

# **Neural Networks for Natural Language Processing**

**Jun.-Prof. Sophie Fellenz**

**Week 01 - Introduction to NLP and Applications**

21 Oct 2024

# Agenda

- Who are we?
- What is NLP?
- What are neural networks/deep learning?
- Course Overview
- Applications/Tasks

## Group Intro

- Junior Professor for Machine Learning since 2020
- Work mostly on probabilistic and generative models
  - Applications in chemistry, engineering, text, biomedicine...

**Prof. Dr.  
Marius Kloft**



**Jun.-Prof. Dr.  
Sophie Fellenz**



**Steffen  
Reitherrmann  
(Lab Manager)**



**Philipp Liznerski  
(Oberinformatiker)**



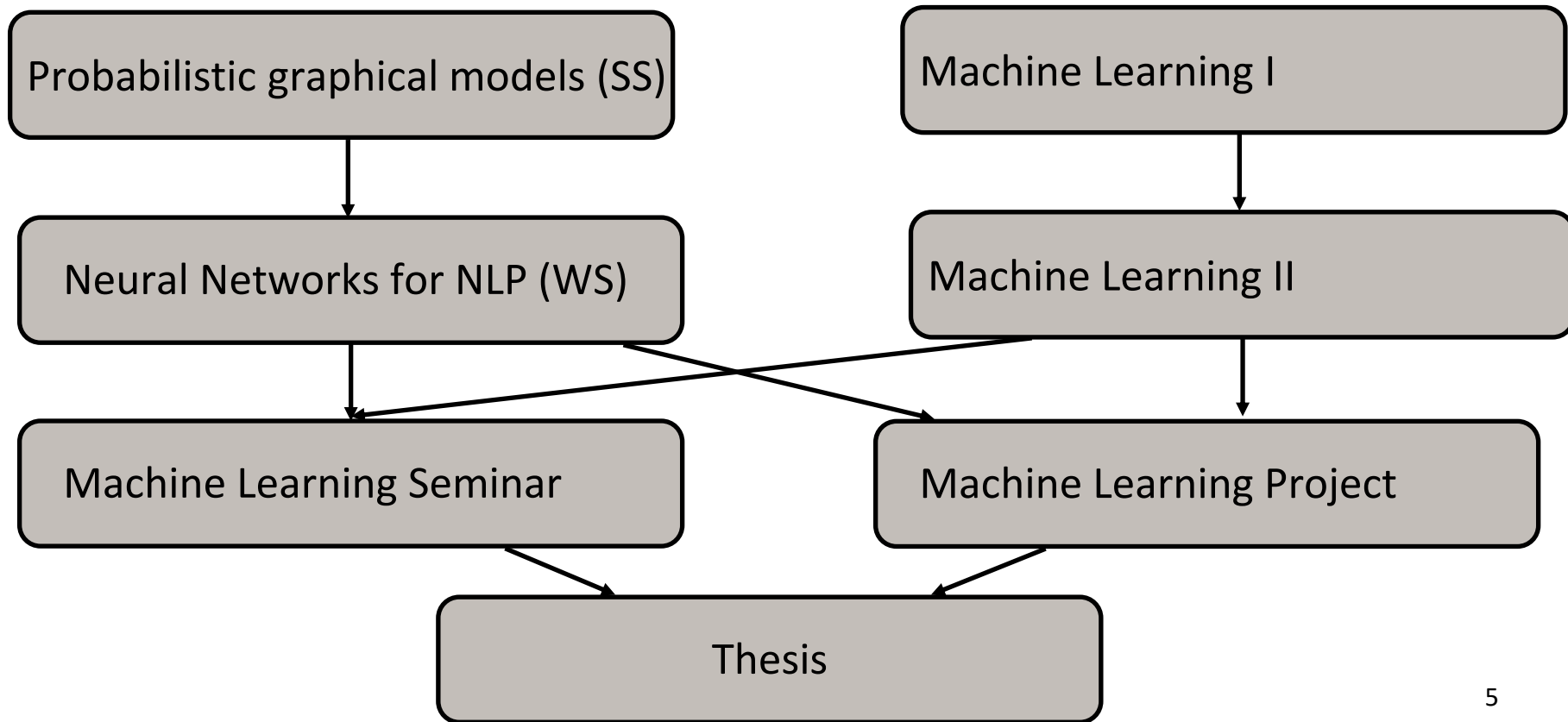
## Leitung:

## Vollzeitkräfte:



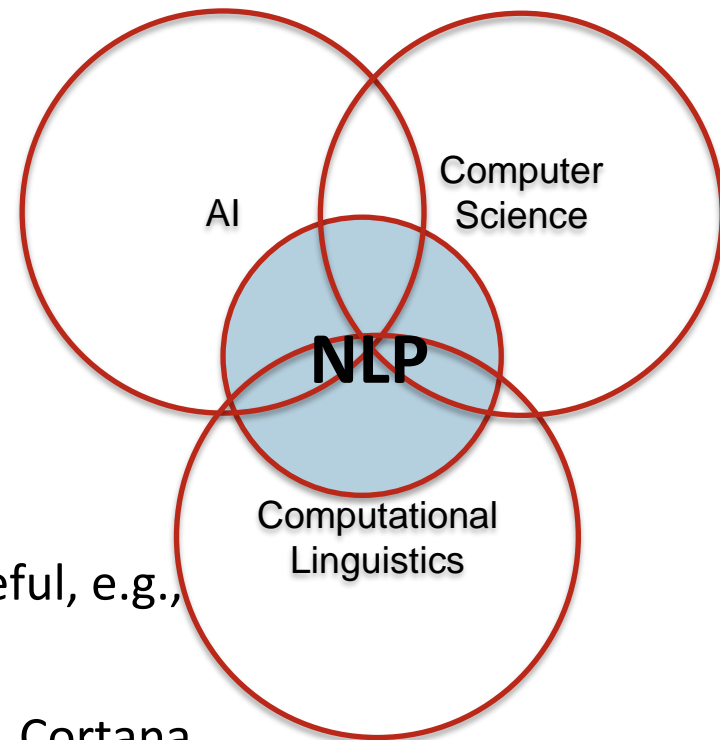
## Teilzeitkräfte:



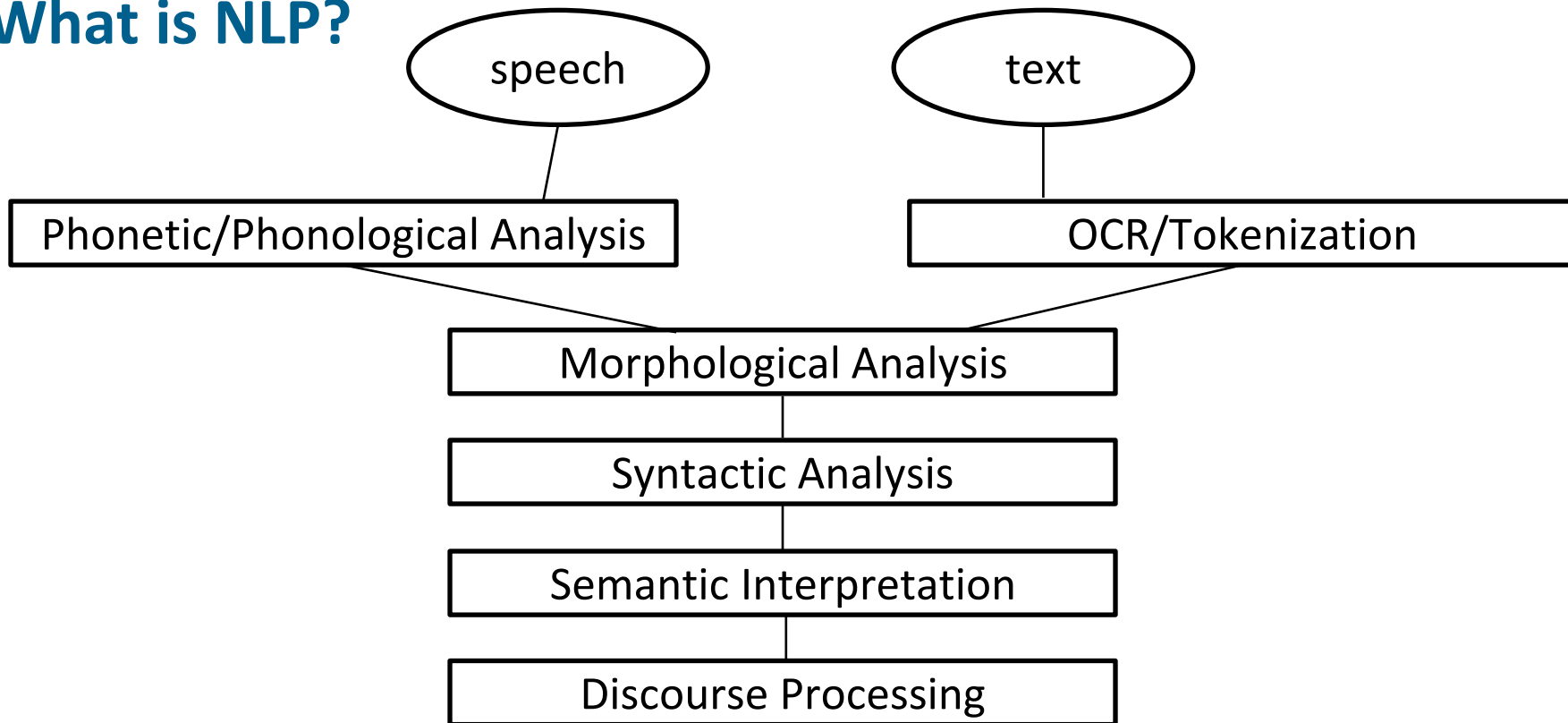


## What is NLP?

- **Natural language processing** is a field at the intersection of
  - computer science
  - artificial intelligence
  - and linguistics.
- NLP vs. Computational Linguistics
- **Goal:** for computers to process natural language in order to perform tasks that are useful, e.g.,
  - making appointments, buying things
  - Language translation
  - Question Answering: Siri, Google Assistant, Cortana, Alexa, ChatGPT ...



