

# Unleashing the Power of Large Language Models: A Hands-On Tutorial

Payel Santra<sup>1</sup>, Madhusudan Ghosh<sup>1</sup>, Shrimon Mukherjee<sup>1</sup>, Debasis Ganguly<sup>2</sup>, Partha Basuchowdhuri<sup>1</sup>, Sudip Kumar Naskar<sup>3</sup>

<sup>1</sup>Indian Association for the Cultivation of Science

<sup>2</sup>University of Glasgow

<sup>3</sup>Jadavpur University



December 27, 2023

# Plan of the Tutorial

- 1 Plan of the Tutorial
- 2 Introduction to NLP
- 3 Overview of Distributional Representation Learning for NLP
- 4 Overview of Transformer based Language Model
- 5 Overview of Large Language Models
- 6 Concept of in-context learning and its application
- 7 Conclusion

# Plan of the Tutorial

- 1 Plan of the Tutorial
- 2 Introduction to NLP
- 3 Overview of Distributional Representation Learning for NLP
- 4 Overview of Transformer based Language Model
- 5 Overview of Large Language Models
- 6 Concept of in-context learning and its application
- 7 Conclusion

# Plan of the Tutorial

- ① Introduction to NLP
  - Sudip Kumar Naskar
- ② Overview of Distributional Representation Learning for NLP
  - Partha Basuchowdhuri
- ③ Overview of Transformer-based Pretrained Language Model
  - Madhusudan Ghosh
- ④ Overview of Large Language Models
  - Payel Santra
- ⑤ Concept of in-context learning and its application
  - Debasis Ganguly
- ⑥ Hands-on Coding/Demo Session
  - Debasis Ganguly, Shrimon Mukherjee, Madhusudan Ghosh, & Payel Santra

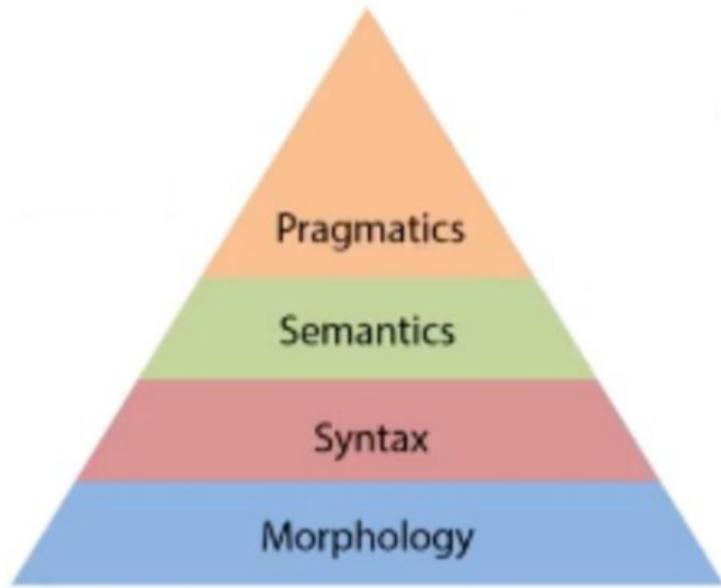
# Plan of the Tutorial

- 1 Plan of the Tutorial
- 2 Introduction to NLP
- 3 Overview of Distributional Representation Learning for NLP
- 4 Overview of Transformer based Language Model
- 5 Overview of Large Language Models
- 6 Concept of in-context learning and its application
- 7 Conclusion

# Natural Language Processing (NLP)

- What is NLP?
- What makes NLP difficult?

# NLP Pyramid



# Ambiguity in Natural Language

- Lexical Ambiguity
- Structural Ambiguity
- Semantic Ambiguity

# Lexical Ambiguity

- When a single word possesses more than one interpretation
  - Morphology can be ambiguous
    - does drier mean 'more dry' or 'machine which dries' ?
  - Words can represent different parts of speech
    - E.g., ship, round, etc.
  - Words can have multiple meanings
    - E.g., bank, letter, ball
- How to disambiguate?
  - Look at the context.
    - "John kicked a ball."
    - "The ball lasted until midnight."
    - "When you hold a ball..."

# Structural Ambiguity

- When multiple syntactic structures can be associated with a sentence.
- E.g.,
  - “Flying planes can be dangerous.”
  - “Visiting relatives can be boring.”
  - “Time flies like an arrow.”
  - “Fruit flies like banana.”
  - “The boy saw the girl with a telescope.”

# Semantic Ambiguity

- How about meanings of polysemous words?

He deposited money in a **bank** account  
with a high **interest** rate.

Sitting on the **bank** of the Mississippi,  
a passing ship piqued his **interest**.

# Multiword Units

- Phrasal Verbs

- “John eats in on Sundays.”
- “John eats in a restaurant on weekdays.”
- “John fell for Mary.”
- “John fell for a lie.”

- Idioms

- kick the bucket
- let your hair down

# Coreference Resolution

- Example

- “The police opened fire at the crowd, and some of them fell.”
- “The police opened fire at the crowd, and some of them missed.”
- Winograd Schema Challenge: “The city councilmen refused the demonstrators a permit because they [feared/advocated] violence”

- Utility in Chatbots



Source:

<https://medium.com/huggingface/state-of-the-art-neural-coreference-resolution-for-chatbots-3302365dcf30>

# Different Levels of Ambiguity

- E.g.,
  - “He noticed her shaking hands.”
  - “I like tennis.”
  - “John is eager to please.”
  - “John is easy to please.”
  - “The chicken is ready to eat.”
  - NLP?

# Ambiguity is Pervasive

- Do you see any ambiguity in this?
  - I made her duck
    - I cooked waterfowl for her benefit (to eat)
    - I cooked waterfowl belonging to her
    - I created the (plaster?) duck she owns
    - I caused her to quickly lower her head or body
    - I waved my magic wand and turned her into undifferentiated waterfowl

# Ambiguity is Pervasive

- Do you see any ambiguity in this?
  - I made her duck
    - I cooked waterfowl for her benefit (to eat)
    - I cooked waterfowl belonging to her
    - I created the (plaster?) duck she owns
    - I caused her to quickly lower her head or body
    - I waved my magic wand and turned her into undifferentiated waterfowl

# Ambiguity in Speech Processing

- Phonetics! – I made her duck
  - I mate or duck
  - I'm eight or duck
  - Eye made her duck
  - Aye mate, her duck
  - I maid her duck
  - I'm aid her duck
  - I mate her duck
  - I'm ate her duck
  - I'm ate or duck
  - I mate or duck

# More Ambiguity



- Difficult to solve

- “Teacher Strikes Idle Kids”
- “oil prices rose gas prices rose up”
- A ship-shipping ship ships shipping-ships.
- Will, will Will will Will Will’s will?
- “Buffalo buffalo Buffalo buffalo  
buffalo buffalo Buffalo buffalo”

# Why else is NLP difficult?

## non-standard English

Great job @justinbieber! Were SOO PROUD of what youve accomplished! U taught us 2 #neversaynever & you yourself should never give up either ❤️

## segmentation issues

the New York-New Haven Railroad  
the New York-New Haven Railroad

## idioms

dark horse  
get cold feet  
lose face  
throw in the towel

## neologisms

unfriend  
Retweet  
Bromance  
Netizens

## world knowledge

Mary and Sue are sisters.  
Mary and Sue are mothers.

## tricky entity names

Where is *A Bug's Life* playing ...  
*Let It Be* was recorded ...  
... a mutation on the *for* gene  
...

But that's what makes it fun!

# Applications of NLP

To name a few...

- Information Retrieval and Extraction
- Web Search
- Text Categorization
- Text Summarization
- Sentiment Analysis
- Math Word Problems
- Recommendation Systems
- Knowledge Extraction
- Fact Checking

# Applications of NLP

To name a few...

