

# Natural Language Processing

## Module 1: Introduction

# Chiraag Lala

Teaching Fellow at Imperial

PhD in Computer Science (NLP)

MSc in Artificial Intelligence

BS in Mathematics

Indian National Mathematics Olympiad

I teach mathematics required for Computing, AI, NLP.

Supervise NLP projects

Year 3 and 4 coordinator

**I am the module leader, your first point of contact**

# Lecturers



**Marek Rei**



**Joe Stacey**



**Nihir Vedd**



**Nuri Cingillioglu**

# Goals of the course

- Introduce different tasks and challenges that are important for NLP.
- Provide an overview of significant NLP methods and models, starting with classical approaches all the way to state-of-the-art models.
- Give you practical experience with designing, implementing and training different NLP systems.

# Timeline



Week	Start Date	Lecture	Lecturer
2	15 Jan	Introduction + Word representations	Nihir
3	22 Jan	Classification	Joe
4	29 Jan	Language Modelling	Joe
5	5 Feb	Machine Translation	<i>Nihir</i>
6	12 Feb	Transformers	<i>Nihir</i>
7	19 Feb	Pre-trained models	Marek
8	26 Feb	Structured Prediction	<i>Nuri</i>
9	4 Mar	Revision + additional stuff	<i>All</i>
10	11 Mar	No lectures	-
11	18 Mar	Exam week	

# Today

1. Structure of the course and logistics
2. Dealing with natural language
3. Linguistic levels of language understanding
4. Quick history of NLP
5. Neural network refresher (self-study)
6. Applications of NLP

Then Nihir will take over with 'Word representation'

# Structure of the course

Course runs for 7 weeks + 1 revision week

Each week: 2+1 hours of lectures and 1 tutorial/lab session

All face-to-face. Lectures will be recorded and available on Panopto.

Lab sessions:

- For you to work on lab exercises and coursework.
- Lab exercises are self-explanatory - do them at your own pace.
- Teaching Assistants will be at the lab sessions to answer questions as needed.

# Structure of the course

## Grading:

- Project: 30%
- Exam: 70%

## Coursework/project: Text classification model

- Released: 29 January
- Submission deadline: 5 March (19:00)
- Designing and implementing a working system, writing a report
- Groups of up to 3 students
- Strict about plagiarism!

Need to form groups on Scientia first.

Group formation deadline: 26 Jan

All members of a group will get same marks.

# Course links

EdStem

<https://edstem.org/us/courses/46846/discussion/>

General information, Links, Discussion, Questions and Announcements

**Main portal for all information and communication regarding this module**

Scientia

<https://scientia.doc.ic.ac.uk/2324/modules/70016/materials>

Slides and materials

Panopto

<https://imperial.cloud.panopto.eu/Panopto/Pages/Sessions>List.aspx?folderID=efa79f41-23eb-4986-a0f0-b02101297ebd>

Lecture Recordings

# Colaboratory (Google Colab)

## Colab and Jupyter notebooks

- Provides GPU access for short processes
- Will be used for lab sessions
- Can also be used for the coursework
- Intro: <https://colab.research.google.com/notebooks/welcome.ipynb>

## PyTorch

- A deep learning framework for Python
- Will be used for lab exercises
- Recommended also for coursework
- Tutorial:  
[https://pytorch.org/tutorials/beginner/deep\\_learning\\_60min\\_blitz.html](https://pytorch.org/tutorials/beginner/deep_learning_60min_blitz.html)

# Materials

Selected NLP books:

- Speech and Language Processing. Dan Jurafsky and James H. Martin.  
<https://web.stanford.edu/~jurafsky/slp3/>
- A Primer on Neural Network Models for Natural Language Processing. Yoav Goldberg.  
<https://u.cs.biu.ac.il/~yogo/nlp.pdf>
- Natural Language Processing. Jacob Eisenstein.  
<https://github.com/jacobeisenstein/gt-nlp-class/blob/master/notes/eisenstein-nlp-notes.pdf>
- Natural Language Processing with Transformers. Lewis Tunstall, Leandro von Werra and Thomas Wolf.

These are optional additional materials, not a requirement.

# Materials

Selected ML books:

- Artificial Intelligence: a Modern Approach. (2009) Stuart Russell & Peter Norvig
- Machine Learning. (1997) Tom Mitchell
- Neural Networks and Deep Learning. Michael A. Nielsen.  
<http://neuralnetworksanddeeplearning.com/>
- Introduction to Deep Learning. Eugene Charniak.

For finding state-of-the-art methods on established NLP tasks:

- <https://paperswithcode.com/sota>
- <https://github.com/sebastianruder/NLP-progress>

Acknowledgements: This course is based on lecture materials from Prof Lucia Specia and Dr Marek Rei and the many PhD students.

# Dealing with natural language

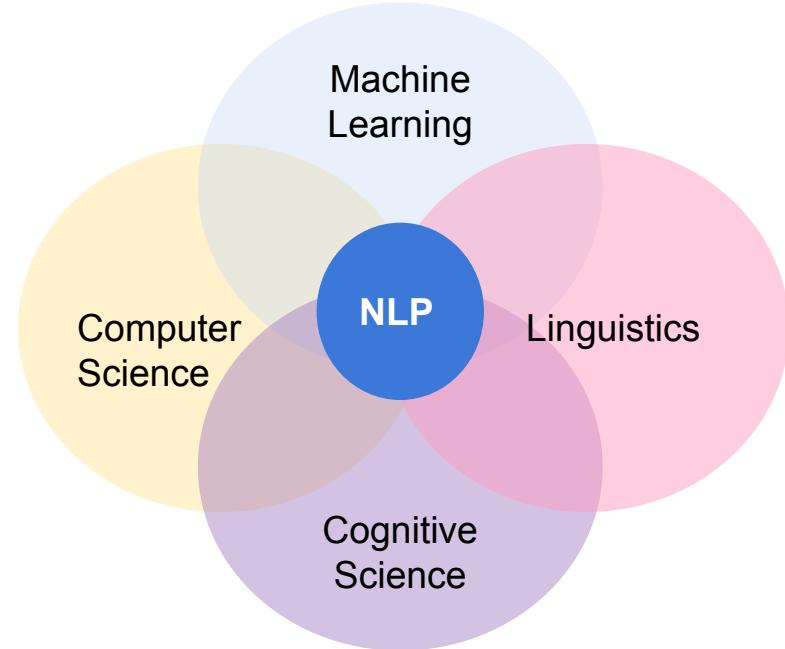
# Natural language processing

*NLP* is the processing of natural language by computers for a task

*Natural language* is a language that has developed naturally in use (i.e. not computer code).

Ultimate goal: systems to 'understand' NL, execute requested tasks & produce NL

Focus on written language (not speech, not OCR)



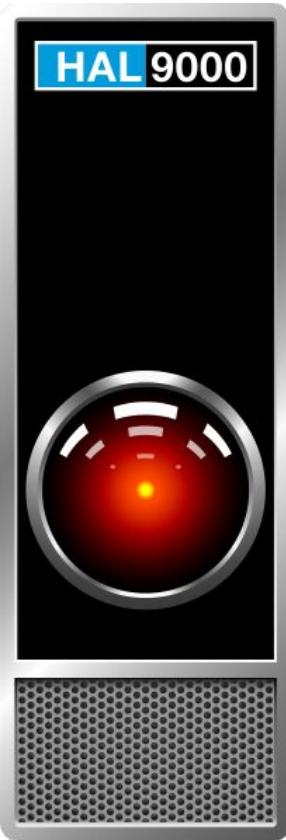
# AI and natural language



HAL: Look Dave, I  
can see you're really  
upset about this. I  
honestly think you  
ought to sit down  
calmly, take a stress  
pill, and think things  
over."

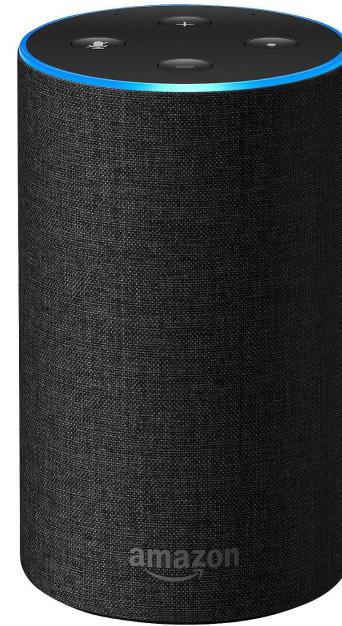
*2001: A Space Odyssey*  
(1968)

# AI and natural language



HAL: Look Dave, I can see you're really upset about this. I honestly think you ought to sit down calmly, take a stress pill, and think things over."

*2001: A Space Odyssey  
(1968)*



Alexa, what is the traffic like?

# How well can computers ‘understand’?



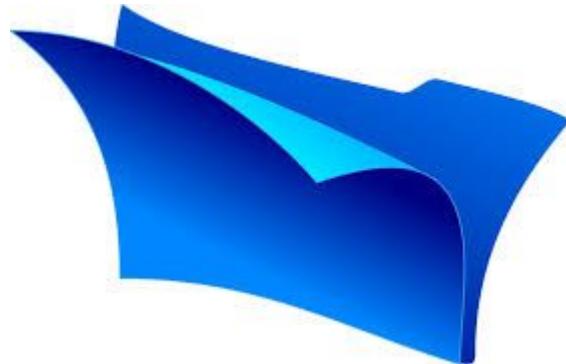
# How well can computers understand?