## What is NLP?



## (A tiny sample of) NLP Applications

- Applications range from simple to complex:
  - Spell checking, keyword search, finding synonyms
  - Extracting information from websites such as product price, dates, location, people or company names
  - Classifying: reading level of school texts, positive/negative sentiment of longer documents
  - Machine translation
  - Spoken dialog systems
  - Complex question answering



# NLP in industry

- Search (written and spoken)
- Online advertisement matching
- Automated/assisted translation
- Sentiment analysis for marketing or finance/trading
- Speech recognition
- Chatbots / Dialog agents
  - Automating customer support
  - Controlling devices
  - Ordering goods

# What's special about human language?

A human language is a system specifically constructed to convey the speaker/writer's meaning

- Not just an environmental signal, it's a deliberate communication
- Using an encoding which little kids can quickly learn and which changes

A human language is mostly a **discrete/symbolic/categorical** signaling system

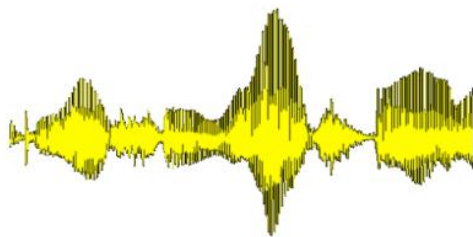
rocket =  ; violin = 

- Presumably because of greater signaling reliability
- Symbols are not just an invention of logic / classical AI!

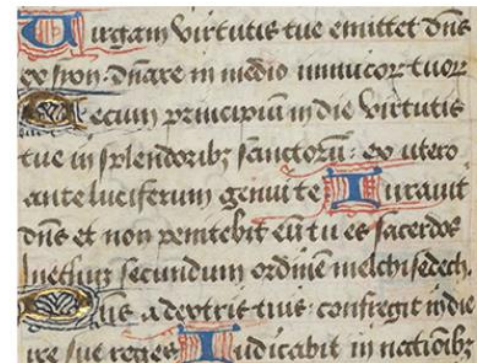
# What's special about human language?

The categorical symbols of a language can be encoded as a signal for communication in several ways: Sound, Gesture, Writing/Images

The symbol is invariant across different encodings!



CC BY 2.0 David Fulmer 2008



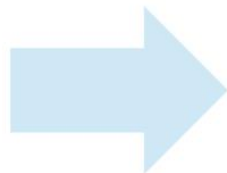
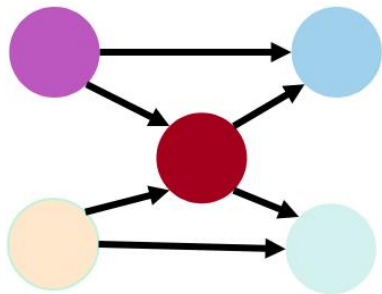
National Library of NZ, no known restrictions

## What's special about human language?

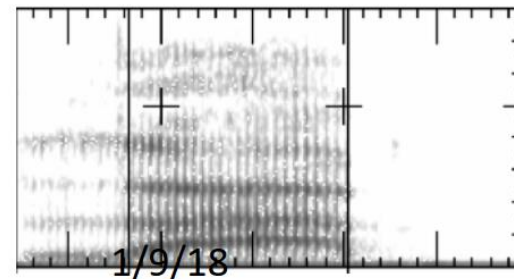
A human language is a **symbolic/categorical signaling system**

However, a brain encoding appears to be a **continuous pattern of activation**, and the symbols are transmitted via **continuous signals** of sound/vision

The large vocabulary, symbolic encoding of words creates a problem for machine learning – sparsity!



lab



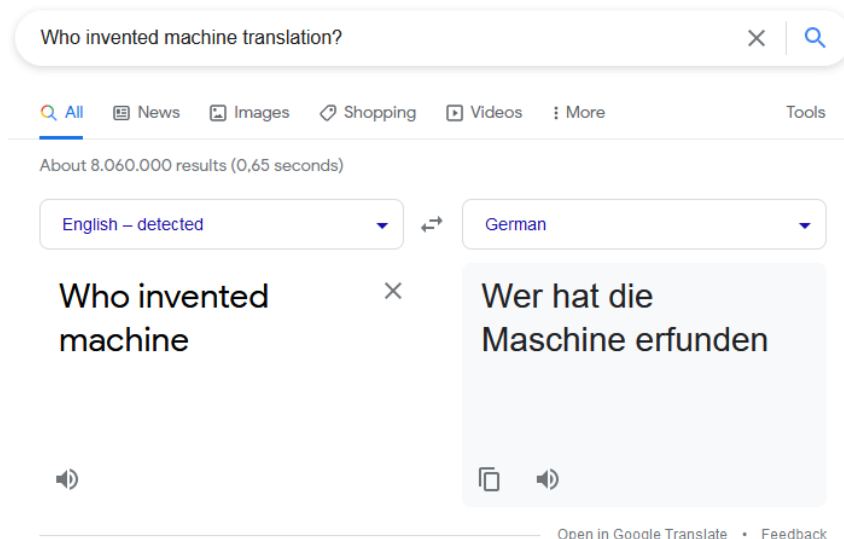
## Why study NLP?

- Two possible motivations behind research in AI, including NLP:
- **Technological:**
  - The goal here is to achieve the *best* performance on some *applied* task.
- **Cognitive:**
  - The goal here is to have a single model with human-like performance for any task across the board (in some domain).
  - Achieve biological/cognitive plausibility.

## Why study NLP?

- Helps in better and easier (text) communication between any two agents
- Helps in better **human-human** communication
  - machine translation
  - text generation
- Helps in better **human-machine** communication
  - dialog systems / conversational AI / chatbots
  - natural language understanding

# Language is challenging



We asked Google - **“Who invented machine translation?”**

Google thought we asked it to translate - **“Who invented machine”**.

# Language is challenging

- There are many things that computers are unable to understand well. For instance:
  - jokes, sarcasm, metaphors, etc...
  - Meaning of a sentence may vary based on the tone of voice used.
  - Context is often important
  - „The pope’s baby steps on gays“



## **NLP**

**How was it? How is it now?**

## Feature Engineering (till 2015)

- Fully supervised learning
- Traditional non-neural network machine learning models
- Manually defined feature extraction
- Notable methods to extract manual features:
  - Linear or Kernel SVMs
  - Conditional Random Fields (CRFs)

## Architecture Engineering (2013 - 2018)

- Still fully supervised learning
- People started to use neural network models
- No need to manually define features, but network structure is important (e.g. LSTM vs. CNN)
- Some models are pre-trained, mainly for shallow text features
- Notable work:
  - CNNs for text classification

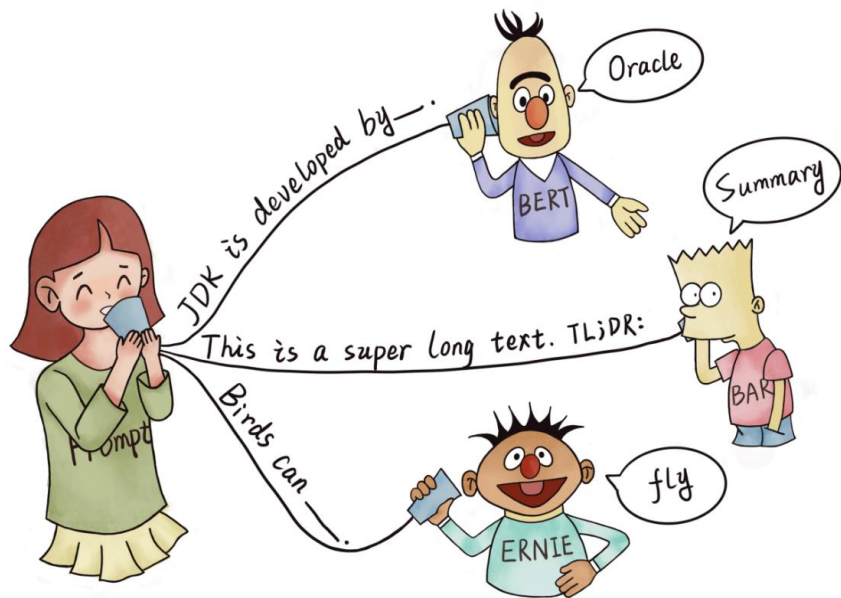
## Objective Engineering (2018 - present)



Source: tengoldenrules.com

- Pre-trained models used as initial models
- Fine-tuned for specific tasks
- More focus on objective functions than architecture
- Notable work:
  - BERT → fine-tuning for tasks

## Prompt Engineering (2019 - present)



Source: phontron.com

- NLP tasks are modeled entirely based on models
- The model extracts features as well as predicts the result
- Engineering of **prompts** needed
- Notable work:
  - GPT-3 from OpenAI

## (a) Few-shot

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A:

(Output) The answer is 8. ✗

## (b) Few-shot-CoT

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls.  $5 + 6 = 11$ . The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A:

(Output) The juggler can juggle 16 balls. Half of the balls are golf balls. So there are  $16 / 2 = 8$  golf balls. Half of the golf balls are blue. So there are  $8 / 2 = 4$  blue golf balls. The answer is 4. ✓

## (c) Zero-shot

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A: The answer (arabic numerals) is

(Output) 8 ✗

## (d) Zero-shot-CoT (Ours)

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A: **Let's think step by step.**

(Output) There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls. ✓

Source: phontron.com

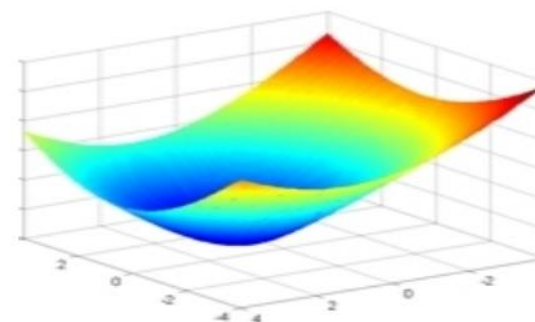
Source: Kojima et al. 2022

# Neural Networks / Deep Learning

# What is Deep Learning (DL)?

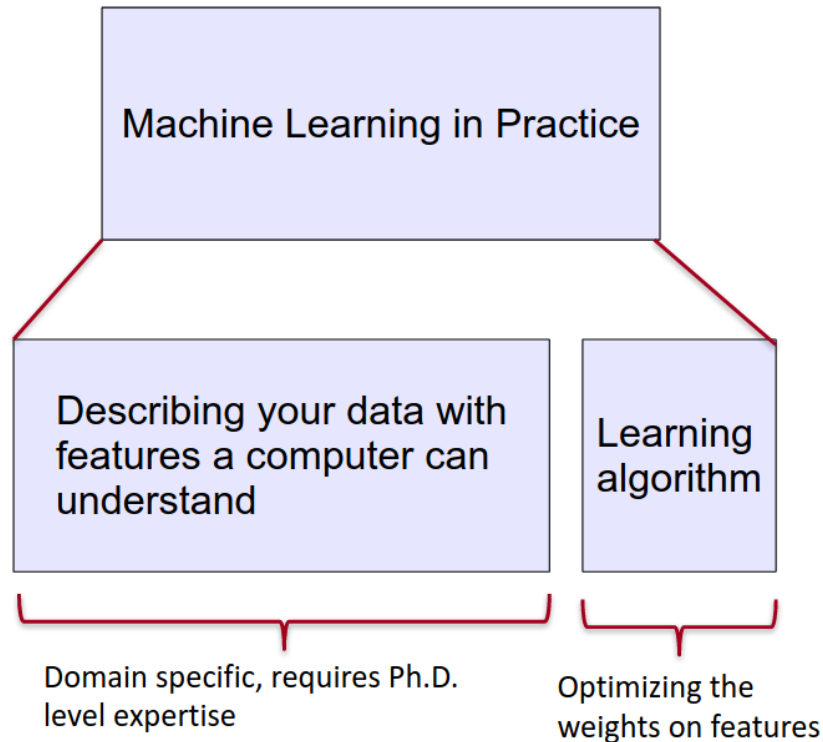
- Deep learning is a subfield of machine learning
- Most machine learning methods used to work well because of human-designed representations and input features
- For example: features for finding named entities like locations or organization names (Finkel et al., 2010):
- Machine learning becomes just optimizing weights to best make a final prediction

Feature	NER
Current Word	✓
Previous Word	✓
Next Word	✓
Current Word Character n-gram	all
Current POS Tag	✓
Surrounding POS Tag Sequence	✓
Current Word Shape	✓
Surrounding Word Shape Sequence	✓
Presence of Word in Left Window	size 4
Presence of Word in Right Window	size 4





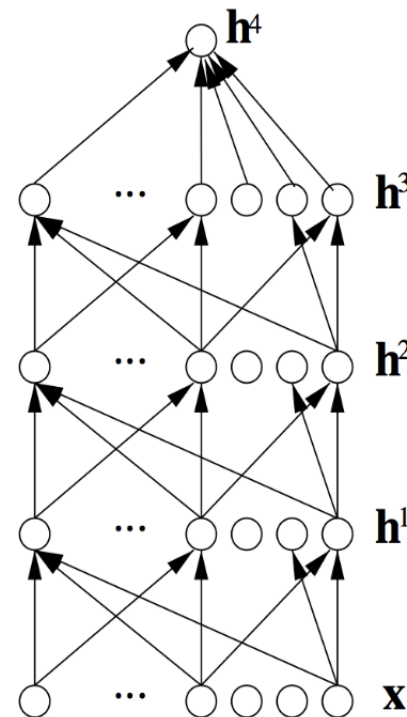
# Machine Learning vs. Deep Learning



# What is Deep Learning (DL)?

In contrast to standard machine learning,

- Representation learning attempts to automatically learn good features or representations
- Deep learning algorithms attempt to learn (multiple levels of) representations (here:  $h^1, h^2, h^3$ ) and an output ( $h^4$ )
- From “raw” inputs  $x$  (e.g. sound, pixels, characters, or words)



# On the history of “Deep Learning”

- We will focus on different kinds of neural networks
  - The dominant model family inside deep learning
  - No historical approach but instead focus on methods which work well on NLP problems now
  - For a long history of deep learning models (starting ~1960s), see: Deep Learning in Neural Networks: An Overview by Jürgen Schmidhuber

# Course Overview

# What do we hope to teach?

1. An understanding of and ability to use the effective modern methods for deep learning
  - Basics first, then key methods used in NLP: Recurrent networks, attention, etc.
2. Some big picture understanding of human languages and the difficulties in understanding and producing them
3. An understanding of and ability to build systems for some of the major problems in NLP:
  - Word similarities, machine translation, entity recognition, question answering, chat bots

## Organization

- Lectures will be held once a week in person (no video upload planned)
- Additional Lectures will be announced as necessary (deepen individual topics)
- Office Hours: Wednesdays 1-2pm
- Course Information: OpenOlat

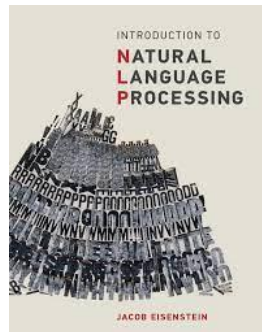
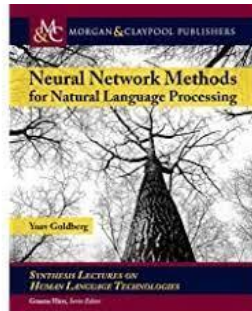
<https://olat.vcrp.de/url/RepositoryEntry/4696769253>

- Questions can be asked on Mattermost [https://ml-chat.cs.uni-kl.de/signup\\_user\\_complete/?id=uh4xb7z3nbdx9gec4ojarjj85h](https://ml-chat.cs.uni-kl.de/signup_user_complete/?id=uh4xb7z3nbdx9gec4ojarjj85h)
- Tutorials will start on **November 5th**, the first exercise sheet will come out after the second lecture, time **Tuesday 10am, room 48-453**
- Tutors: **Mayank Ahuja**
- Work in groups (3 people)
- In general: exercises start Monday after the lecture, two weeks to complete

## Organization cont.

- Exercises are mandatory (50% of points necessary)
- Prerequisites: Basic Python and ML knowledge
- Written Exam, Date 6th of March, 9am
- Admission: 50% of all exercise sheets
- Exercises: practical and theoretical exercises
- Groups of 3 people
- There should always be someone present from each group who can present their solution in the tutorial
- Exercise sheets will be published on Olat and can be submitted on Olat
- Slides are not the only sources for learning

## Literature



- Goldberg, Yoav. "Neural network methods for natural language processing." Synthesis lectures on human language technologies 10.1 (2017): 1-309.
- Eisenstein, Jacob. "Introduction to Natural Language Processing." Adaptive Computation and Machine Learning series (2019).



# Overview

Week 1 (21 Oct 2023)	Intro to NLP
Week 2 (18 Oct 2023)	Text Preprocessing and Representation
Week 3 (04 Nov 2023)	Text Embeddings <b>Tutorial 1</b>
Week 4 (11 Nov 2023)	Neural Networks (crash course)
Week 5 (18 Nov 2023)	Neural Models for NLP <b>Tutorial 2</b>
Week 6 (25 Nov 2023)	Attention and Transformers-based LMs
Week 7 (02 Dec 2023)	Attention and Transformers-based LMs <b>Tutorial 3</b>