- Machine Translation
- Question Answering
  - On Unstructured data (i.e., Text)
  - On Structured data
    - On Database (NL query → SQL)
    - On Knowledge Base (NL query → SPARQL)
- Text Normalization
  - Spell checker
  - Grammar Correction
  - Disfluency Detection in Natural Speech

# Applications of NLP

To name a few...

- Text Similarity
  - Plagiarism Detection
  - Answer / Essay grading
- Readability Assessment and Text Simplification
- Event extraction and Causality
- Natural Language Generation
- Domain-specific NLP
  - FinNLP, BioNLP, LegalNLP
- Multimodal Processing

# Applications of NLP

## Text Classification

- Assigning subject categories, topics, or genres
- Spam detection
- Authorship identification
- Age/gender identification
- Aggression/Hate Speech detection
- Personality identification
- Language identification
- Sentiment analysis
- ...

# Applications of NLP

To name a few...

- Speech Processing
  - Core Tasks
    - Speech Recognition (ASR)
    - Speech Synthesis
  - Additional Tasks
    - Speaker Identification
    - Speech Diarization
    - Normalization of ASR o/p
      - Disfluency Detection
    - Dialog Systems

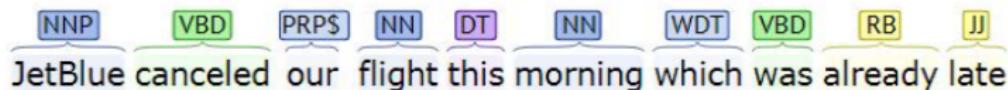
# Core NLP Tasks

- Basic Text Processing
  - Sentence segmentation
  - Word Tokenization
  - Normalization
- Lexical Analysis
  - Lemmatization (e.g., comes/coming/came → come)
  - Morphological Analysis (e.g., unsuccessfully → un+success+ful+ly)

# Core NLP Tasks

## Syntactic Analysis

- POS Tagging
  - JetBlue canceled our flight this morning which was already late



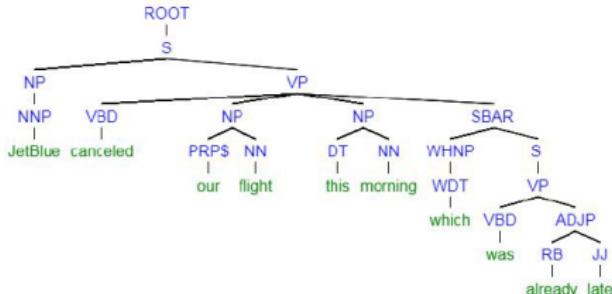
# Core NLP Tasks

## Syntactic Analysis

- POS Tagging

- JetBlue canceled our flight this morning which was already late

```
(ROOT
(S
  (NP (NNP JetBlue))
  (VP (VBD canceled)
    (NP (PRP$ our) (NN flight))
    (NP (DT this) (NN morning)))
  (SBAR
    (WHNP (WDT which))
    (S
      (VP (VBD was)
        (ADJP (RB already) (JJ late))))))))
```

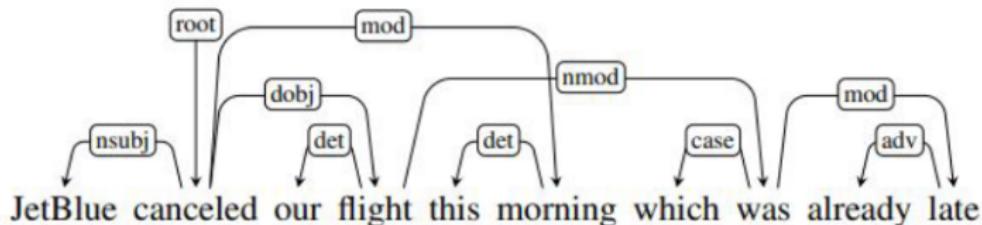
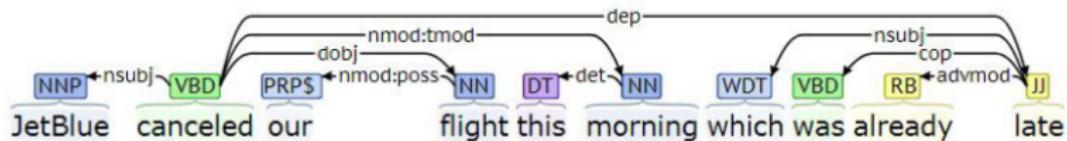


# Core NLP Tasks

## Syntactic Analysis

- POS Tagging

- JetBlue canceled our flight this morning which was already late



# Core NLP Tasks

## Semantic Analysis

- Word Sense Disambiguation (WSD)
- Semantic Similarity
- Word Sense Induction

# Distributional models of meaning

- Also called vector-space models of meaning
- Zellig Harris (1954): “oculist and eye-doctor . . . occur in almost the same environments. . . .
  - If A and B have almost identical environments we say that they are synonyms.
- Firth (1957): “You shall know a word by the company it keeps!”

# Intuition of Distributional Semantics

- Nida example: tesgüino

A bottle of tesgüino is on the table

Everybody likes tesgüino

Tesgüino makes you drunk

We make tesgüino out of corn.

- From context words humans can guess tesgüino
  - means: an alcoholic beverage like beer
- Intuition for the algorithm:
  - Two words are similar if they have similar word contexts.

# NLP Tasks

## More Tasks

- Named Entity Recognition
- Anaphora and Coreference Resolution
- Textual Entailment / Natural Language Inferencing (NLI)
- Paraphrasing
- Entity disambiguation
- ...

# Named Entity Recognition (NER)

A very important sub-task: **find** and **classify** names in text, for example:

- The decision by the independent MP Andrew Wilkie to withdraw his support for the minority Labor government sounded dramatic but it should not further threaten its stability. When, after the 2010 election, Wilkie, Rob Oakeshott, Tony Windsor and the Greens agreed to support Labor, they gave just two guarantees: confidence and supply.

# Named Entity Recognition (NER)

A very important sub-task: **find** and **classify** names in text, for example:

- The decision by the independent MP **Andrew Wilkie** to withdraw his support for the minority **Labor** government sounded dramatic but it should not further threaten its stability. When, after the **2010** election, **Wilkie, Rob Oakeshott, Tony Windsor** and the **Greens** agreed to support **Labor**, they gave just two guarantees: confidence and supply.

# Named Entity Recognition (NER)

A very important sub-task: **find** and **classify** names in text, for example:

- The decision by the independent MP **Andrew Wilkie** to withdraw his support for the minority Labor government sounded dramatic but it should not further threaten its stability. When, after the **2010** election, **Wilkie, Rob Oakeshott, Tony Windsor** and the **Greens** agreed to support **Labor**, they gave just two guarantees: confidence and supply.
- Person, Date, Location, Organization.

# Entity Resolution & Linking

...during the late 60's and early 70's, **Kevin Smith** worked with several local...  
...the term hip-hop is attributed to **Lovebug Starski**. What does it actually mean...



Like Back in 2008, the Lions drafted **Kevin Smith**, even though Smith was badly...  
... backfield in the wake of **Kevin Smith**'s knee injury, and the addition of Haynesworth...



The filmmaker **Kevin Smith** returns to the role of Silent Bob...  
Nothing could be more irrelevant to **Kevin Smith**'s audacious "Dogma" than ticking off...



The Physiological Basis of Politics," by **Kevin Smith**, Douglas Oxley, Matthew Hibbing...



