# Lec 1 Introduction

## # NLP

What is NLP ? Make computers understand & generate human languages

For...
- Human - computer } interaction
- Human - human
- Understand languages ( syntax, etc. )

Applications
- Answer questions
- Translate
- Aid scientific research
  → comp social sci corpus analysis paper
  agency / power analysis of characters in films

← Models can make mistake in all of these

Class
- Build NLP systems
- When do these systems fail
- How to improve

## # NLP System Overview

General framework :

input $X \mapsto$ output $Y$

X or Y involves language

Ex.
- lang1 ↦ lang2
- question + choices ↦ answer
- query + documents ↦ relevant documents
- text ↦ label
- image ↦ caption

Methods
- Rule based
- Prompting
  prompt a lang model, wo training
- Fine - tuning

Data
- Rules / prompt → maybe no data
  use intuition
  spot-check to fix rules / prompt
- Rules / prompt with evaluation → dev set + test set
  200 - 2000 examples
- Fine - tuning → the more the better
  in general:
  performance ↑ linear
  when data ↑ quadratic

## # Make rule-based sentiment analysis * Bad idea

Task: product review ↦ {positive, neutral, negative}

1. Feature extraction $\vec{h} = f(x)$
2. Score calculations $s = w \cdot \vec{h}$ (binary) $\vec{s} = W\vec{h}$ (multiclass, $W$ ← weight matrix)
3. Decision function $\hat{y} = \text{decide}(s)$

good words = [ love, good, ... ]
bad words = [ hate, sad, ... ]
} designing this is complicated
bias = 1
score = 1.0 * count good words
+ 1.0 * count bad words
+ 0.5 * bias ← not that simple

$$\hat{y} = \begin{cases} \text{positive} & \text{if score} > 0 \\ \text{neutral} & : \quad = 0 \\ \text{negative} & : \quad < 0 \end{cases}$$

To improve : comprehensive analysis

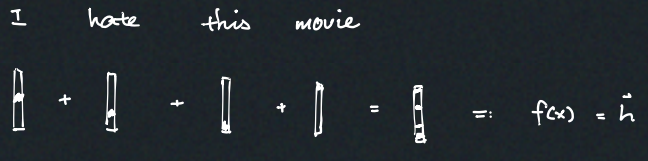
Issues:
- low-frequency words ...
  ↳ manually add them
  ↳ sentiment dictionary
  ↳ use root form of words
  i.e. morphological analysis
- negation "not bad" ↦ neutral / positive ?
  ↳ syntactic analysis to see what's negated
- Metaphor
- Someone sending reviews in another language
  ↳ learn Japanese (?)

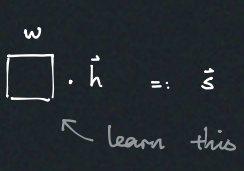
## # Machine learning approach — BOW

→ Learn feature extractor or weight function

Fixed feature extractor

I hate this movie




Structured perception training algorithm

```
for (x, y) in train set
    if y = neutral : skip
    h = f(x)
    ŷ = predict(h)
    if ŷ ≠ y:
        upweight / downweight weight matrix
```

BOW problems
- Conjugation
- Word similarity
- Combined feature
  ↑ love
  don't hate
  don't love
  ↓ hate
- Sentence structure
  "but"

## # Neural network

Theoretically NN can model & solve any problem

## # Assignments

1. Build LLaMa
2. NLP task from scratch for specified task
   ↳ collect data
   ↳ modelling
   ↳ evaluation
3. Survey + re-implementation
   ↳ lit review
   ↳ reimplement an NLP paper
4. New research
   ↳ improve performance
   ↳ apply technique to new task