**Ex 1 released**

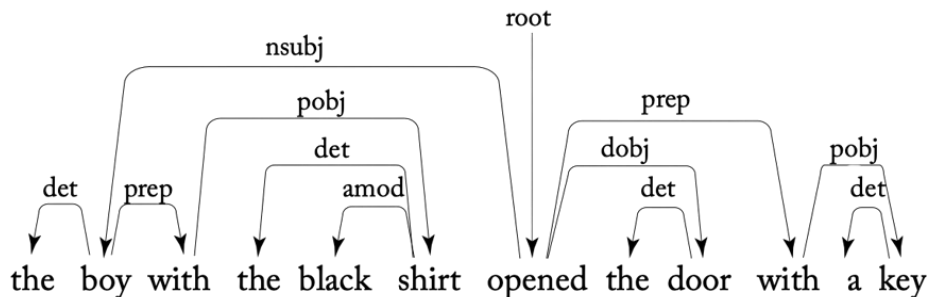
**Deadline Ex 1  
Ex 2 released**

**Deadline Ex 2  
Ex 3 released**

# Overview

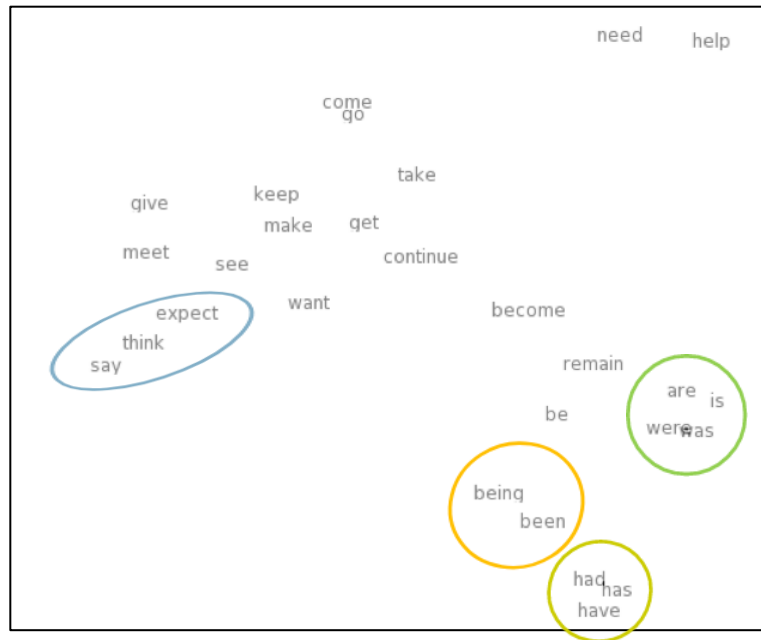
Week 8 (09 Dec 2023)	Fine-tuning, Pre-training, Transfer Learning <b>Tutorial 4</b>	<b>Deadline Ex 3 Ex 4 released</b>
Week 9 (16 Dec 2023)	Reinforcement Learning for NLP	
Week 10 (06 Jan 2024)	Reinforcement Learning for NLP <b>Tutorial 5</b>	<b>Deadline Ex 4 Ex 5 released</b>
Week 11 (13 Jan 2024)	Advanced Fine-Tuning/RAG	
Week 12 (20 Jan 2024)	Self-Supervised Learning/ Style Transfer <b>Tutorial 6</b>	<b>Deadline Ex 5 Ex 6 released</b>
Week 13 (27 Jan 2024)	Topic Models <b>Tutorial 7</b>	<b>Deadline Ex 6</b>
Week 14 (03 Feb 2024)	QA session	

## Week 02 - Text Data and Pre-processing



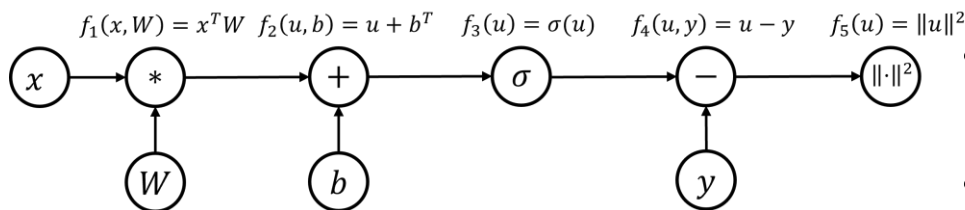
- How is text data different?
- Words, Characters, Sentences ...
- What is a word/token?
- How do we deal with text?
- Do we need predefined features?

## Week 03 - Text Embeddings



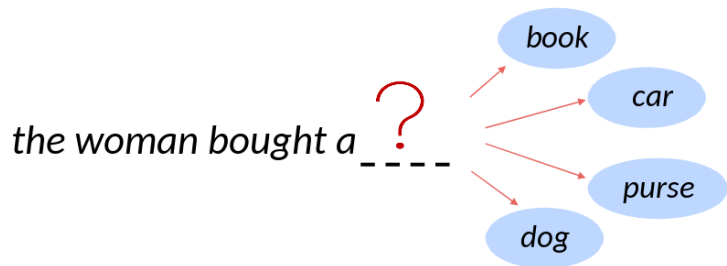
- Why are embeddings needed?
- Word embeddings
- Sentence embeddings
- Different embedding models
  - Count-based
  - Word2Vec, GloVe
- Applications and use in NLP

## Week 04 - Neural Networks (intro/recap)



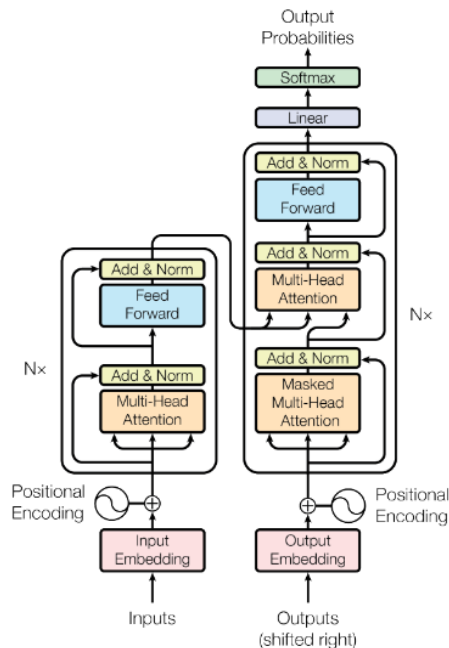
- Intro to Neural Networks
- Linear vs nonlinear models
- Backpropagation
- Language models

## Week 05 - Neural Models for NLP



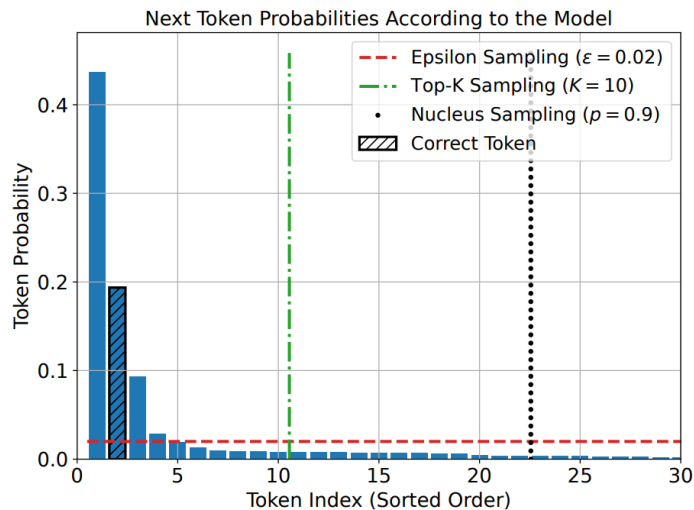
- Seq-to-seq models
- Recurrent Neural Networks (RNNs) for NLP
- Convolutional Neural Networks (CNNs) for NLP
- Applications of such networks in NLP tasks

## Week 06 - Attention and Transformer-based LMs



- Attention mechanism
- Transformer architecture
- Transformer-based models
  - BERT, GPT, PALM, ...
- Applications of such networks in NLP

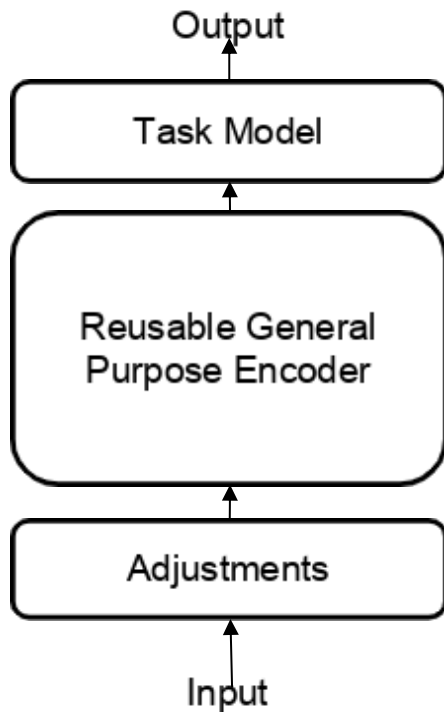
## Week 07 – Generating Text



- How to generate text from LLMs
- Sampling Methods
- Scoring Functions
- Entropy

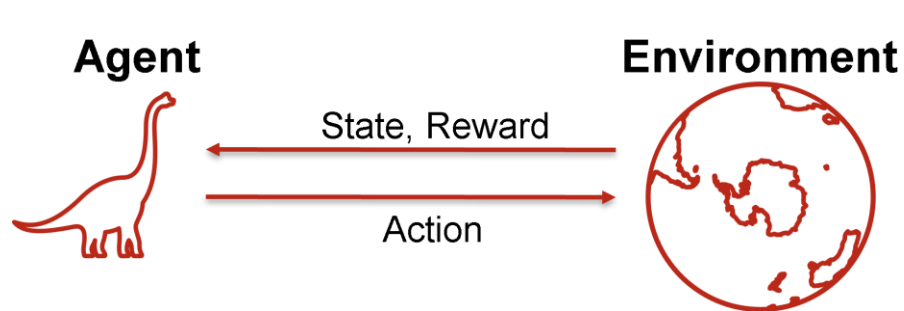


## Week 08 - Pre-training, Fine-tuning, Transfer Learning



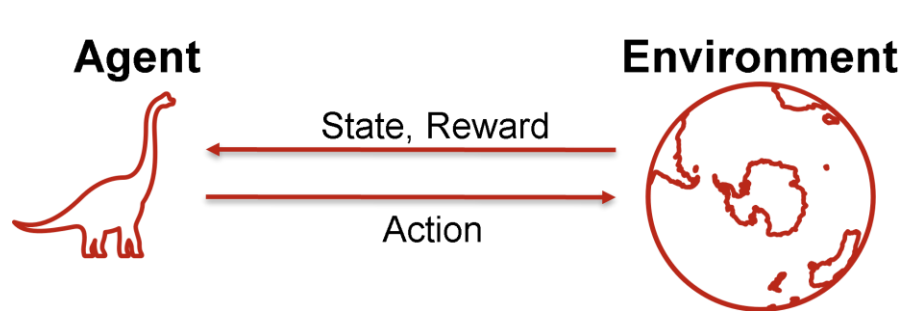
- Pre-training models on a lot of good quality data
- Fine-tuning of models for specific NLP tasks
- What is Transfer learning?
- How transfer learning is useful in NLP?
- Applications of such methods in NLP