# Plan of the Tutorial

- 1 Plan of the Tutorial
- 2 Introduction to NLP
- 3 Overview of Distributional Representation Learning for NLP
- 4 Overview of Transformer based Language Model
- 5 Overview of Large Language Models
- 6 Concept of in-context learning and its application
- 7 Conclusion

# Sparse Vector Representations

## One Hot

The cat sat on the mat

The: [0 1 0 0 0 0]

cat: [0 0 1 0 0 0]

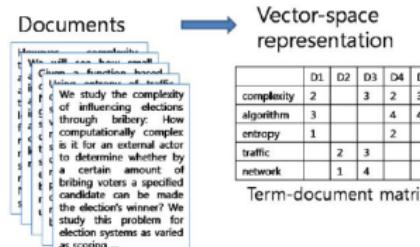
sat: [0 0 0 1 0 0]

on: [0 0 0 0 1 00]

the: [0 0 0 0 0 10]

mat: [0 0 0 0 0 1]

## Term-document matrix



sugar, a sliced lemon, a tablespoonful of their enjoyment. Cautiously she sampled her first well suited to programming on the digital for the purpose of gathering data and

apricot  
pineapple  
computer.  
information  
preserve or jam, a pinch each of, and another fruit whose taste she likened In finding the optimal R-stage policy from necessary for the study authorized in the

	aardvark	computer	data	pinch	result	sugar	...
apricot	0	0	0	1	0	1	
pineapple	0	0	0	1	0	1	
digital	0		2	1	0	1	0
information	0		1	6	0	4	0

## Word-word matrix

# Sparse Vector Representations (PMI)

Raw word frequency is skewed and non-discriminative

$$PMI(word_1, word_2) = \log_2 \frac{P(word_1, word_2)}{P(word_1)P(word_2)} \text{ where } word_1 \neq word_2$$

Positive Pointwise Mutual Information (PPMI) is preferred for its informative context-target word association

Positive PMI (PPMI) between  $word_1$  and  $word_2$

$$PPMI(word_1, word_2) = \max\left(\log_2 \frac{p(word_1, word_2)}{p(word_1)p(word_2)}, 0\right)$$

## Sparse Vector Representations (PMI) Cont'd..

	aardvark	computer	data	pinch	result	sugar
apricot	0	0	0	1	0	1
pineapple	0	0	0	1	0	1
digital	0	2	1	0	1	0
information	0	1	6	0	4	0

$$p_{ij} = \frac{f_{ij}}{\sum_{i=1}^W \sum_{j=1}^C f_{ij}}$$
$$p_{i*} = \frac{\sum_{j=1}^C f_{ij}}{\sum_{i=1}^W \sum_{j=1}^C f_{ij}}$$
$$p_{*j} = \frac{\sum_{i=1}^W f_{ij}}{\sum_{i=1}^W \sum_{j=1}^C f_{ij}}$$
$$pmi_{ij} = \log_2 \frac{p_{ij}}{p_{i*} p_{*j}}$$
$$ppmi_{ij} = \begin{cases} pmi_{ij} & \text{if } pmi_{ij} > 0 \\ 0 & \text{otherwise} \end{cases}$$

# Sparse Vector Representations (PMI) Cont'd..

$$p_{ij} = \frac{f_{ij}}{\sum_{i=1}^W \sum_{j=1}^C f_{ij}}$$

$$p(w_i) = \frac{\sum_{j=1}^C f_{ij}}{N}$$

$$p(c_j) = \frac{\sum_{i=1}^W f_{ij}}{N}$$

$$p(w=\text{information}, c=\text{data}) = 6/19 = .32$$

$$p(w=\text{information}) = 11/19 = .58$$

$$p(c=\text{data}) = 7/19 = .37$$

	Count(w,context)					p(w)
	computer	data	pinch	result	sugar	
apricot	0	0	1	0	1	
pineapple	0	0	1	0	1	
digital	2	1	0	1	0	
information	1	6	0	4	0	
p(w,context)						p(w)
	computer	data	pinch	result	sugar	
apricot	0.00	0.00	0.05	0.00	0.05	0.11
pineapple	0.00	0.00	0.05	0.00	0.05	0.11
digital	0.11	0.05	0.00	0.05	0.00	0.21
information	0.05	0.32	0.00	0.21	0.00	0.58
p(context)	0.16	0.37	0.11	0.26	0.11	

## Sparse Vector Representations (PMI) Cont'd..

		p(w,context)					p(w)
		computer	data	pinch	result	sugar	
$pmi_{ij} = \log_2 \frac{p_{ij}}{p_i * p_j}$	apricot	0.00	0.00	0.05	0.00	0.05	0.11
	pineapple	0.00	0.00	0.05	0.00	0.05	0.11
	digital	0.11	0.05	0.00	0.05	0.00	0.21
	information	0.05	0.32	0.00	0.21	0.00	0.58
p(context)		0.16	0.37	0.11	0.26	0.11	

$$pmi(\text{information}, \text{data}) = \log_2 (.32 / (.37 * .58)) = .58$$

	PPMI(w,context)				
	computer	data	pinch	result	sugar
apricot	-	-	2.25	-	2.25
pineapple	-	-	2.25	-	2.25
digital	1.66	0.00	-	0.00	-
information	0.00	0.57	-	0.47	-

# Distributional Representation - Knowing Words by their Context

C1: A bottle of \_\_\_ is on the table.

C2: Everybody likes \_\_\_.

C3: Don't have \_\_\_ before you drive.

C4: We make \_\_\_ out of corn.

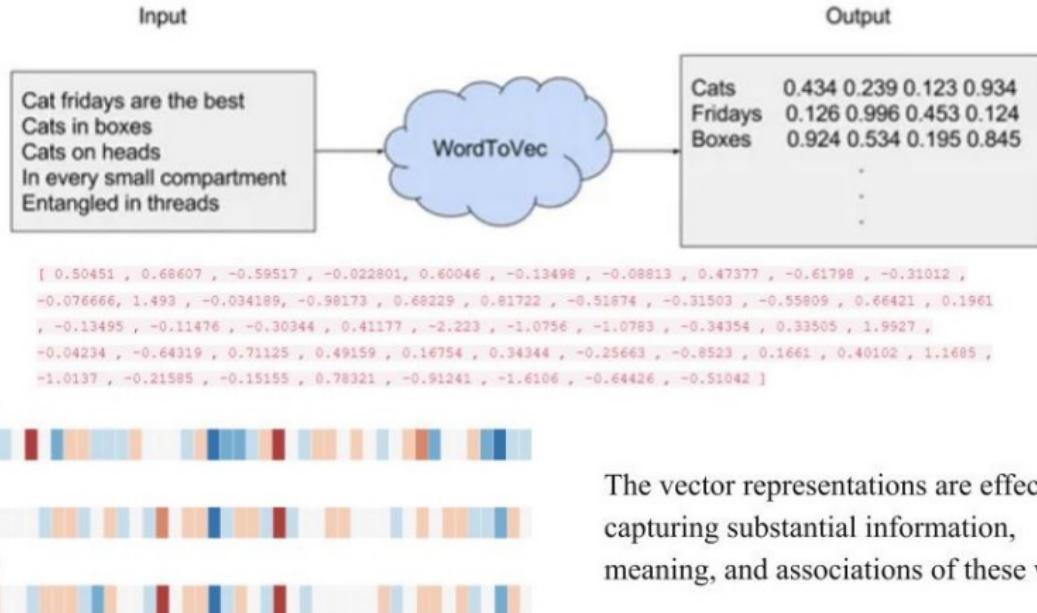
	C1	C2	C3	C4
tejuino	1	1	1	1
loud	0	0	0	0
motor-oil	1	0	0	0
tortillas	0	1	0	1
choices	0	1	0	0
wine	1	1	1	0

“words that occur in similar contexts tend to have similar meanings”