# Natural language is complex

Ambiguity at the word level

*Can you bring me that file?*



# Natural language is complex

Syntactic ambiguity (prepositional phrase attachment ambiguity)

*I saw the boy **with** a telescope.*



# Natural language is complex

Semantic ambiguity

*I haven't slept **for 10 days**.*



Because that  
would have  
been too long.

# Natural language is complex

Semantic ambiguity

*The **rabbit** is ready for lunch.*



# Natural language is complex

## Referential ambiguity

*We gave the monkeys bananas because **they** were more than ready to eat.*



# Natural language is complex

Non-literal meaning

*Call me a cab, it's raining cats and dogs*



# Linguistic levels of language understanding

# Subtasks of language understanding

- Lexicon
  - Words: segmentation, normalisation, morphology
- Syntax
  - Sentence structure
- Semantics
  - Meaning of words and sentences
- Discourse
  - Meaning of a text (relationship between sentences)
- Pragmatics
  - Intentions, commands

Previously: A separate system for dealing with each level

Currently: Most of this learned with a single model end-to-end

# Lexicon - morphological analysis

What is a ‘word’ and what is it composed of?

- Word segmentation (tokenisation, decompounding, etc)
  - hi, → **hi**,
  - Run 10,000 meters → **Run 10,000 meters**
  - Wienerschnitzel → **Wiener schnitzel** (cutlet Vienna style)
- Word normalisation (capitalisation, acronyms, spelling variants)
  - Cat → **cat**, Turkey → **turkey**
  - USA and U.S.A and U S A → **USA**
  - chequebook and **cheque book** → **cheque book**

# Lexicon - morphological analysis

What is a ‘word’ and what is it composed of?

- Lemmatisation (reduce to base form = valid word)
  - Ate, eating → eat
  - girls → girl
  - connected → connect
  - worried, worries → worry
- Stemming (reduce to root = not always valid word)
  - connected, connecting, connection, connections → connect
  - worried, worryingly, worries, worrying → worr
- Byte-pair encoding (BPE) and *wordpieces*
  - Data-driven methods for breaking words apart, used by latest neural models. Will be covered in more detail.

# Lexicon - morphological analysis

What is a ‘word’ and what is it composed of?

- Part-of-speech tagging (recognise category of word): verb, noun, adverb, adjective, determiner, preposition...
  - girl = noun
  - play = verb/noun
  - happy = adjective
  - happily = adverb
  - the = determiner
  - of = preposition
  - they = pronoun

# Lexicon - morphological analysis

What is a ‘word’ and what is it composed of?

- Morphological analysis (recognise/generate word variants):
  - Number, gender, tense:
    - boys = plural form of ‘boy’
    - heureuse = feminine form of ‘heureux’
    - plays = third form of verb ‘play’
  - Affixes (prefixes/suffixes):
    - unhappy
    - happily
    - unhappiness
    - John’s

# Structure, syntax - Parsing

How are words put together?

- Ruled by the language's **grammar**, e.g. phrase-structure:

$S \rightarrow NP\ VP$

$NP \rightarrow Det\ N$

$VP \rightarrow V\ NP$

$VP \rightarrow V$

$VP \rightarrow V\ PP$

$PP \rightarrow P\ NP$

# Structure, syntax - Parsing

How are words put together?

- Ruled by the language's **grammar**, e.g. phrase-structure:

$S \rightarrow NP\ VP$

$NP \rightarrow Det\ N$

$VP \rightarrow V\ NP$

$VP \rightarrow V$

$VP \rightarrow V\ PP$

$PP \rightarrow P\ NP$

Sentence  $\rightarrow$  Noun-phrase Verb-phrase

Noun-phrase  $\rightarrow$  Determiner Noun

Verb-phrase  $\rightarrow$  Verb Noun-phrase

Verb-phrase  $\rightarrow$  Verb

Verb-phrase  $\rightarrow$  Verb Prepositional-phrase

Prepositional-phrase  $\rightarrow$  Preposition Noun-phrase

# Structure, syntax - Parsing

How are words put together?

- Ruled by the language's **grammar**, e.g. phrase-structure:

$S \rightarrow NP\ VP$

$NP \rightarrow Det\ N$

$VP \rightarrow V\ NP$

$VP \rightarrow V$

$VP \rightarrow V\ PP$

$PP \rightarrow P\ NP$

and **lexicon**

$N \rightarrow \text{puppy, garden}$

$\text{Det} \rightarrow \text{the}$

$V \rightarrow \text{played}$

$P \rightarrow \text{in}$

# Structure, syntax - Parsing

How are words put together?

$S \rightarrow NP\ VP$

$NP \rightarrow Det\ N$

$VP \rightarrow V\ NP$

$VP \rightarrow V$

$VP \rightarrow V\ PP$

$PP \rightarrow P\ NP$

$N \rightarrow \text{puppy, garden}$

$\text{Det} \rightarrow \text{the}$

$V \rightarrow \text{played}$

$P \rightarrow \text{in}$

**the puppy played in the garden**

# Structure, syntax - Parsing

How are words put together?

$S \rightarrow NP\ VP$

$NP \rightarrow Det\ N$

$VP \rightarrow V\ NP$

$VP \rightarrow V$

$VP \rightarrow V\ PP$

$PP \rightarrow P\ NP$

$N \rightarrow \text{puppy, garden}$

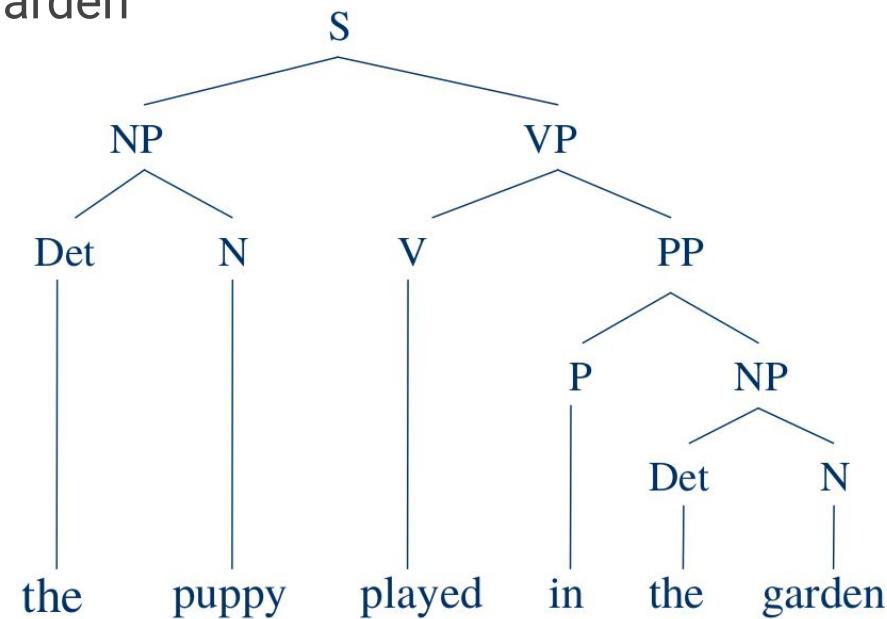
$\text{Det} \rightarrow \text{the}$

$V \rightarrow \text{played}$

$P \rightarrow \text{in}$



**the puppy played in the garden**



# Structure, syntax - Parsing

How are words put together?

$S \rightarrow NP\ VP$

$NP \rightarrow Det\ N$

$VP \rightarrow V\ NP$

$VP \rightarrow V$

$VP \rightarrow V\ PP$

$PP \rightarrow P\ NP$

$N \rightarrow \text{puppy, garden}$

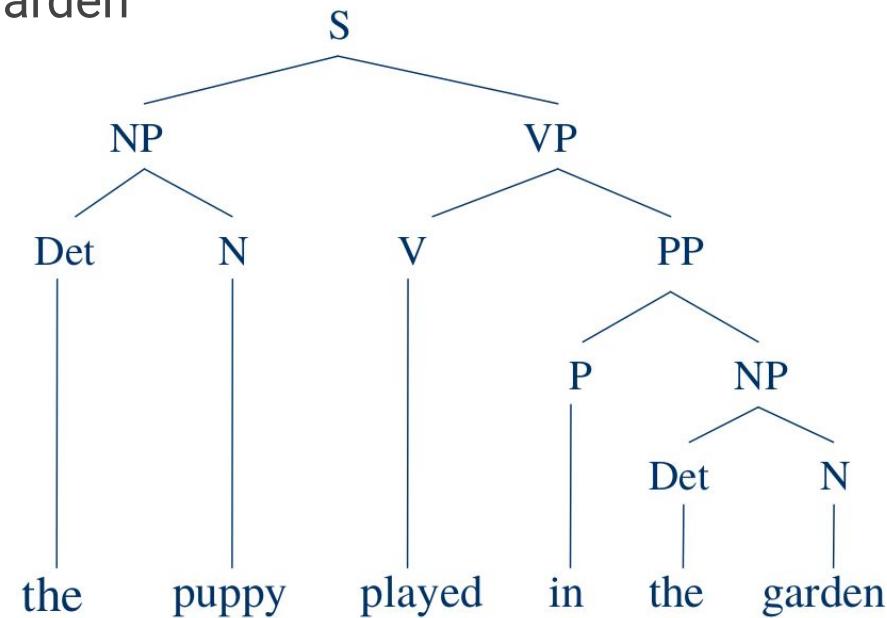
$\text{Det} \rightarrow \text{the}$

$V \rightarrow \text{played}$

$P \rightarrow \text{in}$

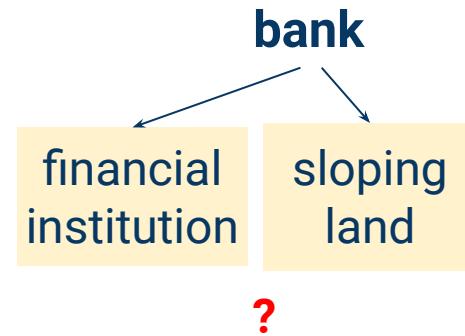
This is a sentence that belongs to /  
is accepted by the language

**the puppy played in the garden**



# Semantics

Meaning of words:



# Semantics

Meaning of words in context:

We sat on a bench at the river **bank**.



sloping  
land

**Word Sense Disambiguation:** given a word and its context, assign the correct meaning given a set of candidates

# Semantics

Understanding who did what to whom, when, where, how and why?