## Week 09 - Reinforcement Learning for NLP



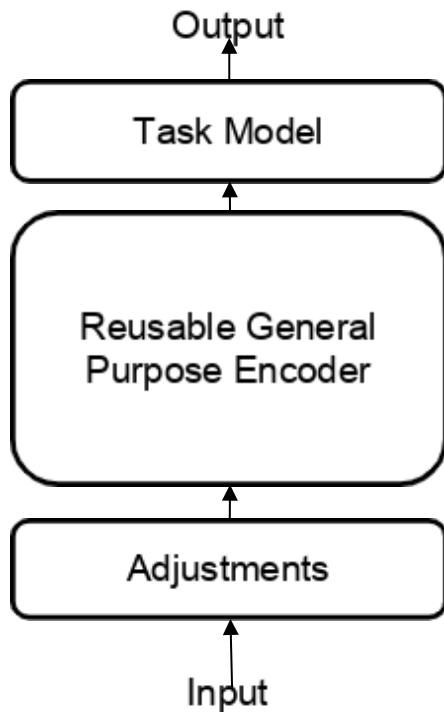
- Introduction to Reinforcement Learning
- Deep Q-Learning

## Week 10 - Reinforcement Learning for NLP



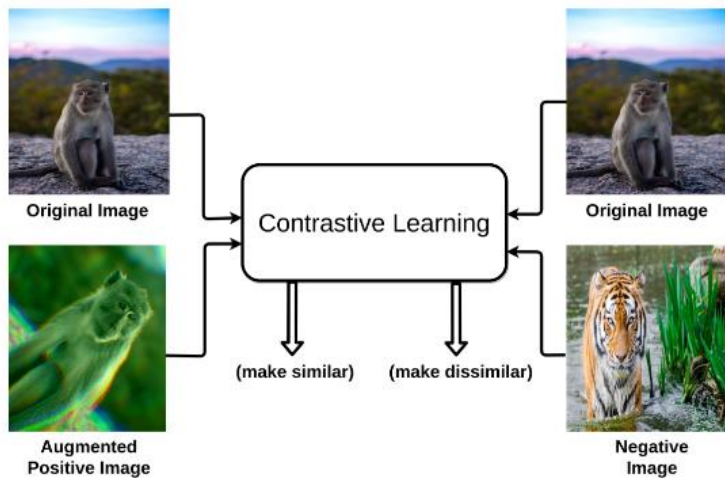
- Policy Gradient Learning
- Actor Critic
- PPO

## Week 11 - Advanced Prompting



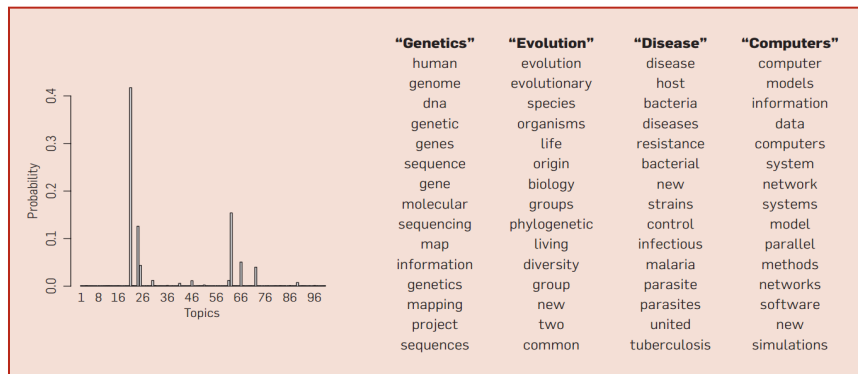
- RL-based Fine-tuning
- Prompt Engineering
- Retrieval Augmented Generation
- Prompt Learning
- LLM Teams – Inter-LLM-Communication

## Week 12 - Self-Supervised Learning/Style Transfer



- What is Self-Supervised Learning (SSL)?
- SSL in NLP
- Contrastive Learning (intro, method, uses, CLIP)
- Style Transfer

# Weeks 13 - Topic Models

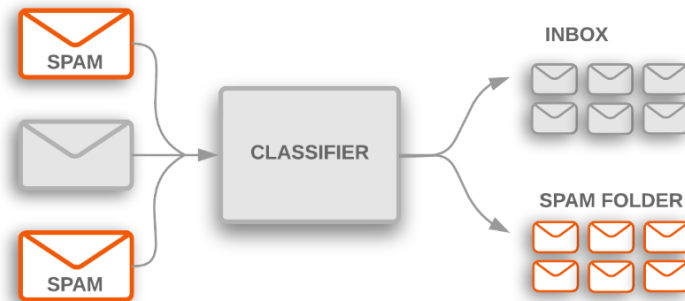


- Brief recap of topic models (LDA)
- Neural topic models
  - VAE-based topic models
  - LLM-based topic models

**Let's look at some applications**

# Text classification

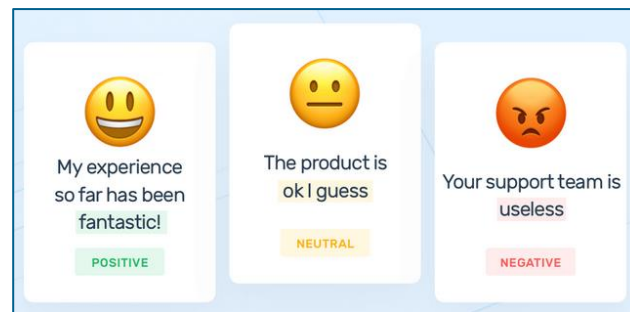
- The task of predicting the class of a given text.
- Used in any application where texts or documents need to be sorted into specific categories or classes (i.e. spam, ads, bills, friends and family, etc.)





# Sentiment Analysis

- The task of predicting the human mood, opinion, and attitude from an input text or review.
  - Predicting a movie rating from audience reviews.
  - Predicting an e-commerce product rating from customer reviews.
  - Predicting the public mood after the President's speech from social media comments.



# Named Entity Recognition (NER)

[**PER** John Smith ] , president of [**ORG** McCormik Industries ] visited his niece [**PER** Paris ] in [**LOC** Milan ] , reporters say .

BIO tags for named entity recognition

Task: Find named entities and categorize them into predefined set of categories such as location, organization, person or other.

Tag	Meaning
O	Not part of a named entity
B-PER	First word of a person name
I-PER	Continuation of a person name
B-LOC	First word of a location name
I-LOC	Continuation of a location name
B-ORG	First word of an organization name
I-ORG	Continuation of an organization name
B-MISC	First word of another kind of named entity
I-MISC	Continuation of another kind of named entity

## Text clustering

- Task: Similar to text classification. Sorting texts or documents into **non-predefined** classes by clustering similar texts into one class.
- Example:

“Don’t go out too late”

“Foxes do not hibernate”

“Don’t eat too much”

“Some bears hibernate”

“Stop using the word “hibernate””

“Don’t go out too late”

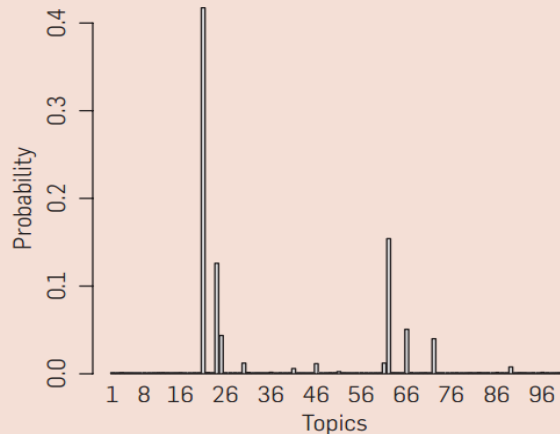
“Don’t eat too much”

“Foxes do not hibernate”

“Some bears hibernate”

“Stop using the word “hibernate””

# Topic Models



## “Genetics”

human  
genome  
dna  
genetic  
genes  
sequence  
gene  
molecular  
sequencing  
map  
information  
genetics  
mapping  
project  
sequences

## “Evolution”

evolution  
evolutionary  
species  
organisms  
life  
origin  
biology  
groups  
phylogenetic  
living  
diversity  
group  
new  
two  
common

## “Disease”

disease  
host  
bacteria  
diseases  
resistance  
bacterial  
new  
strains  
control  
infectious  
malaria  
parasite  
parasites  
united  
tuberculosis

## “Computers”

computer  
models  
information  
data  
computers  
system  
network  
systems  
model  
parallel  
methods  
networks  
software  
new  
simulations

# Text Generation

Please write a story about a researcher named Sophie who is developing an algorithm. Her goal is to stop flooding. The story ends with an umbrella.