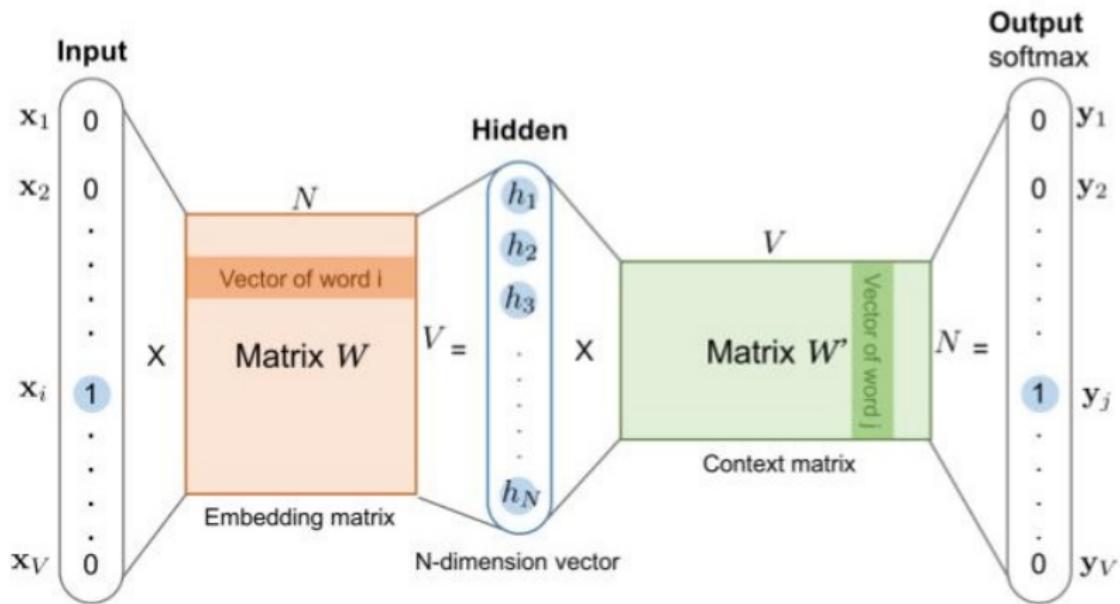
# Dense Vector Representations

- Dense vectors, being shorter and requiring fewer adjustments in machine learning, offer ease in feature utilization.
- Additionally, they excel in capturing nuanced relationships between words like synonyms, unlike sparse representations, allowing better generalization and similarity capture in language models.

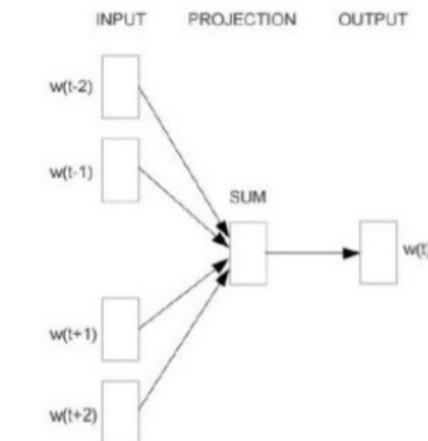
# Word Embedding - Overview of Word2Vec



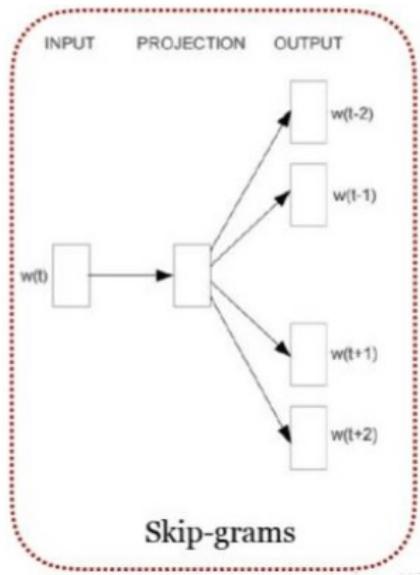
# Overall Skipgram Architecture



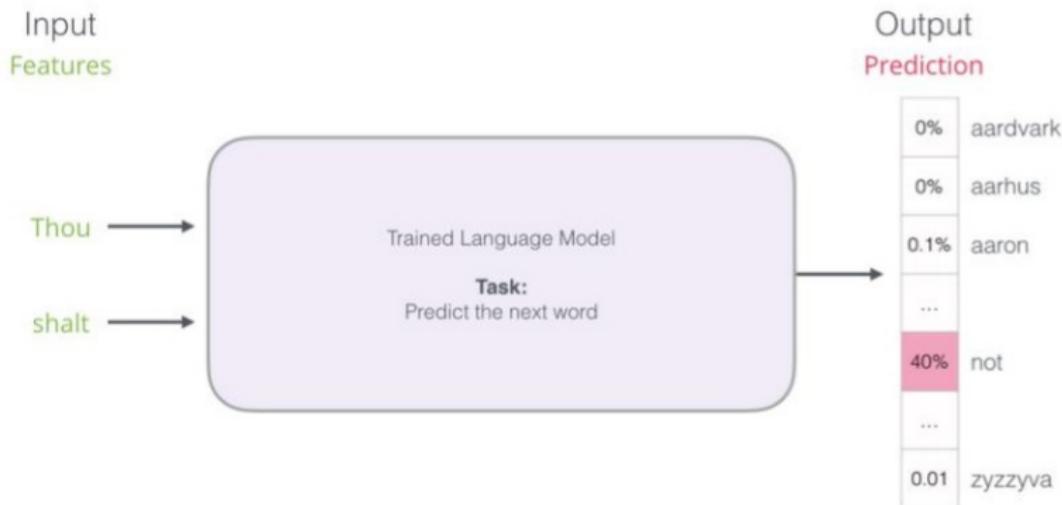
# Dense Vector Representations - Skipgram



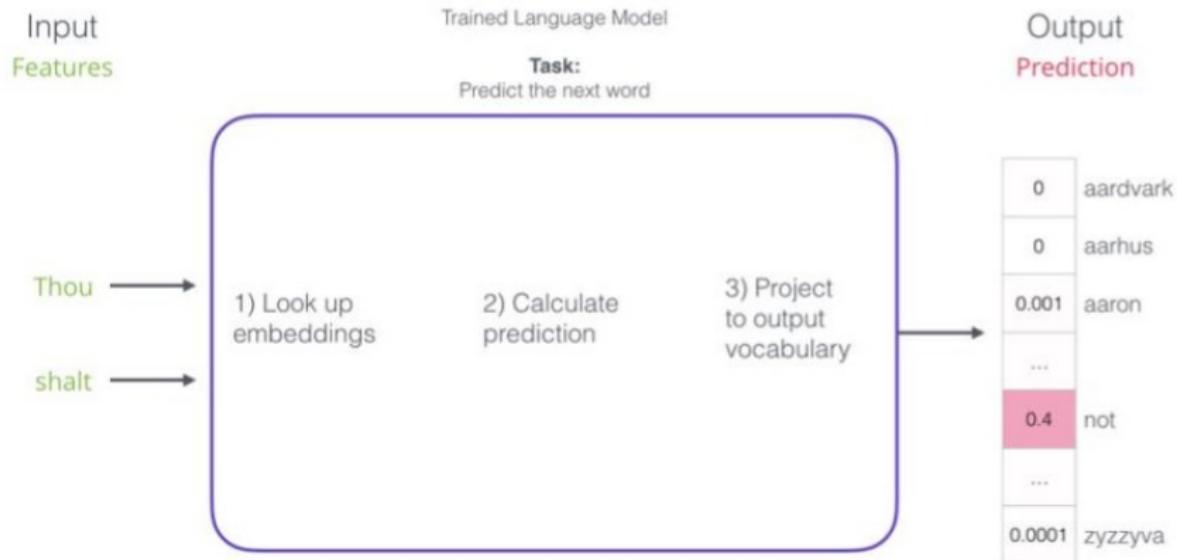
Continuous Bag of Words (CBOW)