**Compositional** meaning. Composing the meaning of the sentence, based on the meaning of the words and the structure of the sentence.

The dog chased the man = The man was chased by the dog

The dog bit the man != The man bit the dog

**Semantic role labelling:** assign semantic roles to words in sentence; fail a sentence if that is not possible

# Discourse

How do sentences relate to each other?

## References & relationships within and across sentences

- Hurricane Hugo destroyed 20,000 houses. **The disaster** has been the most costly in the estate's history.
- Go home now. I can't do **that**.
- I bought a new computer and a new mouse. **It** is so fast!

**Coreference resolution:** Mapping different mentions of the same entity together.

# Pragmatics

What is the intent of the text? How to react to it?

- Do you have the time?

YES!



- Can you close the window?
- It's cold in here
- The gas bill arrived yesterday

# Quick history of NLP (a.k.a. Computational Linguistics)

# History

## 1950's: the beginning...

- Foundational work: automata theory, information theory
- Machine translation funded by military
- Toy word models

The IBM-Georgetown experiment: Jan 7, 1954 the first public demonstration of a Russian-English MT system: within a few seconds a computer translated sentences into English using a lexicon of 250 words and six grammar rules.

This project began in 1951 and was the first non-numerical application of a digital computer.

<http://www.hutchinsweb.me.uk/>



Fig. 2: Hurd, Dostert and Watson at the demonstration

# History

## 1960's-1970's: NLP winter

- Rule-based models
  - Hand-crafted **rules** to model linguistic intuitions
- Disappointing results for machine translation
- Work shifts to deeper models, syntax but toy domains

## 1980's-1990's: the empirical revolution

- Corpus-based methods introduced
- Deep analysis traded for robust, simple approximations

# History

## 2000's: better statistical and machine learning methods

- Models increasingly merge linguistically sophisticated representations with statistical and data-driven methods

## 2014+: the neural revolution

- Models with large numbers of parameters optimized on big datasets
- Deep analysis traded for simple yet robust and better performing approximations

## 2018+: the era of pre-trained language models

- Language models trained on large amounts of data show excellent language generalisation abilities
- Essentially all NLP models start incorporating some form of pre-trained language models

# Why machine learning

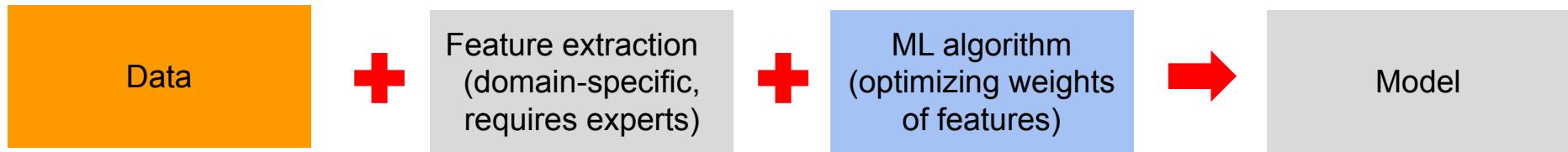
- Creation and maintenance of linguistic rules often infeasible or impractical
  - Number of rules
  - Contradiction in rules
  - Cost and need for experts
- Learning functions from data instead creating rules based on intuition
- Optimizing weights and balances automatically instead of tuning them manually
- Data/examples can be abundant for certain applications and languages, e.g. human translations, movie reviews, ...

# Why deep learning

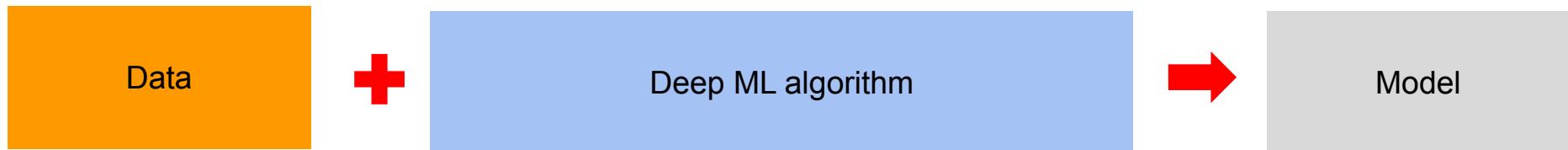
- Provides very flexible, learnable framework for representing information from many different modalities (linguistic, visual, numerical, etc.).
- Optimize models end-to-end, instead of having a pipeline of models
- Can learn from **supervised** and **unsupervised** data
- State-of-the-art performance

# Why deep learning

Feature engineering:



Feature learning (the deep learning approach):



# Neural network refresher

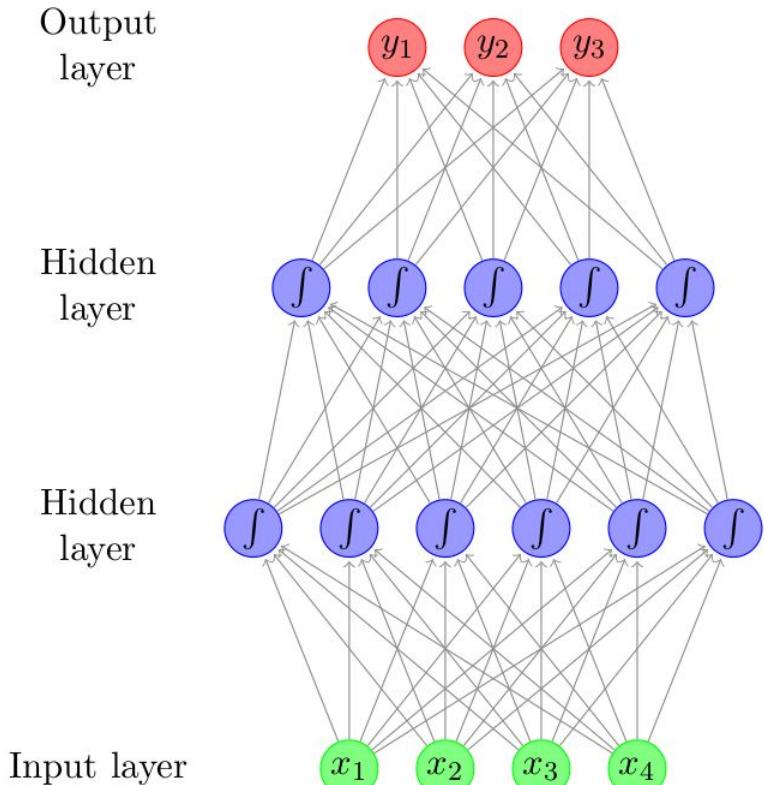
# Feed-forward networks

A neural network consists of

1. An **input layer**
2. An **output layer**
3. A number of **hidden layers** between them

A layer consists of a number of **neurons**.

Each neuron in a layer takes as input the outputs from all the neurons in the previous layer (**fully connected**).

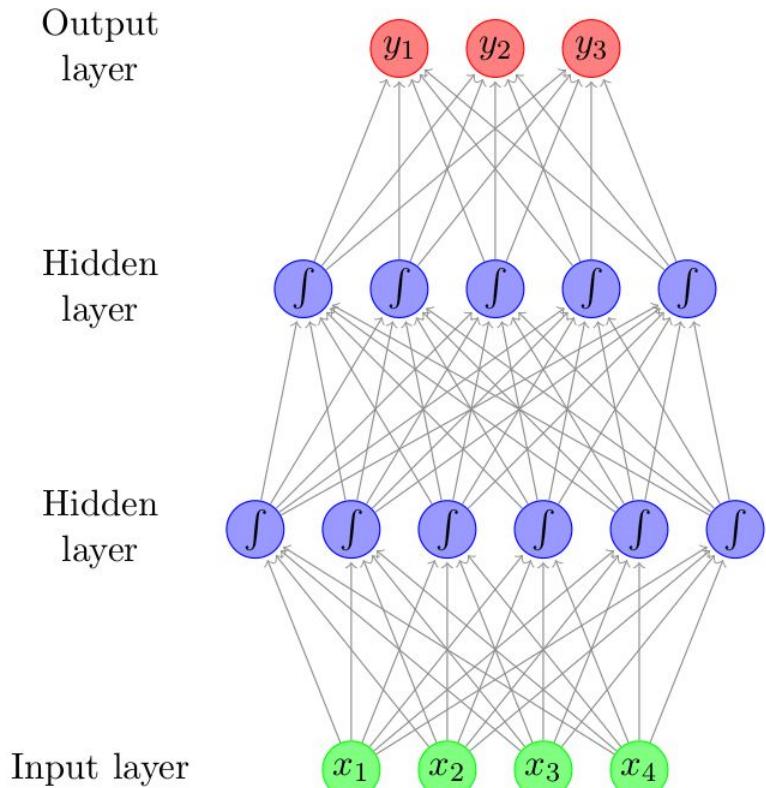


# Feed-forward networks

Each connection between the neurons has a weight.