Sophie was a researcher who was developing an algorithm to stop flooding. She had been working on this for months, and she was finally getting close to a working prototype. The last thing she needed to test was the umbrella.

She went to the park on a sunny day, and she set up her equipment. She placed the umbrella in the middle of a large puddle, and she turned on the machine. The umbrella started to spin, and the water started to rise up into the air. Sophie was amazed. It worked!

She packed up her things and went home, feeling triumphant. She had finally done it. She had developed a working prototype that could stop flooding.

## Text generated using OpenAI's GPT-3

# Text Generation

Consider the following Python 3 module:

```
```python
import numpy as np
import math
x = np.linspace(-math.pi, math.pi, 2000)
y = np.sin(x)
a = np.random.randn()
b = np.random.randn()
c = np.random.randn()
d = np.random.randn()
learning_rate = 1e-6
for t in range(2000):
    y_pred = a + b * x + c * x ** 2 + d * x ** 3
    loss = np.square(y_pred - y).sum()
    if t % 100 == 99:
        print(t, loss)
        grad_y_pred = 2.0 * (y_pred - y)
        grad_a = grad_y_pred.sum()
        grad_b = (grad_y_pred * x).sum()
        grad_c = (grad_y_pred * x ** 2).sum()
        grad_d = (grad_y_pred * x ** 3).sum()
        a -= learning_rate * grad_a
        b -= learning_rate * grad_b
        c -= learning_rate * grad_c
        d -= learning_rate * grad_d
    print(f'Result: y = {a} + {b} x + {c} x^2 + {d} x^3')
```
```

# Text Generation

Add explanatory comments to this module, clarifying the purpose of each piece of code:

```
```python
# import the numpy and math libraries
import numpy as np
import math

# create an array of x values from -pi to pi, with 2000 values
x = np.linspace(-math.pi, math.pi, 2000)

# create an array of y values using the sin function
y = np.sin(x)

# create variables a, b, c, and d, with random values
a = np.random.randn()
b = np.random.randn()
c = np.random.randn()
d = np.random.randn()

# set the learning rate
learning_rate = 1e-6

# for each value of t from 0 to 1999,
for t in range(2000):
    # predict y using the current values of a, b, c, and d
    y_pred = a + b * x + c * x ** 2 + d * x ** 3

    # calculate the loss
    loss = np.square(y_pred - y).sum()

    # print the loss every 100 iterations
    if t % 100 == 99:
        print(t, loss)
```

Explain this code in a rhyming, vaguely Shakespearean sonnet:

'''

This code imports numpy and math,  
And sets some variables for our path.  
We create an array called x  
From -pi to pi, with 2000 ticks.

Then y is created with  $\sin(x)$ ,  
And a, b, c, and d with random tricks.  
The learning rate is then set to low,  
And we begin to iterate and go.

For each value of t from 0 to 1999,  
We calculate y\_pred with a, b, c, and d.  
Then loss is found by comparing y and y\_pred,  
And printed every 100 iterations with a nod.

The gradient is found for y\_pred next,  
And then for a, b, c, and d with context.  
Then a, b, c, and d are updated with the gradient and learning rate,  
And we finish with a result that's great!

'''

## Text Generation



# Text Generation

Translate this code to use PyTorch:

```

'''python
import torch
import math

x = torch.linspace(-math.pi, math.pi, 2000)
y = torch.sin(x)

a = torch.randn(1)
b = torch.randn(1)
c = torch.randn(1)
d = torch.randn(1)

learning_rate = 1e-6

for t in range(2000):
    y_pred = a + b * x + c * x ** 2 + d * x ** 3

    loss = torch.square(y_pred - y).sum()

    if t % 100 == 99:
        print(t, loss.item())

    grad_y_pred = 2.0 * (y_pred - y)

    grad_a = grad_y_pred.sum()
    grad_b = (grad_y_pred * x).sum()
    grad_c = (grad_y_pred * x ** 2).sum()
    grad_d = (grad_y_pred * x ** 3).sum()

    a -= learning_rate * grad_a
    b -= learning_rate * grad_b
    c -= learning_rate * grad_c
    d -= learning_rate * grad_d