# Dense Vector Representations - Language Modeling



# Dense Vector Representations - Language Modeling Cont'd



# Language Modeling - Training Dataset Generation with CBOW

Thou shalt not make a machine in the likeness of a human mind

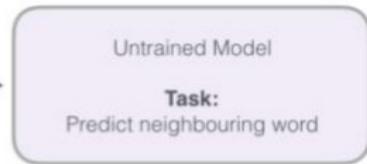
thou	shalt	not	make	a	machine	in	the	...
------	-------	-----	------	---	---------	----	-----	-----

input word	target word
not	thou
not	shalt
not	make
not	a

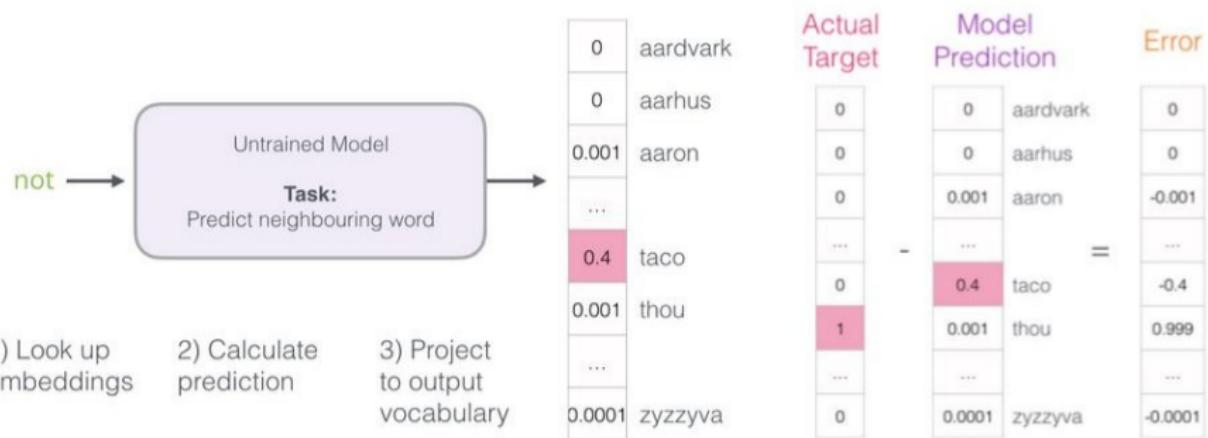
# Language Modeling - Training Skipgram

input word	target word
not	thou
not	shalt
not	make
not	a
make	shalt
make	not
make	a
make	machine
a	not
a	make
a	machine
a	in
machine	make
machine	a
machine	in
machine	the
in	a
in	machine
in	the
in	likeness

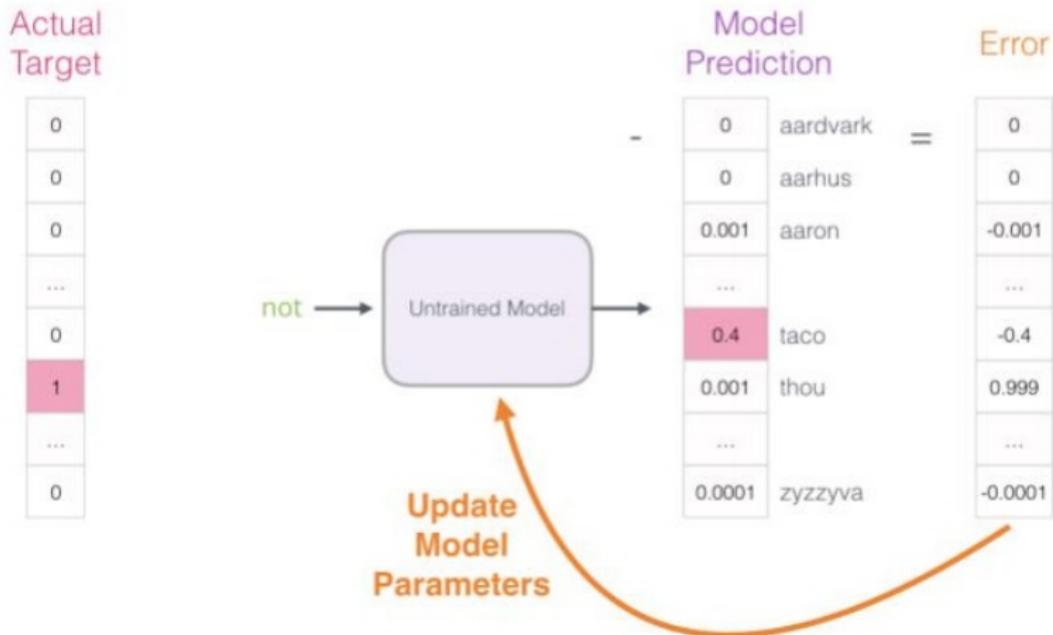
not →



# Language Modeling - Training Skipgram Cont'd

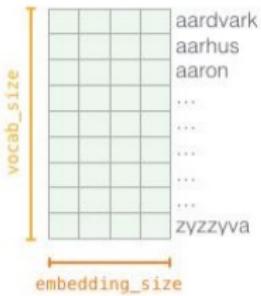


# Language Modeling - Training Skipgram Cont'd



# Word2Vec Training - Skipgram with Negative Sampling

Embedding



Context



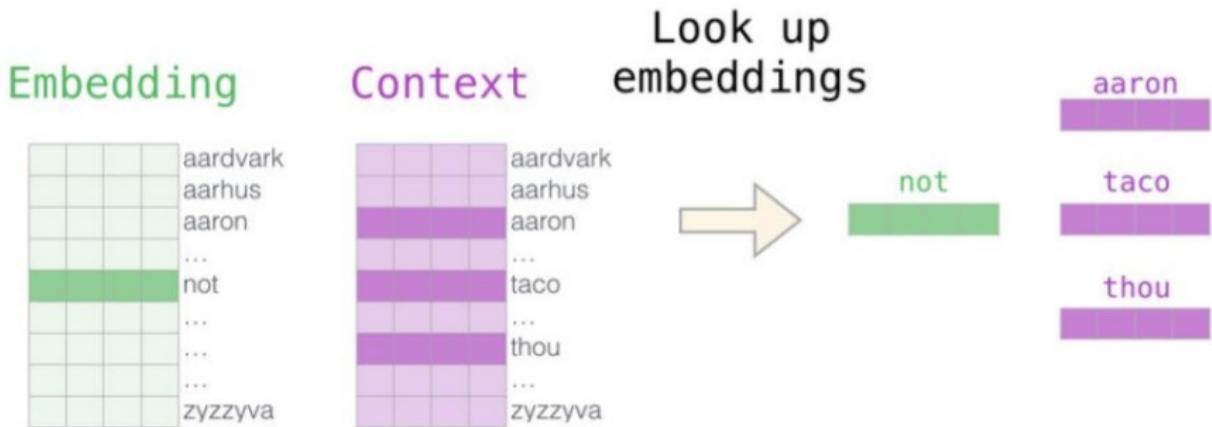
Initialized with random values

dataset

input word	output word	target
not	thou	1
not	aaron	0
not	taco	0
not	shalt	1
not	mango	0
not	finglonger	0
not	make	1
not	plumbus	0
...	...	...

# Word2Vec Training - Skipgram with Negative Sampling

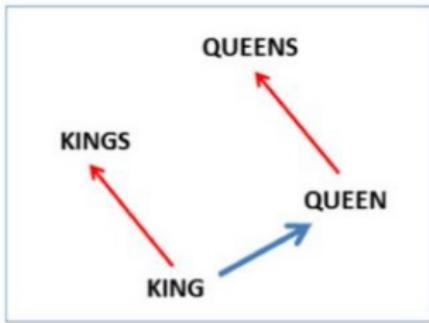
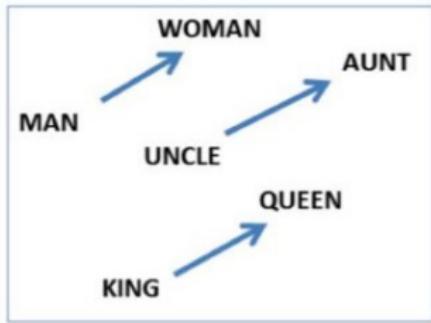
## Cont'd



# Analogy

$\text{vector}(\text{'king'}) - \text{vector}(\text{'man'}) + \text{vector}(\text{'woman'}) \approx \text{vector}(\text{'queen'})$