A neuron is a computational unit with scalar inputs & outputs. It:

1. multiplies its inputs by their corresponding weights
2. then sums them up
3. applies a non-linear function to this sum
4. outputs the result



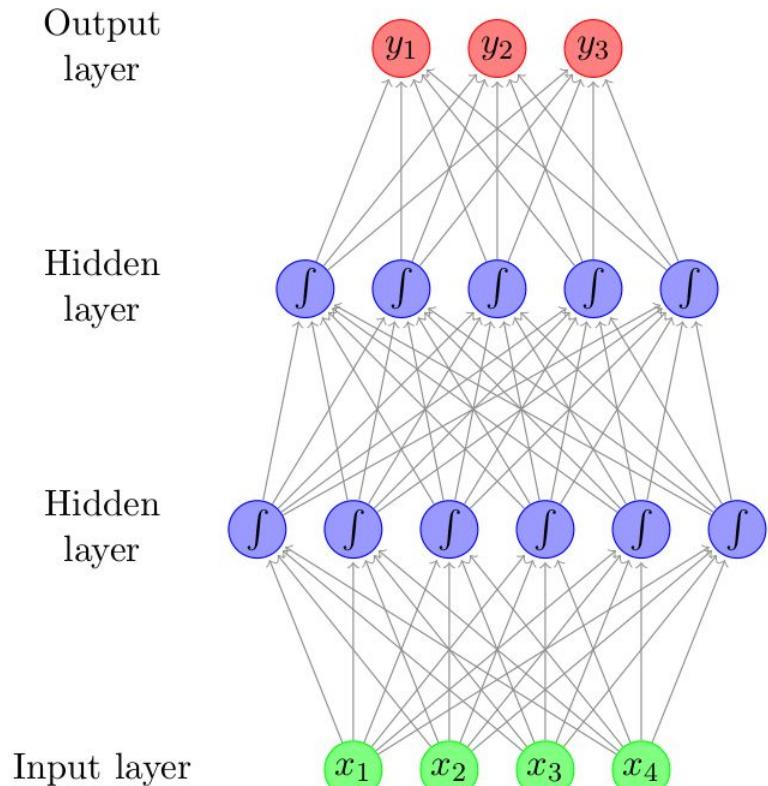
# Feed-forward networks

The output of each layer can be thought of as a vector.

In this example:

1. the input to this network is a vector of length 4 ( $x$ )
2. The first hidden layer produces a vector of length 6 ( $h_1$ )
3. The output of this network is a vector of length 3 ( $y$ )

*Affine layer* = fully connected layer



# Feed-forward networks

To calculate the first hidden layer based on the input  $x$ :

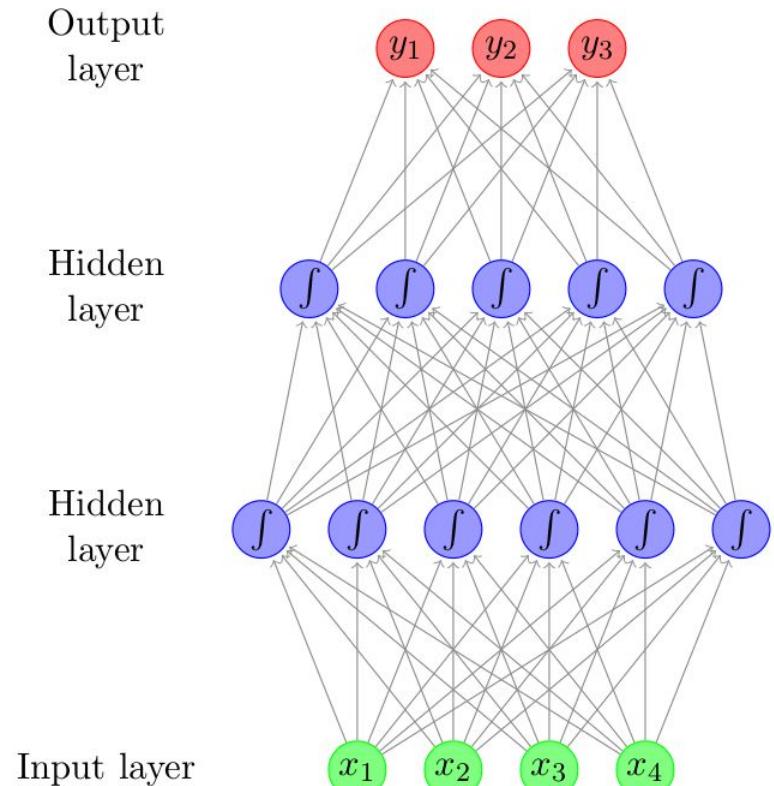
$$h_1 = g(xW_1 + b_1)$$

$x$  is a vector of size 4

$W_1$  is a matrix of size (4, 6), containing all the weights between the input layer and the first hidden layer

$b_1$  is a vector of bias parameters

$g()$  is a non-linear activation function



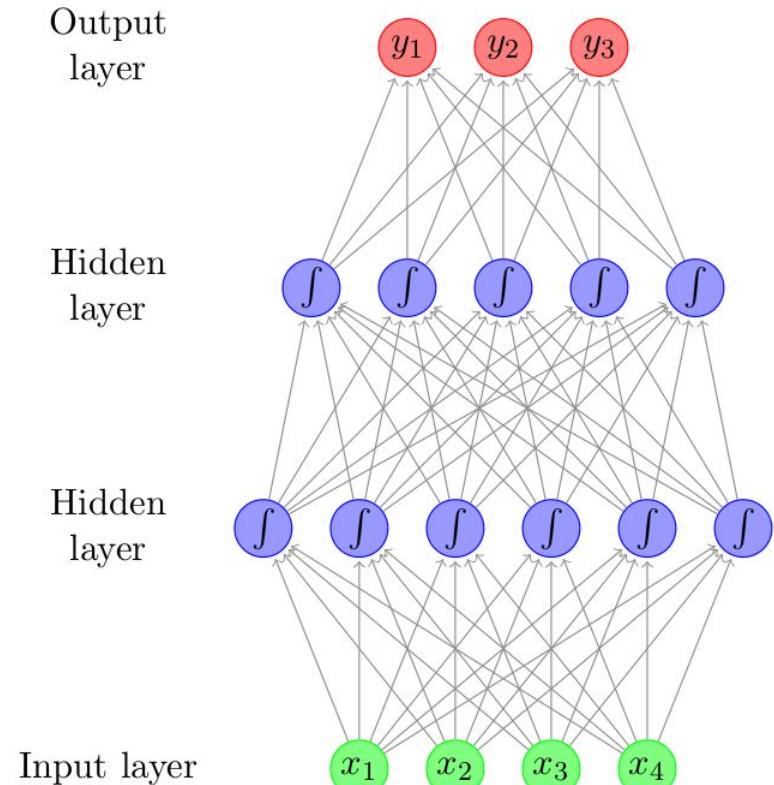
# Feed-forward networks

To calculate the output of the second hidden layer:

$$h_2 = g(h_1 W_2 + b_2)$$

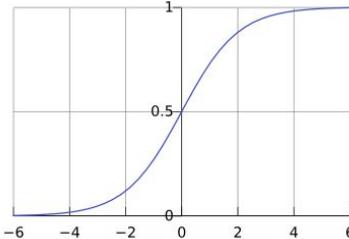
$$h_2 = g(g(xW_1 + b_1)W_2 + b_2)$$

We can stack as many hidden layers as we want.



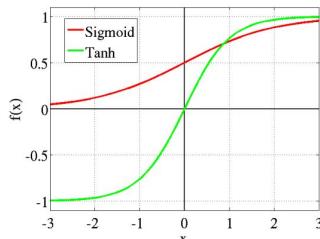
# Non-linear activation functions

## Sigmoid/logistic activation



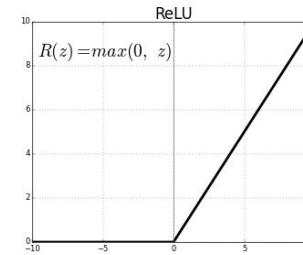
$$f(x) = \sigma(x) = \frac{1}{1 + e^{-x}}$$

## Tanh activation



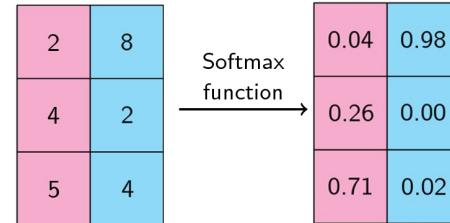
$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

## ReLU activation



$$f(x) = \text{ReLU}(x) = \begin{cases} 0 & \text{for } x \leq 0 \\ x & \text{for } x > 0 \end{cases}$$

## Softmax activation



$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_k e^{z_k}}$$

# Loss functions

A **loss function** indicates how well we are doing. We aim to get the lowest loss value possible.