print(f'Result: y = {a.item()} + {b.item()} x + {c.item()} x^2 + {d.item()} x^3')
'''

```

# Text Generation

Source: PALM paper

**Input:** When I found out my grandma was in the hospital I felt a particular color. When someone cut me off in traffic I felt a different color. What is the most likely color I would see if I combined these two colors?

**Model Output:** The color that I felt when I found out my grandma was in the hospital was likely a sad color, like blue. The color that I felt when someone cut me off in traffic was likely an angry color, like red. If I combined these two colors, I would likely see purple. The answer is "purple".

**Input:** Michael is at that really famous museum in France looking at its most famous painting. However, the artist who made this painting just makes Michael think of his favorite cartoon character from his childhood. What was the country of origin of the thing that the cartoon character usually holds in his hand?

**Model Output:** The most famous painting in the Louvre is the Mona Lisa. The artist who made the Mona Lisa is Leonardo da Vinci. Leonardo da Vinci is also the name of the main character in the cartoon Teenage Mutant Ninja Turtles. Leonardo da Vinci is from Italy. The thing that Leonardo da Vinci usually holds in his hand is a katana. The country of origin of the katana is Japan. The answer is "Japan".

# Text Generation

Source: PALM paper

**Input:** Guido was writing a Python program. He keeps getting confused by variable scope and looking it up on StackOverflow. Is it possible that the Guido in the prompt refers to Guido van Rossum?

**Model Output:** Guido van Rossum is the creator of Python. He would not need to look up variable scope on StackOverflow. The answer is "no".

**Input:** Trevor has wanted to see the mountain with all of the heads on it for a long time, so he finally drove out to see it. What is the capital of the state that is directly east of the state that Trevor is currently in?

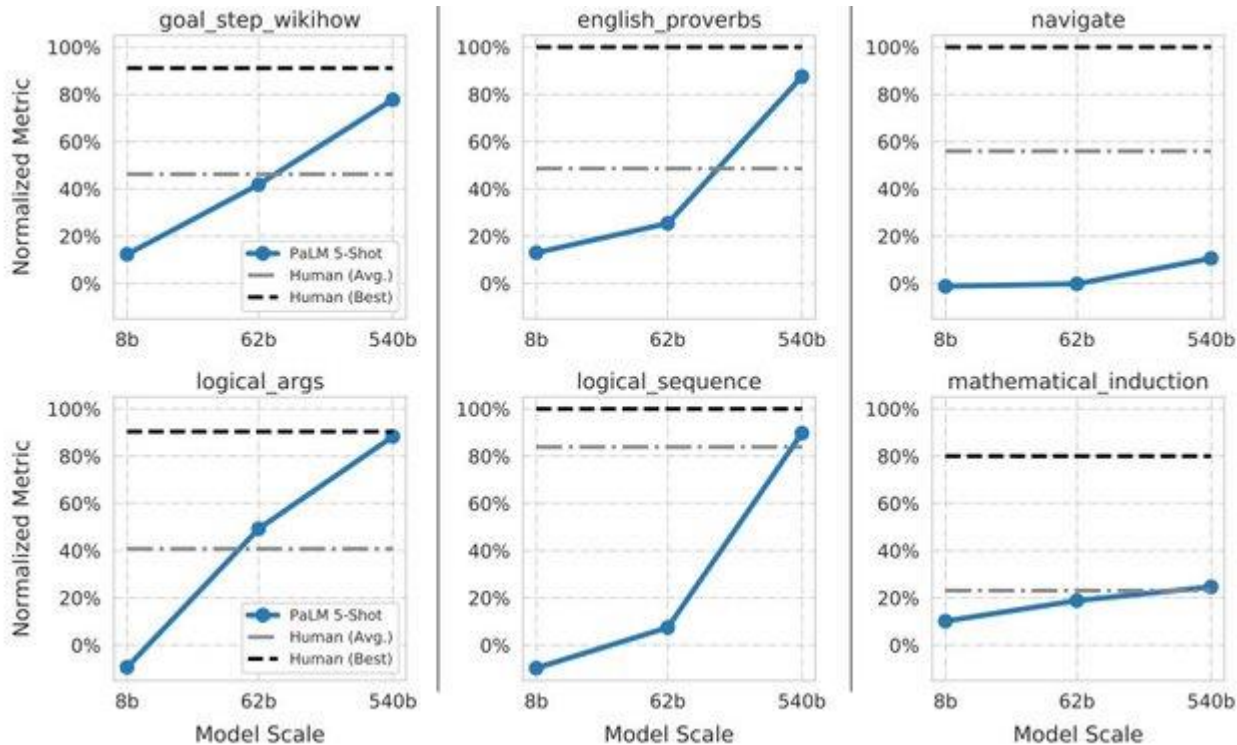
**Model Output:** The mountain with all of the heads on it is Mount Rushmore. Mount Rushmore is in South Dakota. The state directly east of South Dakota is Minnesota. The capital of Minnesota is St. Paul. The answer is "St. Paul".

**Input:** Jennifer looked out her window and sees a really cool cloud below her. She unbuckles her seatbelt and heads to the bathroom. Is Jennifer probably traveling more than 300 miles per hour relative to the earth?

**Model Output:** 300 miles per hour is about 480 km/h. This is about the speed of a commercial airplane. Clouds are usually below airplanes, so Jennifer is probably on an airplane. The answer is "yes".

# Text Generation

Source: PALM paper



# Machine Translation



Source: istockphoto.com

- One of the most widely-used NLP applications across the world.
- You probably have used online MT engines like **Google Translate** or **DeepL**.

## Text Style Transfer

I went to the store

more sad



I went to the store to buy ice-cream,  
but they didn't have any

The tree is dying  
because no one  
watered it

more melodramatic



The young tree, just starting to grow,  
hung it's head in sorrow at the neglect  
it had experienced

I first met Dean not long  
after my wife and I split  
up.

pirate style



I first met Dean not long after me' old  
lady kicked me to the curb, matey

## Dialog systems

- From ELIZA, Google Assistant, and Siri to ChatGPT, chatbots have come a long way over the last 60 years.
- Conversational AI systems that can interact with humans in dialog.
- *chatbots*.
- This field of NLP combines *natural language understanding* and *text generation*.

# Dialog systems

The task, in general is:

- Accept human text/speech as input. It could be a question, a normal sentence, or maybe just a simple “Hello!”.
- Understand the meaning of the text using natural language understanding.
- Predict a list of actions/replies based on the input received.
- Choose the best-ranked option.



## Question Answering

- The task of predicting answer(s) to a given question with respect to a given text passage reference.
- This can be viewed as a special case of Dialog Systems with two distinct differences:
  - there is an associated text passage provided.
  - the human input is always a question asked.