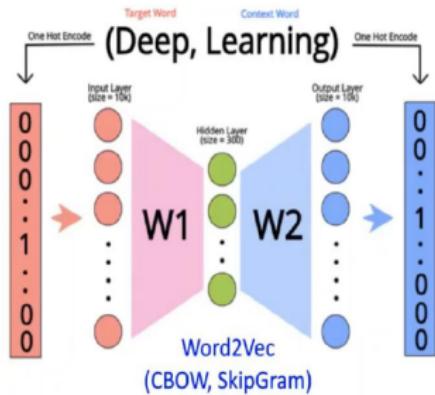
$\text{vector}(\text{'Paris'}) - \text{vector}(\text{'France'}) + \text{vector}(\text{'Italy'}) \approx \text{vector}(\text{'Rome'})$



# Plan of the Tutorial

- 1 Plan of the Tutorial
- 2 Introduction to NLP
- 3 Overview of Distributional Representation Learning for NLP
- 4 Overview of Transformer based Language Model
- 5 Overview of Large Language Models
- 6 Concept of in-context learning and its application
- 7 Conclusion

# Dramatic Entry of Deep Learning!



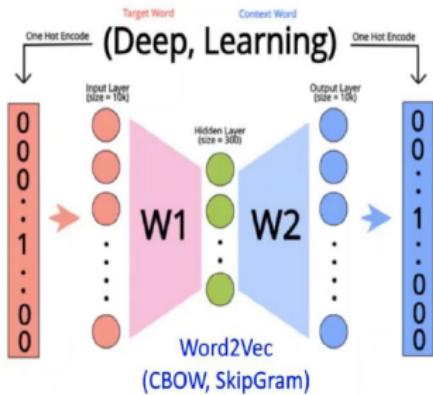
## Advantages

- Unsupervised
- Can work with less data

## Problems

- No shared representations at sub-word levels
- Scaling to new languages requires new embedding matrices
- Positions of tokens are overlooked

# Dramatic Entry of Deep Learning!



## Advantages

- Unsupervised
- Can work with less data

## Problems

- No shared representations at sub-word levels
- Scaling to new languages requires new embedding matrices
- Positions of tokens are overlooked

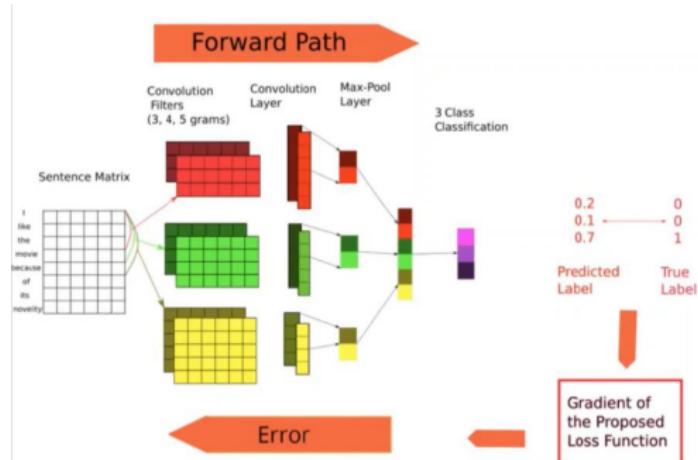
# Inspiration from Computer Vision

## Advantages

- Captures local structures
- Performs well on the classification task
- Very fast (In GPUs)

## Problems

- Can't capture long-range dependencies (words often don't need to be adjacent to be related) in POS tagging, entity extraction, etc.
- Can't capture sequential/temporal information.



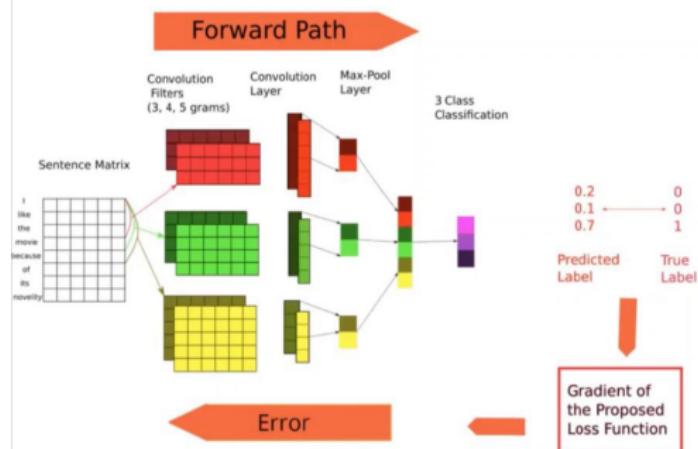
# Inspiration from Computer Vision

## Advantages

- Captures local structures
- Performs well on the classification task
- Very fast (In GPUs)

## Problems

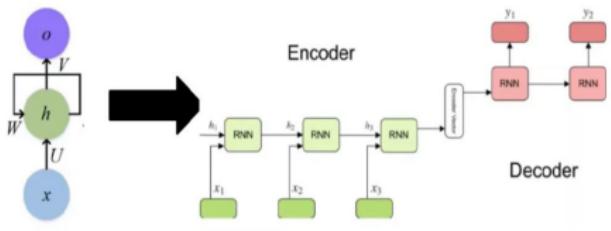
- Can't capture long-range dependencies (words often don't need to be adjacent to be related) in POS tagging, entity extraction, etc.
- Can't capture sequential/temporal information.



# Seq2Seq Model

## Advantages

- Captures sequential structures
- Process inputs of any length
- Even if the input sizes increases, the model size remains same
- Weights can be shared across the time steps



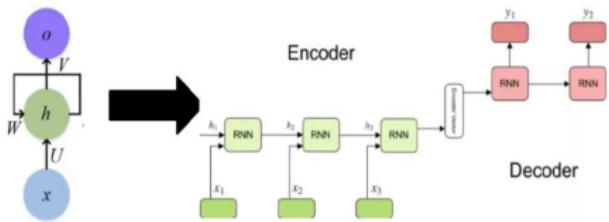
## Problems

- Dealing with long-range dependencies
- The sequential nature of the model prevents parallelization (slow to train)

# Seq2Seq Model

## Advantages

- Captures sequential structures
- Process inputs of any length
- Even if the input sizes increases, the model size remains same
- Weights can be shared across the time steps

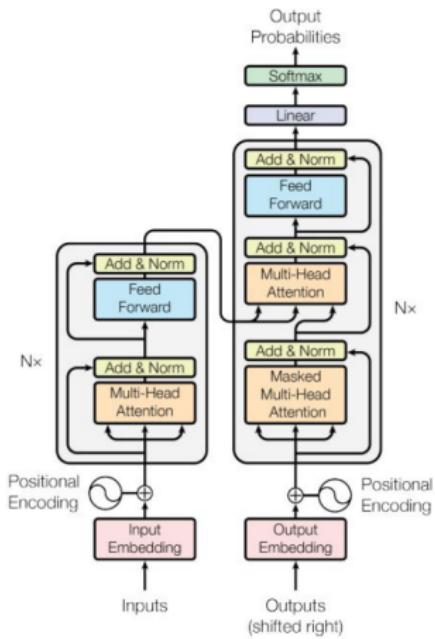


## Problems

- Dealing with long-range dependencies
- The sequential nature of the model prevents parallelization (slow to train)

# Transformer

- No Convolutions or recurrence
- Easy to Parallelize than recurrent network.
- Captures more long-range Dependencies.



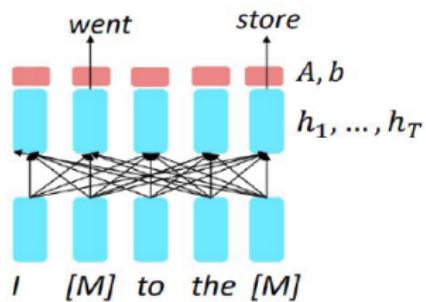
# The Uprising of Language Models

- After the invention of the Transformer in the year 2017. The language model comes in the post-2017 era.
- Early 2018 era most of the language models follow the paradigm of **Pretrain** → **Finetune** → **Predict**.
- Encoder Only
- Encoder- Decoder
- Decoder Only



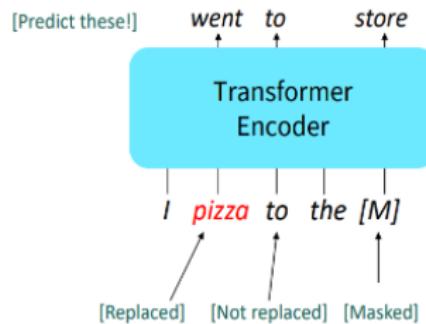
# Pretraining Encoder

- Idea: replace some fraction of words in the input with a special [MASK] token; predict these words.
- Only add loss terms from words that are “masked out.”