Mean squared error:

$$L = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

Binary cross-entropy:

$$L = -\frac{1}{N} \sum_{i=1}^N (y^{(i)} \log(\hat{y}^{(i)}) + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}))$$

Categorical cross-entropy:

$$L = -\frac{1}{N} \sum_{i=1}^N \sum_{c=1}^C y_c^{(i)} \log(\hat{y}_c^{(i)})$$

# Model training

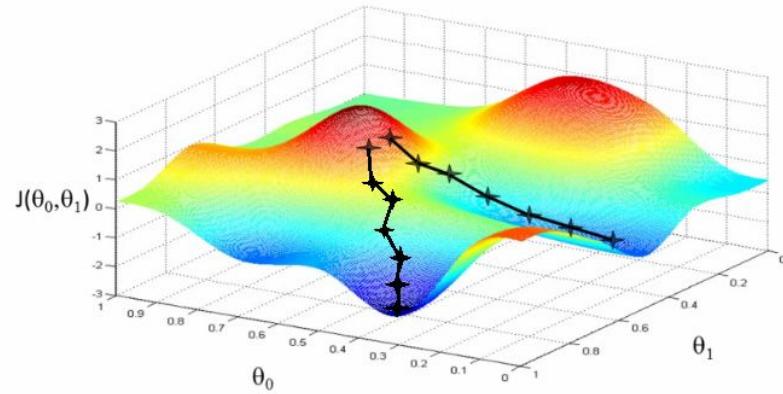
During training, update the parameters in order to get a lower loss.

Calculate the gradients (partial derivatives) for all the parameters using **backpropagation**.

Then update each parameter using gradient descent, by taking small steps in the negative direction of the partial derivative

$$W = W - \alpha \frac{\partial L}{\partial W}$$

where  $\alpha$  is the learning rate



# Common network setups

For regression (predicting a score):

- Output layer of size 1
- Linear activation for the output layer, so the value is not restricted
- Use Mean Squared Error (MSE) as the loss

For binary classification:

- Output layer of size 1
- Use sigmoid to predict between two classes (0 and 1)
- Use binary cross-entropy as the loss

# Common network setups

For multi-class classification (predicting one class out of many):

- With  $k$  classes, have output layer of size  $k$
- Use softmax activation to get a probability distribution
- Use categorical cross-entropy as the loss

For multi-label classification (possibly predicting many classes):

- With  $k$  classes, have output of size  $k$
- Use sigmoid activation, making each output is an independent binary classifier
- Use binary cross-entropy as a loss on each output neuron

# Additional notes

- Before training starts, parameters should be initialised somehow. Random small values around 0 usually work well.
- Often need regularisation (L2, dropout, etc) to avoid overfitting.
- Normally process multiple datapoints in a batch, to take advantage of GPU parallelisation.
- Adaptive learning rates (AdamW, Adam, AdaDelta, etc.) can make training easier.
- For more detailed video lectures on machine learning and neural networks, see the Intro2ML course website:  
<https://intro2ml.pages.doc.ic.ac.uk/>

# Applications of NLP

# Spam filtering

<input type="checkbox"/>	<input type="star"/>	<input type="envelope"/>	Europub Journal of .	Invitation To Publishing - Scoring Lexical Entailment with a Supervised Directional Similarity Network - Dear Researcher, How...	Jan 13
<input type="checkbox"/>	<input type="star"/>	<input type="envelope"/>	AI2 Incubator	AI2 Incubator Insights 10: Conversational Programming, AI Assistants, Foundation Model Operations - Insights 10: Conversatio...	Jan 10
<input type="checkbox"/>	<input type="star"/>	<input type="envelope"/>	Let's Keto Capsules	Let's Keto Capsules - melt fat fast! without diet or exercise - Let's KETO Capsules - Voted #1 Weight Loss Product in RSA WHY IS ...	Jan 10
<input type="checkbox"/>	<input type="star"/>	<input type="envelope"/>	Let's Keto Capsules	Let's Keto Capsules - melt fat fast! without diet or exercise - Let's KETO Capsules - Voted #1 Weight Loss Product in RSA WHY IS ...	Jan 10
<input type="checkbox"/>	<input type="star"/>	<input type="envelope"/>	Andreia Cruz	1st Call for Papers - 8th International Conference on Big Data Analytics, Data Mining and Computational Intelligence, Porto, ...	Jan 9
<input type="checkbox"/>	<input type="star"/>	<input type="envelope"/>	IIIS 2023 Spring Co.	[REMINDER] January 18th LAST Submission Deadline - Dear Marek Rei, This is a friendly reminder that January 18 th , 2023 is the ...	Jan 9
<input type="checkbox"/>	<input type="star"/>	<input type="envelope"/>	IIIS 2023 Spring Co.	[REMINDER] January 18th LAST Submission Deadline - Dear Marek Rei, This is a friendly reminder that January 18 th , 2023 is the ...	Jan 9
<input type="checkbox"/>	<input type="star"/>	<input type="envelope"/>	"Mis...@...ac-orleans-t...	Gud välsigne dig - Hej min kärä Jag vet att min post kommer till dig som en överraskning eftersom du inte känner mig. Snälla, jag är ...	Jan 7
<input type="checkbox"/>	<input type="star"/>	<input type="envelope"/>	Arthur/E-Able Power	Solar Energy Product 2023 - Hello, Sorry if disturb and Happy New year 2023! We are a manufacturer of Solar Energy products, are ...	Jan 6
<input type="checkbox"/>	<input type="star"/>	<input type="envelope"/>	acad...@academic-co...	Academic and research positions at Computeroxy - Dear Colleague, We are pleased to send you the next edition of our specialise...	Jan 5
<input type="checkbox"/>	<input type="star"/>	<input type="envelope"/>	Arthur/E-Able Power	Solar Energy Product 2023 - Happy New year 2023! We are a manufacturer of Solar Energy products, are you interested to get the l...	Jan 4
<input type="checkbox"/>	<input type="star"/>	<input type="envelope"/>	acad...@academic-co...	Academic posts at Computeroxy - Dear Colleague, We are pleased to send you the next edition of our specialised journal including...	12/31/22
<input type="checkbox"/>	<input type="star"/>	<input type="envelope"/>	cfp.esp@whioce.net	Paper Invitation: [Environment and Social Psychology] (Indexed in Scopus) Special Issue "Implications of Emotional Intellige...	12/30/22

# Translation

The screenshot shows the Google Translate interface. At the top, there are three tabs: 'Text' (selected), 'Documents', and 'Websites'. Below the tabs, language selection bars are shown for both the source and target languages. The source language is set to 'ESTONIAN' and the target language is set to 'ENGLISH'. The input text in Estonian is: 'Laskesuusatamise MK-etapil Saksamaal Ruhpoldingis võitis laupäevase naiste 4x6 km teatesõidu Norra, Eesti naiskond sai hea sõidu juures kaheksanda koha.' The translated output in English is: 'At the MK stage of biathlon in Ruhpolding, Germany, Norway won Saturday's women's 4x6 km relay, while the Estonian women's team took eighth place with a good run.' There are also icons for microphone, speaker, and sharing at the bottom.

Text Documents Websites

DETECT LANGUAGE ESTONIAN ENGLISH SPAI ↗ ENGLISH ESTONIAN SPANISH ↗

Laskesuusatamise MK-etapil Saksamaal Ruhpoldingis võitis laupäevase naiste 4x6 km teatesõidu Norra, Eesti naiskond sai hea sõidu juures kaheksanda koha.

At the MK stage of biathlon in Ruhpolding, Germany, Norway won Saturday's women's 4x6 km relay, while the Estonian women's team took eighth place with a good run.

152 / 5,000

<https://translate.google.com/>

# Grammar correction



Untitled document

This work investigates minimally-supervised methods for solving NLP tasks, without requiring explicit annotation or training data. Our motivation is to create systems that require substantially reduced effort from domain and/or NLP experts, compared to annotating a corresponding dataset, and also offer easier domain adaptation and better generalisation properties. We apply these principles to four separate language processing tasks and analyse their performance compared to supervised alternatives. First, we investigate the task of detecting the scope of speculative language, and develop a system that applies manually-defined rules over dependency graphs. Next, we experiment with distributional similarity measures for detecting and generating hyponyms, and describe a new measure that achieves the highest performance on hyponym generation. We also extend the distributional hypothesis to larger structures and propose the task of

## All suggestions

- requiring · Correct your spelling

### SPELLING

reduced → reduced

The word **reduced** doesn't seem to fit this context. Consider replacing it with a different one.

[Learn more](#)



- language, · Remove the comma

- hyponyms, · Remove the comma

- own · Remove redundancy

HIDE ASSISTANT >

80

Overall score

See performance >

Goals

Adjust goals >

## All suggestions

Correctness

5 alerts

Clarity

A bit unclear

Engagement

A bit bland

Get Export

# Text prediction

18:18 31 ↶

G what is the quickest rou| ×

what is the quickest rou

what is the quickest route to australia



18:19 31 ↶

G why do people ×

why do people

why do people vape



why do people yawn



why do people snore



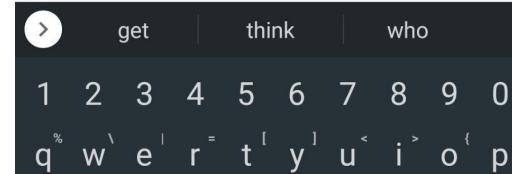
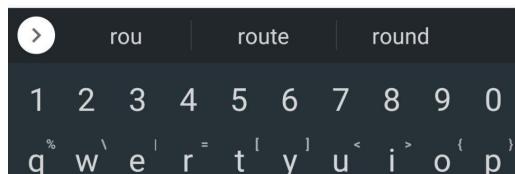
why do people cry



why do people sneeze



why do people stare at me in public



# GPT-3



Overview Documentation Examples **Playground**

Upgrade

Help

Personal

## Get started



Enter an instruction or select a preset, and watch the API respond with a **completion** that attempts to match the context or pattern you provided.