- The overall algorithm involved is similar otherwise.

# Question Answering

## ***Input passage:***

In meteorology, precipitation is any product of the condensation of atmospheric water vapour that falls under gravity.

***Question:*** What causes precipitation to fall?

***Prediction:*** Gravity

Example from the Stanford SQuAD question answering dataset.

# Summarization

- The task of generating a (short) summary from a given text.
- The aim is:
  - to generate a summary short and concise enough for the reader to read it.
  - not to miss out any important information.

**Source Text:** Peter and Elizabeth took a taxi to attend the night party in the city.

While in the party, Elizabeth collapsed and was rushed to the hospital.

**Summary:** Elizabeth was hospitalized after attending a party with Peter.

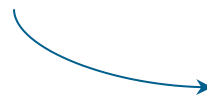


Source: floydhub.com

# Summarization

The neural network model should be able to:

- Identify and extract important terms from the input text.



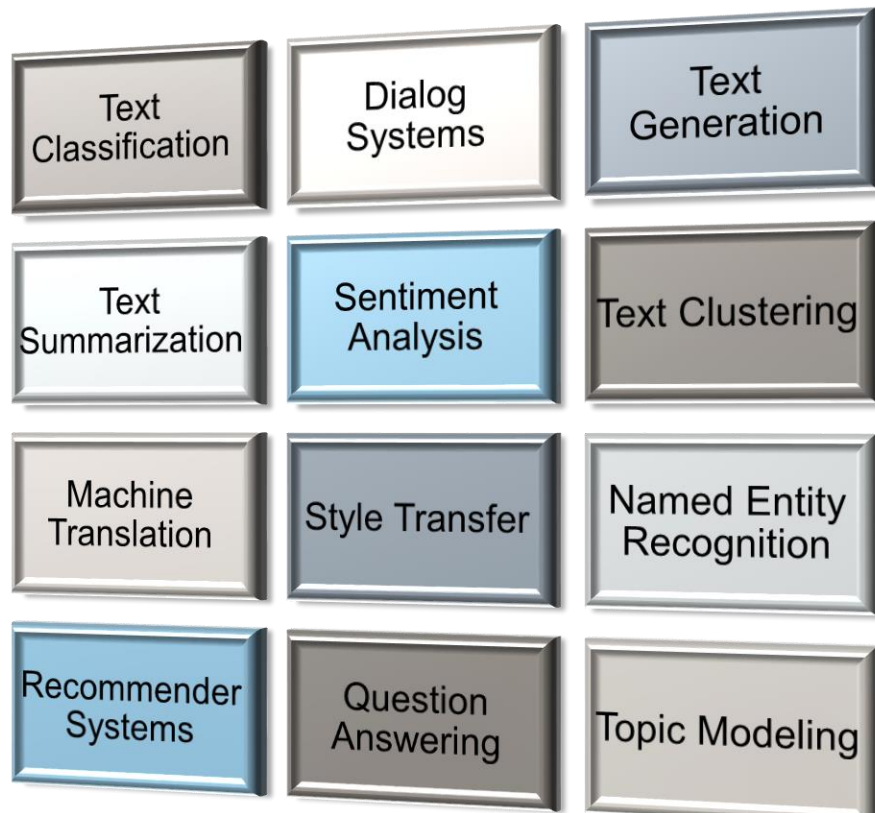
Natural language  
understanding

- Create a short and informative summary using the extracted terms.

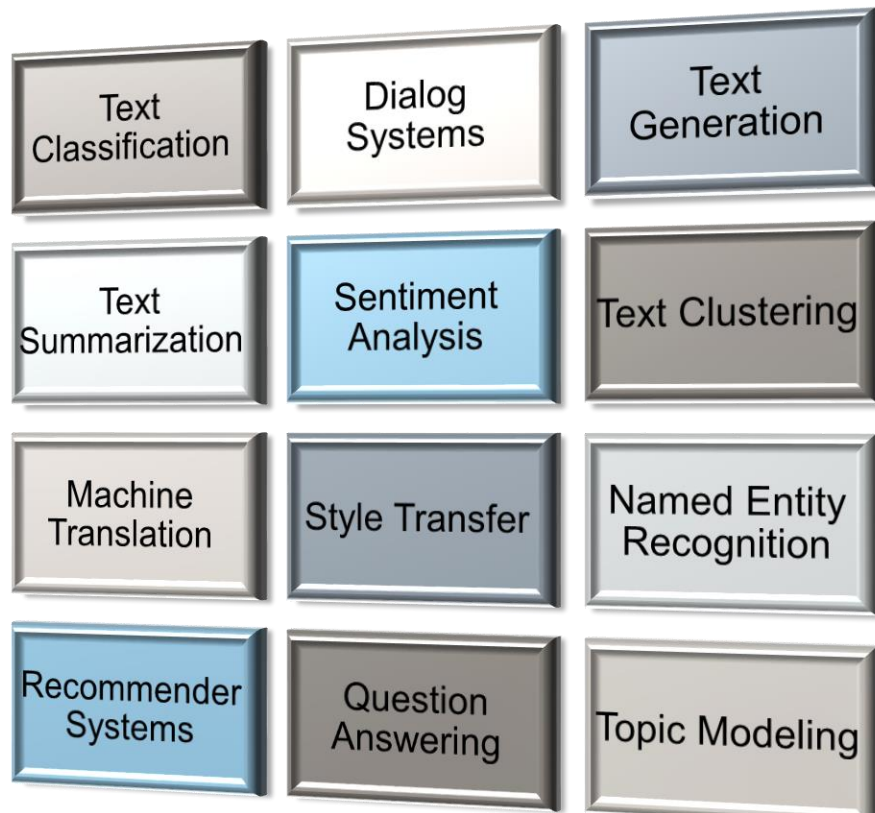


Text  
Generation

## Tasks for AI on Texts

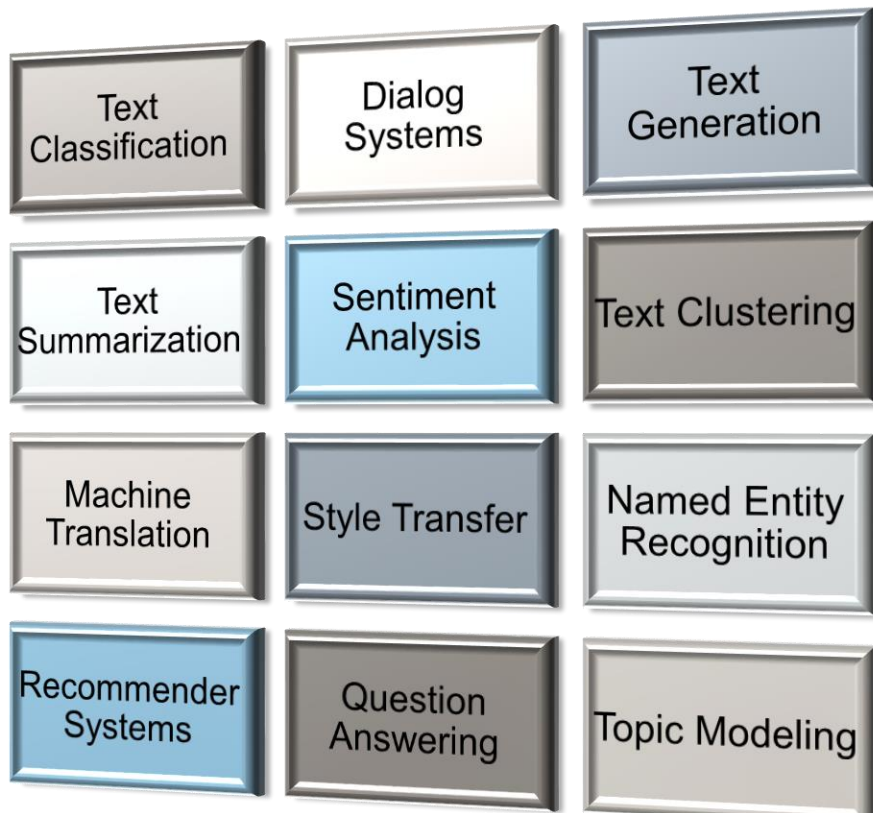


## Tasks for AI on Texts



Is there an AI to solve all of these tasks for us?

## Tasks for AI on Texts



Neural Networks for NLP

Is there an AI to solve all of these tasks for us?



Language Models

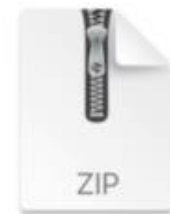
## Language Models Compress the Internet



Chunk of the internet,  
~10TB of text



6,000 GPUs for 12 days, ~\$2M  
~1e24 FLOPS



parameters.zip

~140GB file

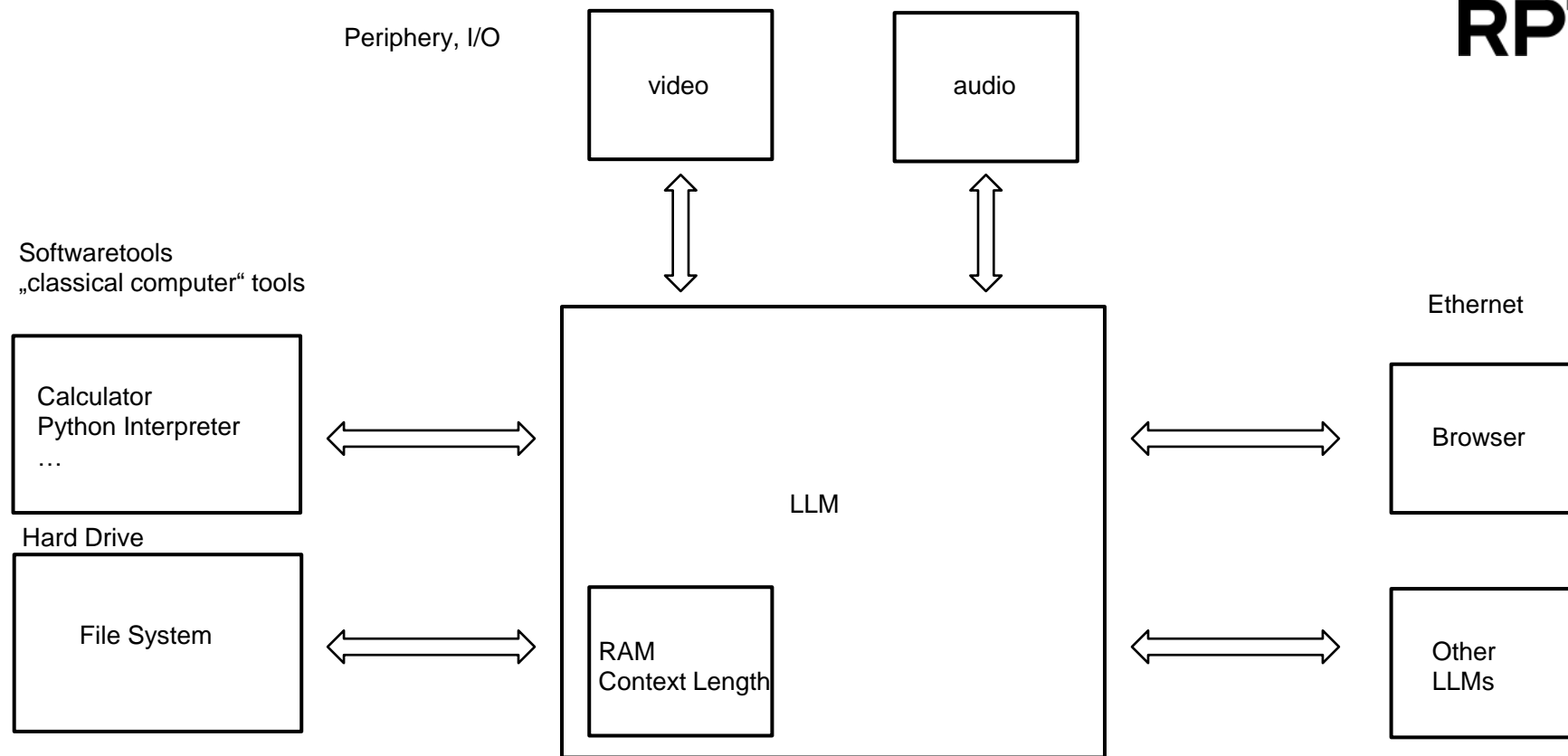
Numbers for Llama 2 70B

Source:  
Andrej Karphathy



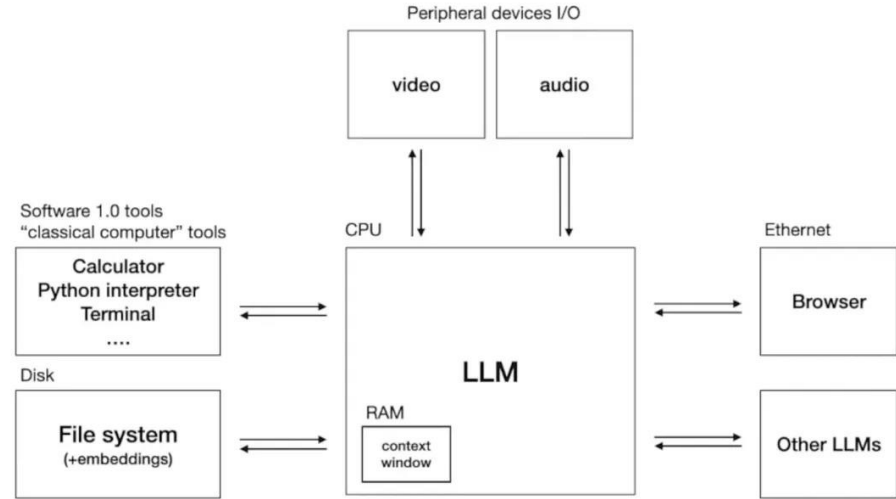
## Language Models „Dream“ Internet Documents

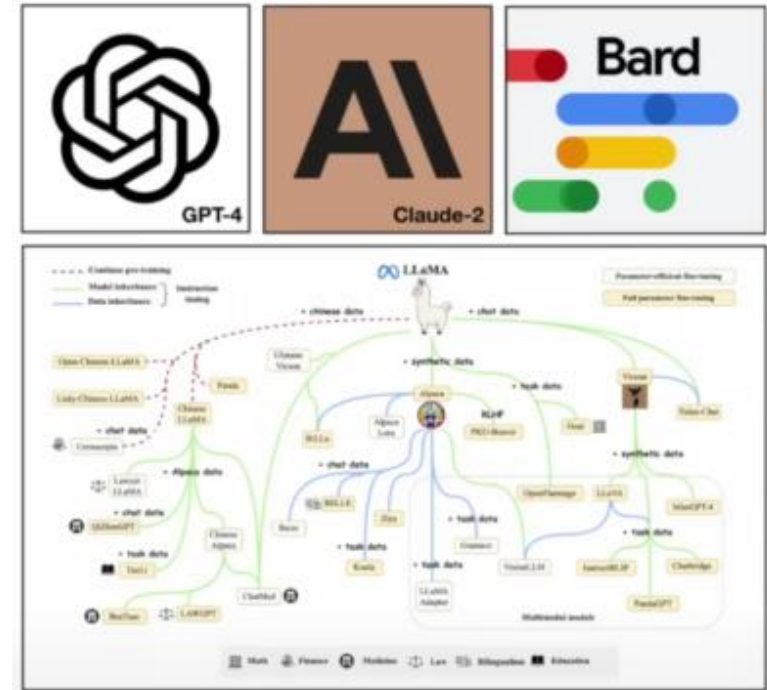
<pre>package lecture06;  import java.util.*;  public class Field {      public static Map&lt;String, Card&gt; cardIndex = new HashMap&lt;String, Card&gt;();      public void addCard(Card card){         this.cardIndex.put(card.getRegistrationNumber(),card);     }      public Card getCard(String registrationNumber) {         return this.cardIndex.get(registrationNumber);     }      public int getCardCount() {         return this.cardIndex.size();     }  }</pre>	<p>Hades Heroes</p> <p>Author: Maureen Fergus</p> <p>Product Details:</p> <p>ISBN: 9780545946767</p> <p>Format: Hardback</p> <p>Pages: 176</p> <p>Dims (mm): 140 × 223</p> <p>Pub Date: 01-05-16</p> <p>Pub Country: United States</p> <p>Condition: NEW</p> <p>Description:</p> <p>Work hard. Love. Make your own luck. These are the keys to success in life at school. But when a student named Hercules wanders in and challenges him</p>	<p>Blacknose dace</p> <p>The blacknose dace (<i>Rhinichthys atratulus</i>) is a small freshwater fish found in North America. They are typically found in streams, rivers, and lakes, and are widely distributed throughout the eastern and central United States and Canada</p> <p>These fish have a slender body with a black stripe running horizontally along their sides, giving them their distinctive "blacknose" appearance. They have a flattened head with a pointed snout and small, round eyes. Their color can vary from dark brown to olive green on their back and sides, with a silvery-white underside.</p> <p>Blacknose dace are omnivorous and feed on a variety of small aquatic insects, crustaceans, and plant matter. They are an important food source for larger predators such as bass,</p>
Java Code - Dream	Product Description Dream	Wikipedia - Dream



## Capabilities of Language Models

- Read and generate text
- Have more knowledge than any human on all kinds of topics
- Can search for information online
- Can use software (Calculator, Python, Mouse/Keyboard)
- Can see and generate images/videos
- Can hear and speak
- Can think and reflect on their own thoughts
- Can communicate with other language models
- Can be fine-tuned for many different specialized tasks
- ...





## Conclusion

- NLP has many applications
- Language models are becoming more and more important

# Acknowledgements

- Stanford Course:  
<https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1184/syllabus.html>
- Weichen Li
- Feibai Huang
- Sourav Dutta