# BERT: Bidirectional Encoder Representations from Transformers

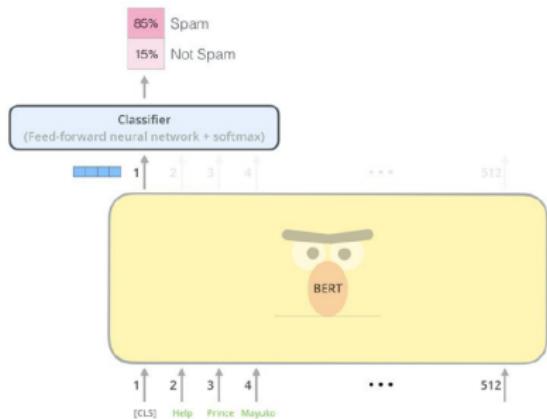
- Devlin et al., 2018 proposed the “Masked LM” objective and released the weights of a pretrained Transformer, a model they labeled BERT.
- Predict a random 15% of word tokens.
  - Replace input word with [MASK] 80% of the time.
  - Replace input word with a random token 10% of the time
  - Leave input word unchanged 10% of the time
- Trained on Bookcorpus (800M words) and English Wikipedia (2500M words)



# Finetuning BERT

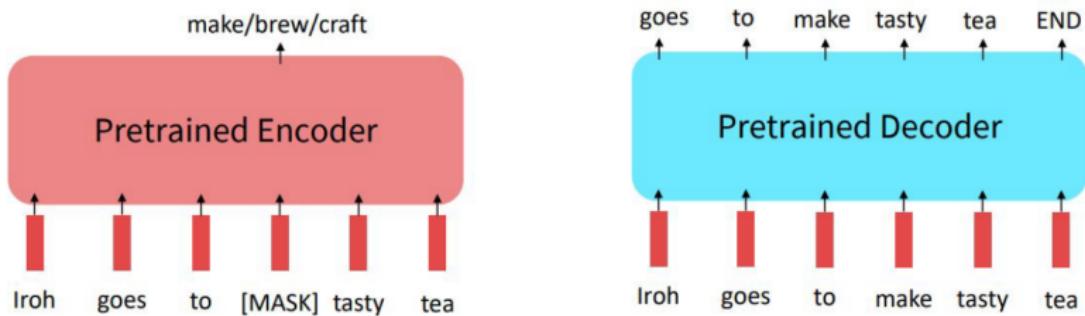
## Input Features

Help Prince Mayuko Transfer  
Huge Inheritance



# Limitations of pretrained encoders

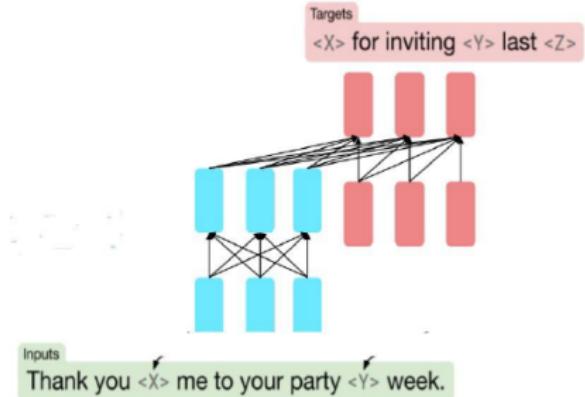
- If our tasks involves generating sequences, consider using a pretrained decoder, BERT, and other pretrained encoders don't naturally lead to nice autoregressive generation methods.



# Pretraining encoder-decoders

- For encoder-decoders, we could do something like language modeling, but where a prefix of every input is provided to the encoder and is not predicted.
- This is implemented in text preprocessing: It's still an objective that looks like language modeling on the decoder side.

Original text  
Thank you for inviting me to your party last week.

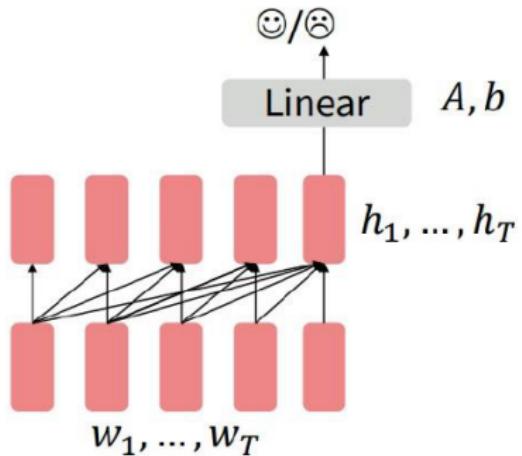


# Plan of the Tutorial

- 1 Plan of the Tutorial
- 2 Introduction to NLP
- 3 Overview of Distributional Representation Learning for NLP
- 4 Overview of Transformer based Language Model
- 5 Overview of Large Language Models**
- 6 Concept of in-context learning and its application
- 7 Conclusion

## Pretraining decoders

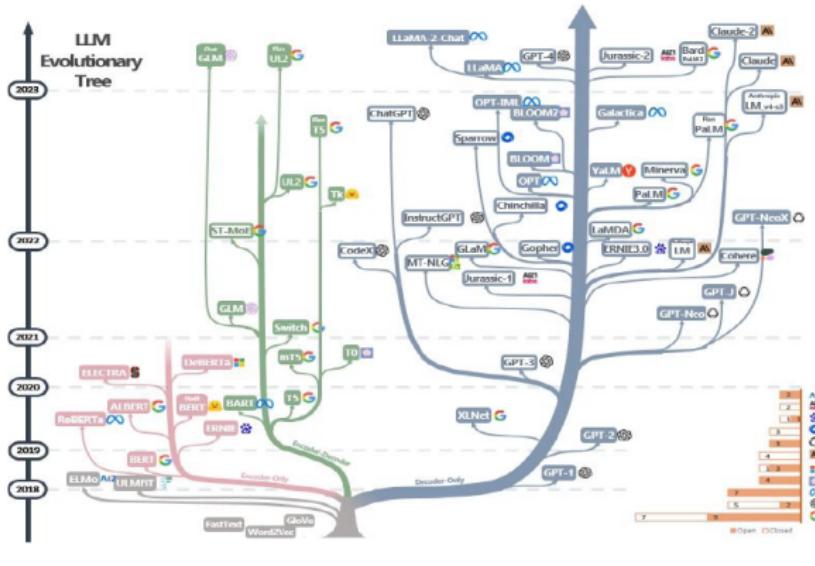
- We can finetune them by training a classifier on the last word's hidden state.



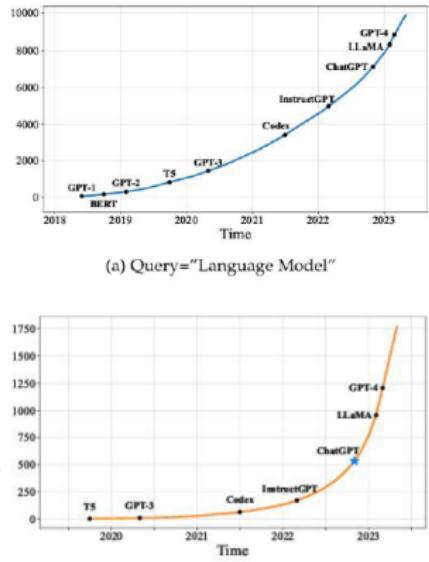
$$h_1, \dots, h_T = \text{Decoder}(w_1, \dots, w_T)$$
$$y \sim Ah_T + b$$

Where  $A$  and  $b$  are randomly initialized and specified by the downstream task.