You can control which **model** completes your request by changing the model.

### KEEP IN MIND

- ⚠ Use good judgment when sharing outputs, and attribute them to your name or company. [Learn more](#).
- ⚠ Requests submitted to our models may be used to train and improve future models. [Learn more](#).
- ⚠ Our default models' training data cuts off in 2021, so they may not have knowledge of current events.

## Playground

Load a preset...

Save

View code

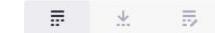
Share



What is a fixed asset?

A fixed asset is a long-term tangible piece of property or equipment that a business owns and uses to generate income. Examples of fixed assets include buildings, land, furniture, vehicles, machinery, and equipment. These assets are held for more than one year and are not expected to be converted into cash within the fiscal year.

Mode



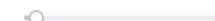
Model



Temperature 0.7



Maximum length 256



Stop sequences

Enter sequence and press Tab

Looking for ChatGPT? [Try it now](#)

Submit



Note: The output of these models is always changing. You will likely get slightly different output when trying the same input.

# GPT-3



Overview Documentation Examples **Playground**

⚡ Upgrade

?



Personal

## Get started



Enter an instruction or select a preset, and watch the API respond with a **completion** that attempts to match the context or pattern you provided.

You can control which **model** completes your request by changing the model.

## KEEP IN MIND

⚠ Use good judgment when sharing outputs, and attribute them to your name or company. [Learn more](#).

⚠ Requests submitted to our models may be used to train and improve future models. [Learn more](#).

ⓘ Our default models' training data cuts off in 2021, so they may not have knowledge of current events.

## Playground

Load a preset...

Save

View code

Share



Write the first act for a play where two old friends meet after a long time in Paris and one holds a secret grudge against the other.

### ACT ONE

(The scene opens on the streets of Paris. A beautiful day and the sounds of the city fill the air. Two old friends, JACOB and JENNY, are walking together. Both are in their late twenties and appear to be in good spirits. They have not seen each other in years.

JACOB: Wow, it's been so long! How have you been?

JENNY: I've been good! It's so great to see you again.

JACOB: Yeah, I can't believe it's been years since we last saw each other.

JENNY: Me neither! So, what have you been up to?

JACOB: Oh, you know, the usual. Working, travelling, enjoying life.

ⓘ Looking for ChatGPT? [Try it now](#)

Submit



Mode



Model

text-davinci-003

Temperature 0.7

Maximum length 256

Stop sequences  
Enter sequence and press Tab

Top P 1

Frequency penalty 0

Presence penalty 0

Best of 1

# Stable Diffusion (text to image)

Input: "A smiling bear with a hat strolling through the streets of London"



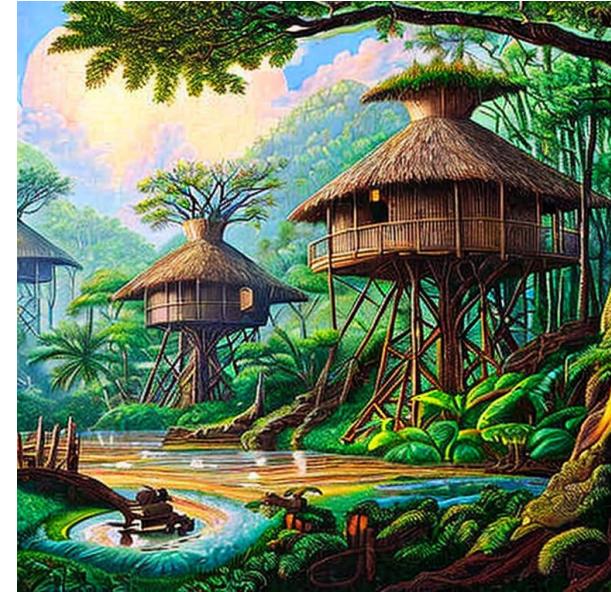
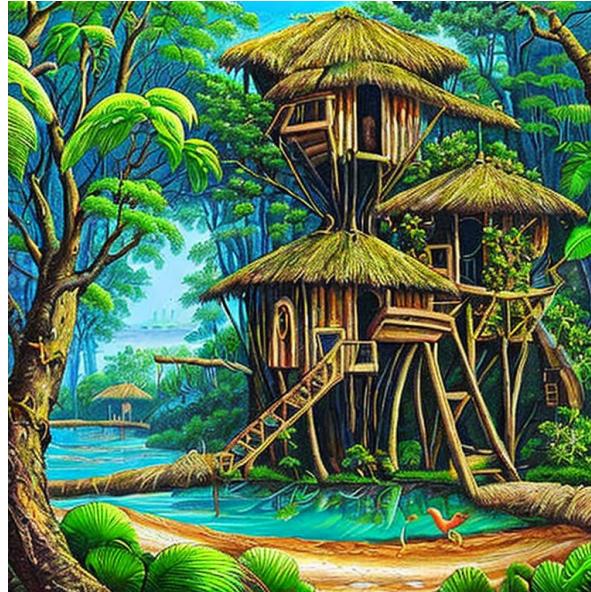
# Stable Diffusion (text to image)

Input: "An elven warrior in green armour fighting a large badger with a broad sword."



# Stable Diffusion (text to image)

Input: "A painting of a mysterious futuristic treehouse village in the Amazon jungle"



# ChatGPT

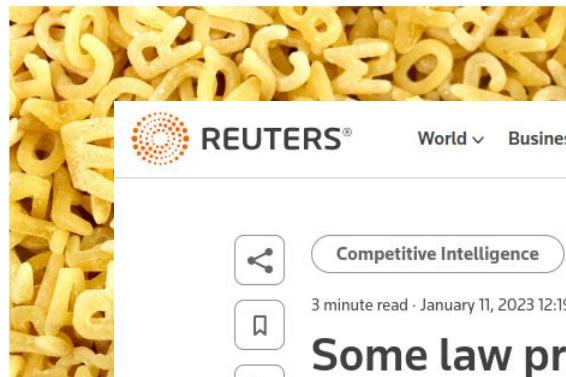
≡ WIRED

BACKCHANNEL BUSINESS CULTURE GEAR IDEAS

*Microsoft Bets Big on the Creator of ChatGPT in Race to Dominate A.I.*

## ChatGPT Has Investors Bring Home the Bacon?

The loquacious bot has Microsoft ready to sink a reported \$1 billion into OpenAI, and new products can be built on the technology.



REUTERS®

World ▾ Business ▾ Legal ▾ Markets ▾ Breakingviews Technology ▾ Investigati

butor [Follow](#)

Jan 13, 2023, 11:20am EST



Competitive Intelligence

Legal Innovation

Legal Industry

Data Privacy



3 minute read · January 11, 2023 12:19 AM GMT · Last Updated 4 days ago



## Some law professors fear ChatGPT's rise as others see opportunity

By Karen Sloan

As a new chatbot wows the world with its conversational abilities, the resurgent tech giant is poised to reap a relationship with the start-up OpenAI.

Forbes

FORBES > BUSINESS > RETAIL



## Beyond The Buzz: 7 Ways Brands Can Use ChatGPT And OpenAI For Boosting Sales On Amazon

butor [Follow](#)

Jan 13, 2023, 11:20am EST

ChatGPT, openAI and DALL-E have dominated the newsfeed of marketers and consumers alike. But how much is simply buzz, versus a business opportunity that's real? That's what this is just the start of a big shift in how we implement digital technologies across many channels. Many more use cases will come out of the further experimentation and technology overlays. In a few months or even years, look back on this list and consider it narrow and elementary.

That's what this is just the start of a big shift in how we implement digital technologies across many channels. Many more use cases will come out of the further experimentation and technology overlays. In a few months or even years, look back on this list and consider it narrow and elementary.

# ChatGPT

MA

How could London improve traffic management?



There are several ways London could improve traffic management:



1. Implement intelligent traffic management systems, such as traffic signal optimization and real-time traffic monitoring, to more efficiently manage the flow of vehicles.
2. Encourage the use of public transportation, cycling, and walking by investing in infrastructure and offering incentives for individuals to use these modes of transportation.
3. Utilize congestion pricing to reduce the number of cars on the road during peak travel times.
4. Invest in and promote carpooling and ride-sharing initiatives to reduce the number of vehicles on the road.
5. Develop and implement a comprehensive transportation plan that addresses the needs of all road users, including drivers, cyclists, and pedestrians.
6. Invest in technology like smart sensors that can detect traffic

⟳ Regenerate response

<https://chat.openai.com>

# ChatGPT

MA

Tell me more about number 4



Investing in and promoting carpooling and ride-sharing initiatives is a way to reduce the number of vehicles on the road and improve traffic management. Carpooling is when several people share a car to travel to the same destination, such as work or school. Ride-sharing is when individuals use a car-sharing service, such as Uber or Lyft, to get to their destination.



Carpooling and ride-sharing can reduce traffic congestion by reducing the number of cars on the road, which in turn can reduce travel time, fuel consumption and air pollution. It can also reduce the demand for parking and improve mobility for people who do not own a car. Carpooling and ride-sharing can also be more cost-effective for individuals, as they can split the cost of fuel and tolls.

