9 Statistical machine translation / Jaakko Väyrynen

19 Mar 10 Neural machine translation / Sami Virpioja

26 Mar 11 LLMs in industry / Shantipriya Parida

02 April (spring break - no lecture)

09 Apr 12 Course conclusion / Mikko Kurimo

16 April Exam