To promote carpooling and ride-sharing, governments and organizations can provide incentives such as carpool lanes, tax credits. They can also invest in carpooling and ride-shari

Regenerate response



# ChatGPT

Carpooling and ride-sharing can reduce traffic congestion by reducing the number of cars on the road, which in turn can reduce travel time, fuel consumption and air pollution. It can also reduce the demand for parking and improve mobility for people who do not own a car. Carpooling and ride-sharing can also be more cost-effective for individuals, as they can split the cost of fuel and tolls.

To promote carpooling and ride-sharing, governments and organizations can provide incentives such as carpool lanes, reduced parking fees, and tax credits. They can also invest in carpooling and ride-sharing platforms and technologies, such as ride-matching apps, to make it easier for people to find and connect with potential carpool or ride-share partners.

Additionally, governments and organizations can educate the public about the benefits of carpooling and ride-sharing, and encourage employers to implement carpooling and ride-sharing programs for their employees.

# ChatGPT

MA

Given an array of integers `nums` and an integer `target`, return indices of the two numbers such that they add up to `target`.

You may assume that each input would have exactly one solution, and you may not use the same element twice.

You can return the answer in any order.



Here is one way to solve this problem using a hash map:



1. Create an empty hash map
2. Iterate through the array of integers, `nums`
3. For each element in the array, check if the target minus the current element exists in the hash map
4. If it does, return the current element's index and the index of the target minus the current element from the hash map
5. If it doesn't, add the current element to the hash map

Regenerate response



<https://chat.openai.com>

# ChatGPT

4. If it does, return the current element's index and the index of the target minus the current element from the hash map
5. If it doesn't, add the current element and its index to the hash map
6. Return the indices of the two numbers that add up to the target

```
Copy code
```

```
def twoSum(nums, target):
    hash_map = {}
    for i, num in enumerate(nums):
        if target - num in hash_map:
            return [hash_map[target - num], i]
        hash_map[num] = i
```

This solution has a time complexity of  $O(n)$  and a space complexity of  $O(n)$

⟳ Regenerate response



<https://chat.openai.com>

# ChatGPT

AI is not quite solved yet...

MA

Can you name NLP researchers at Imperial College London?



Imperial College London is a prestigious research university located in London, England, with a strong tradition in the field of natural language processing (NLP). Some researchers in the field of NLP at Imperial College London include:

1. Sebastian Riedel: Professor of Computer Science and Head of the Natural Language Processing group at Imperial College London.
2. Tim Baldwin: Professor of Computer Science and Head of the Computational Linguistics group at Imperial College London.
3. Andrew C. Brock: Assistant Professor of Computer Science at Imperial College London.
4. Eva Miller: Assistant Professor of Computer Science at Imperial College London.
5. Christian Rupprecht: Assistant Professor of Computer Science at Imperial College London.

<https://chat.openai.com>

# ChatGPT

AI is not quite solved yet...

MA

Can you name NLP researchers at Imperial College London?



Imperial College London is a prestigious research university located in London, England, with a strong tradition in the field of natural language processing (NLP). Some researchers in the field of NLP at Imperial College London include:

1. Sebastian Riedel: Professor of Computer Science and Head of the Natural Language Processing group at Imperial College London.
2. Tim Baldwin: Professor of Computer Science and Head of the Computational Linguistics group at Imperial College London.
3. Andrew C. Brock: Assistant Professor of Computer Science at Imperial College London.
4. Eva Miller: Assistant Professor of Computer Science at Imperial College London.
5. Christian Rupprecht: Assistant Professor of Computer Science at Imperial College London.

NLP at DeepMind/UCL

NLP at MBZUAI/Melbourne

US Republican politician

UCL historian OR Instagram influencer

Computer vision at Oxford

# ChatGPT

MA

Which NLP researchers are at Imperial College London?



I am not aware of specific individuals at Imperial College London working on NLP research, but the college is known for its research in the field of Artificial Intelligence and Natural Language Processing, and it is likely that there are several researchers at the institution working in these areas. Some notable researchers in AI and NLP at Imperial College London are Dr. Yulia Tsvetkov, Dr. Ivan Titov, Dr. Sebastian Riedel, Dr. Pascale Fung, Dr. Marek Rei, Dr. John Shawe-Taylor and Dr. Mark Fisher.

Yay!

Contradicts itself

# ChatGPT

KA

If 1 woman can make 1 baby in 9 months, how many months does it take 9 women to make 1 baby? Explain each step you used to arrive at your answer.



First, we need to determine how many babies each woman makes per month. We can do this by dividing the number of babies each woman makes (1) by the number of months it takes for each woman to make a baby (9). This gives us a rate of  $1/9$  babies per month for each woman.

Next, we need to determine how many babies 9 women make per month. We can do this by multiplying the number of women (9) by the rate at which each woman makes babies ( $1/9$  babies per month). This gives us a total rate of  $9 * (1/9) = 1$  baby per month.

Finally, we need to determine how many months it takes for 9 women to make 1 baby. We can do this by dividing the number of babies we want (1) by the rate at which the women are making babies (1 baby per month). This gives us a total of  $1 / 1 = 1$  month.

Therefore, it takes 9 women 1 month to make 1 baby.

<https://twitter.com/KaiaVintr/status/1607453952525881344>

This seems to have been  
fixed by OpenAI already

# Discussion

- NLP has many practical applications, e.g. SA and MT
- NLP is hard, mainly due to **ambiguities, non-literal meaning, extra-linguistic context**
- ML and Deep ML made NLP more effective, both for intermediate tasks and final applications
- **In this module** we will see how some of these tasks and applications are modelled using standard ML and DL
  - Set of methods/algorithms that can be used for many more applications

# Extra slides

The following slides are for your reference only, they will not be covered in the lectures

# Feed forward nets - maths

Each row of neurons can be thought of as a **vector**

- Input layer is a 4 dimensional vector ( $\mathbf{x}$ ), layer above is a 6 dimensional vector ( $\mathbf{h}_1$ ), etc.

Fully connected layer (**affine layer**)

- Each neuron connected to all neurons of next layer
- Vector-matrix multiplication,  $\mathbf{h} = \mathbf{xW}$  where the weight of the connection from the  $i$ -th neuron in the input row to the  $j$ -th neuron in the output row is  $W_{ij}$
- E.g. linear transformation from 4 to 6 dimensions

# Feed forward nets - maths

The values of  $\mathbf{h}$  are transformed by a non-linear function  $g$  that is applied to each value before being passed on to the next layer

The whole computation from input to output can be written as

$$(g(\mathbf{x}\mathbf{W}_1))\mathbf{W}_2$$

where  $\mathbf{W}_1$  are the weights of the first layer and  $\mathbf{W}_2$  are the weights of the second one

# Multi-layer perceptron

Perceptron is a linear function of its inputs

$$NN_{Perceptron}(\mathbf{x}) = \mathbf{x}\mathbf{W} + \mathbf{b}$$
$$\mathbf{x} \in \mathbb{R}^{d_{in}}, \quad \mathbf{W} \in \mathbb{R}^{d_{in} \times d_{out}}, \quad \mathbf{b} \in \mathbb{R}^{d_{out}}$$

where **W** is the weight matrix, **b** is a bias term

To make it 'deep', add non-linear hidden layer

$$NN_{MLP1}(\mathbf{x}) = g(\mathbf{x}\mathbf{W}^1 + \mathbf{b}^1)\mathbf{W}^2 + \mathbf{b}^2$$
$$\mathbf{x} \in \mathbb{R}^{d_{in}}, \quad \mathbf{W}^1 \in \mathbb{R}^{d_{in} \times d_1}, \quad \mathbf{b}^1 \in \mathbb{R}^{d_1}, \quad \mathbf{W}^2 \in \mathbb{R}^{d_1 \times d_2}, \quad \mathbf{b}^2 \in \mathbb{R}^{d_2}$$

# Multi-layer perceptron

Can add more linear and non-linear transformations:

$$NN_{MLP2}(\mathbf{x}) = (g^2(g^1(\mathbf{x}\mathbf{W}^1 + \mathbf{b}^1)\mathbf{W}^2 + \mathbf{b}^2))\mathbf{W}^3$$

which can also be written as:

$$NN_{MLP2}(\mathbf{x}) = \mathbf{y}$$

$$\mathbf{h}^1 = g^1(\mathbf{x}\mathbf{W}^1 + \mathbf{b}^1)$$

$$\mathbf{h}^2 = g^2(\mathbf{h}^1\mathbf{W}^2 + \mathbf{b}^2)$$

$$\mathbf{y} = \mathbf{h}^2\mathbf{W}^3$$

The vector resulting from each transform is a “**layer**”

# Important concepts - dimensionality

Dimensions of layers, input & output need to be defined

A layer will expect a  $d_{in}$  dimensional vector as its input, and transform it into a  $d_{out}$  dimensional vector

Dimensionality of the layer = output dimensionality

Dimension of  $x$  is  $1 \times d_{in}$

Dimension of  $W$  is  $d_{in} \times d_{out}$

Dimension of  $b$  is  $1 \times d_{out}$

# Important concepts - dimensionality

Output:

- If  $d_{\text{out}} = 1$ , the network's output is a scalar
  - **Regression** by taking the **value** of the output, or for **binary classification** by consulting the **sign** of the output
- If  $d_{\text{out}} = k > 1$ 
  - **k-class classification**, by associating each dimension with a class & looking for the dimension with maximal value

# Important concepts - output

Output layer can be transformed, e.g. **softmax**

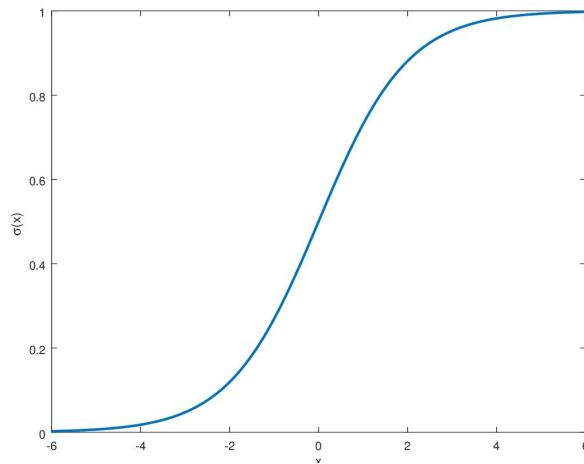
$$\mathbf{x} = x_1, \dots, x_k$$

$$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^k e^{x_j}}$$

**Result** = vector of non-negative real numbers that sum to 1: probability distribution over k possible classes

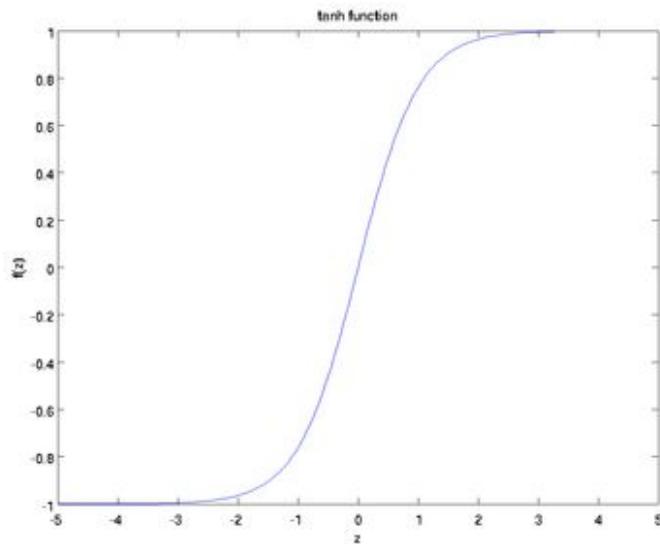
# Important concepts - non-linearities

- Which function to apply? Empirical decision
- Common non-linearities
  - The **sigmoid** activation function  $\sigma(x) = 1/(1 + e^{-x})$  is an S-shaped function, makes each value  $x$  into range  $[0, 1]$



# Important concepts - non-linearities

The hyperbolic tangent (**tanh**) is an S-shaped activation function, transforming the values  $x$  into the range  $[-1, 1]$

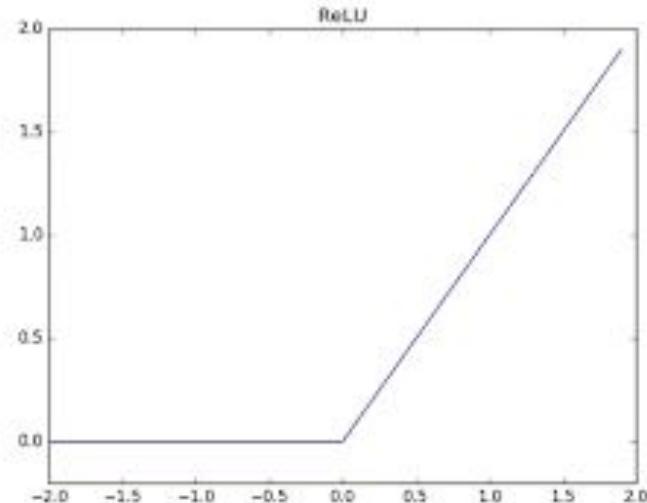


$$\tanh(x) = \frac{e^{2x}-1}{e^{2x}+1}$$

# Important concepts - non-linearities

Rectifier (**ReLU**) - very simple activation function that clips each value  $x < 0$  at 0

$$ReLU(x) = \max(0, x) = \begin{cases} 0 & x < 0 \\ x & \text{otherwise} \end{cases}$$



# Important concepts - input

## Embeddings

- $\mathbf{x}$  is usually composed of various embeddings vectors
- $c(\cdot)$  is a function from input to embedding vectors
- $c(\cdot)$  generally converts each feature into an embedding vector and concatenates (or sums or averages) them

$$\mathbf{x} = c(f_1, f_2, f_3) = [v(f_1); v(f_2); v(f_3)]$$

$$\begin{aligned}NN_{MLP1}(\mathbf{x}) &= NN_{MLP1}(c(f_1, f_2, f_3)) \\&= NN_{MLP1}([v(f_1); v(f_2); v(f_3)]) \\&= (g([v(f_1); v(f_2); v(f_3)] \mathbf{W^1} + \mathbf{b^1})) \mathbf{W^2} + \mathbf{b^2}\end{aligned}$$

# Important concepts - input

## Embeddings

$$\mathbf{x} = c(f_1, f_2, f_3) = v(f_1) + v(f_2) + v(f_3)$$

$$\begin{aligned} NN_{MLP1}(\mathbf{x}) &= NN_{MLP1}(c(f_1, f_2, f_3)) \\ &= NN_{MLP1}(v(f_1) + v(f_2) + v(f_3)) \\ &= (g((v(f_1) + v(f_2) + v(f_3)) \mathbf{W^1} + \mathbf{b^1})) \mathbf{W^2} + \mathbf{b^2} \end{aligned}$$

Sum (or average) vs concatenation: what would the consequences be?  
E.g. input is a sentence.

# Important concepts - input

- $c(\cdot)$  can be **part of the network**:  $v(f_i)$  as result of an “embedding layer” or “lookup layer”
- Given vocabulary of  $|V|$  words, each embedded as a  $d$  dimensional vector
  - Collection of vectors =  $|V| \times d$  embedding matrix  $\mathbf{E}$  where each row corresponds to an embedded feature
  - **One-hot vector** =  $f_i$  is a  $|V|$ -dimensional vector, which is all zeros except from one index: value 1 for  $i$ -th feature
  - $f_i \mathbf{E}$  will select the corresponding row of  $\mathbf{E}$
  - $v(f_i)$  can be defined in terms of  $\mathbf{E}$  and  $f_i$ :  $v(f_i) = f_i \mathbf{E}$

# Important concepts - input

- `c(.)` can also be **pre-computed**
- Network's inputs are dense real-valued input vectors
- Many good pre-computed embeddings out there!

# Important concepts - parameters

- $\theta$  = collection of **all parameters** of the network: weight matrices  $W_i$ , and the biases terms  $b_i$ , and optionally embeddings  $E$
- Together with the input, parameters determine the network's output
- The **training algorithm** aims to set parameter values such that the network's predictions are less erroneous

# Important concepts - training loss

What does it mean to be less erroneous?

- Loss functions  $L(\hat{y}, y)$  state the loss (a scalar) of predicting  $\hat{y}$  when the true output is  $y$
- Can be any function mapping two vectors to a scalar
- Training objective is to **minimise loss** across training set
- For purposes of optimisation, use functions for which (sub-)gradients can be computed easily

# Important concepts - training loss

Common loss functions:

- Hinge/**margin** loss for binary classification:  $y$  is in  $\{-1,1\}$ ; classification rule is  $\text{sign}(\hat{y})$ ; class is correct if  $y.\hat{y} > 0$ :  
$$\text{loss}(y, \hat{y}) = \max(0, 1-y.\hat{y})$$
  
loss is 0 if  $y$  &  $\hat{y}$  share same signal, otherwise it is linear
- Hinge (multiclass)
- Log loss
- Categorical **cross-entropy** loss / negative log-likelihood
- Ranking losses, ...

# Important concepts - training algorithm

## Gradient-based training algorithms

- Repeatedly compute an estimate of the error over dataset
- Compute the gradient of the error wrt the parameters
- Move the parameters in the direction of the gradient

## Basic algorithm: **Stochastic Gradient Descent** (SGD)

- Commonly computed over minibatches

# Important concepts - training algorithm

How to compute the gradients of the network's error with respect to **all** parameters, over all layers?

- **Backpropagation algorithm** = the backward propagation of errors
- Compute derivatives of a complex expression using the **chain rule**
  - Error is computed at the output (a layer) and distributed backwards throughout the network's layers

# Discussion

What made deep learning feasible?

- Large amounts of training **data**
- Faster **machines** and multicore CPU/GPUs
- New **architectures, algorithms, ideas**
  - Better, more flexible learning of representations
  - Effective end-to-end joint system learning
  - Effective learning methods for using contexts and transferring between tasks
- Better **regularisation** and **optimisation** methods

# Applications - use components

- **Sentiment analysis**
- Offensive language detection and categorisation
- Question answering (Watson, Google Search)
- **Machine translation (Google Translate)**
- Text summarisation
- Text prediction (email, phone...)
- Dialogue systems ... (SIRI, Alexa, Google assistant)

# Do we really need all these components?

It depends on the application... E.g.:

- **Sentiment analysis**: basic pre-processing, i.e. tokenisation, stop word removal, lemmatisation
- **Question answering**: basic pre-processing plus parsing and maybe semantic role labelling
- **Dialogue systems**: All of these and more!

# How good are these components?

It depends on the application and data (F1)

- POS tagging: 98
- Parsing: 96
- WSD: 81
- Co-reference resolution: 80
- ...

Some sources for finding state-of-the-art:

<https://paperswithcode.com/sota>

<https://github.com/sebastianruder/NLP-progress>