# LLM Evolution Tree



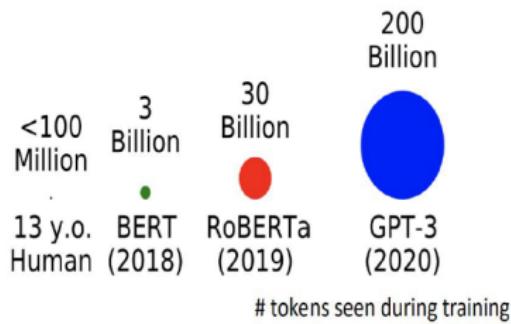
(a) Query="Language Model"



(b) Query="Large Language Model"

# Bigger the Better

- GPT (117M parameters): Transformer decoder with 12 layers. Trained on BooksCorpus: over 7000 unique books (4.6GB text).
- Showed that language modeling at scale can be an effective pretraining technique for downstream tasks like natural language inference.
- GPT-2 (1.5B parameters): Same architecture as GPT, just bigger (117M → 1.5B). Trained on much more data: 4GB → 40GB of internet text data
- GPT-3 (175B parameters): Same architecture as GPT, just bigger (1.5B → 175B). Trained on much more data: (40GB → over 600GB) of internet text data.



## Prompt → Predict

GPT3 shows an unique ability!

- Perform the task by generating natural language tokens, aka, prompting

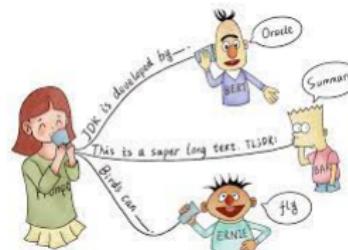
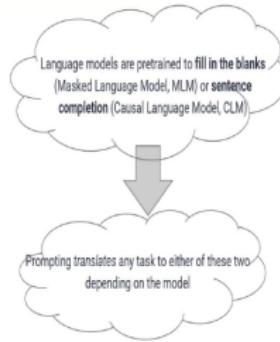
That movie was great. Sentiment: Positive

It was a horrible day! Sentiment: Negative

This is an absolute mess. Sentiment:



# Prompting



# Example

Language models may do rudimentary modeling of agents, beliefs, and actions:



**You**

Pat watches demonstration of a bowling ball and a leaf being dropped at the same time in a vacuum chamber. If Pat is a Physicist, what he will predict?

< 1 / 2 >



**ChatGPT**

As a physicist, Pat would predict that in a vacuum chamber where air resistance is eliminated, both the bowling ball and the leaf would fall at the same rate, accelerating equally due to gravity. This concept is known as the equivalence principle, famously demonstrated by Galileo and later confirmed by Newton's laws of motion.

# Example

SP

## You

Pat watches demonstration of a bowling ball and a leaf being dropped at the same time in a vacuum chamber. If Pat who has never seen this demonstration before, what he will predict?

< 2 / 2 >



## ChatGPT

If Pat has never seen the demonstration before and is unaware of the concept of the equivalence principle (which states that in a vacuum, all objects fall at the same rate regardless of their mass), they might predict that the bowling ball would fall faster than the leaf due to its larger size or mass. This prediction is based on everyday observations where heavier objects often seem to fall faster than lighter ones. However, in a vacuum chamber where air resistance is eliminated, both the bowling ball and the leaf would fall at the same rate, defying Pat's initial prediction.



# Example .. Code

SP

You

Write a code in Python to arrange a sequence of numbers in ascending order.

< 2 / 2 >

ChatGPT

Certainly! You can use the `sorted()` function in Python to arrange a sequence of numbers in ascending order. Here's an example:

```
python Copy code

# Define a sequence of numbers
numbers = [9, 4, 7, 2, 11, 5]

# Arrange the sequence in ascending order
ascending_order = sorted(numbers)

# Display the sequence in ascending order
print("Numbers in ascending order:", ascending_order)
```

Replace the `numbers` list with your desired sequence of numbers. When you run this code, it will output the sequence arranged in ascending order.

# Example .. mathematical reasoning

SP

You

What is the volume of a cube with edge  $2a$ ?



ChatGPT

The volume of a cube is calculated by cubing the length of one of its edges.

So, if the edge of the cube is  $2a$ , the volume would be:

$$\text{Volume of cube} = (\text{Edge length})^3 = (2a)^3 = 8a^3$$

Therefore, the volume of the cube with an edge length of  $2a$  is  $8a^3$ .



# Emergent few-shot learning

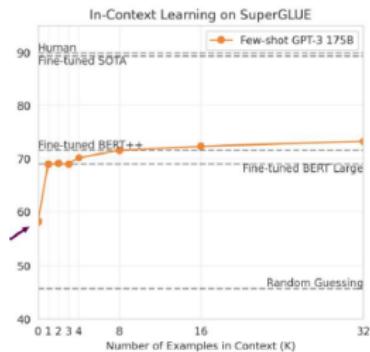
- Specify a task by simply prepending examples of the task before your example
- no gradient updates are performed

1 gaot => goat  
2 sakne => snake  
3 brid => bird  
4 fsih => fish  
5 dcuk => duck  
6 cmihp => chimp

In-context learning

1 thanks => merci  
2 hello => bonjour  
3 mint => menthe  
4 wall => mur  
5 otter => loutre  
6 bread => pain

In-context learning



# Example

## Zero-Shot Learning

Classify the given text using Sentiment Analysis.  
Possible classes: POSITIVE, NEGATIVE, NEUTRAL.

Input: "I watched La La Land... yeah it was OK I guess"  
Class:

## Few-Shot Learning

Classify the given text using Sentiment Analysis.  
Possible classes: POSITIVE, NEGATIVE, NEUTRAL.

Input: "Unbelievable, I wasted 2h on 'The Room'!"  
Class: NEGATIVE

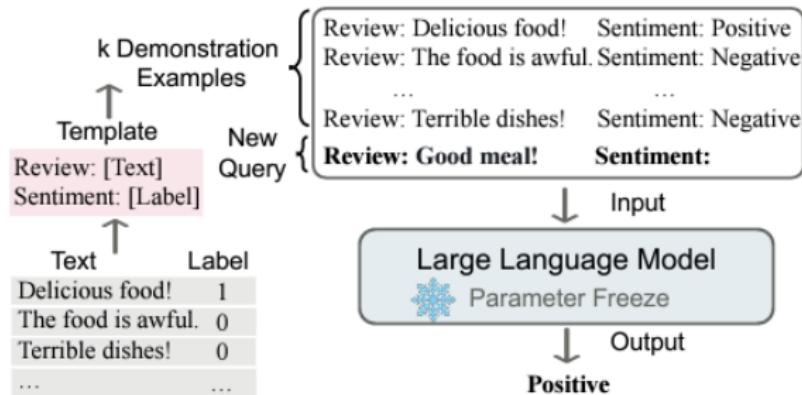
Input: "Dr Strangelove is my favorite movie <3 <3"  
Class: POSITIVE

Input: "I watched La La Land... yeah it was OK I guess"  
Class:

# Plan of the Tutorial

- 1 Plan of the Tutorial
- 2 Introduction to NLP
- 3 Overview of Distributional Representation Learning for NLP
- 4 Overview of Transformer based Language Model
- 5 Overview of Large Language Models
- 6 Concept of in-context learning and its application
- 7 Conclusion

# An brief overview on In-Context Learning (ICL)



ICL does not update parameters and instead directly makes predictions using the pretrained language model.

# In-Context Learning Approaches

Now let us explore some popular ICL methods.

- Chain of thought (COT)
- Self-consistency COT

# Chain-of-thought prompting

## Standard Prompting

### Model Input

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: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

### Model Output

A: The answer is 27. 

## Chain-of-Thought Prompting

### Model Input

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?

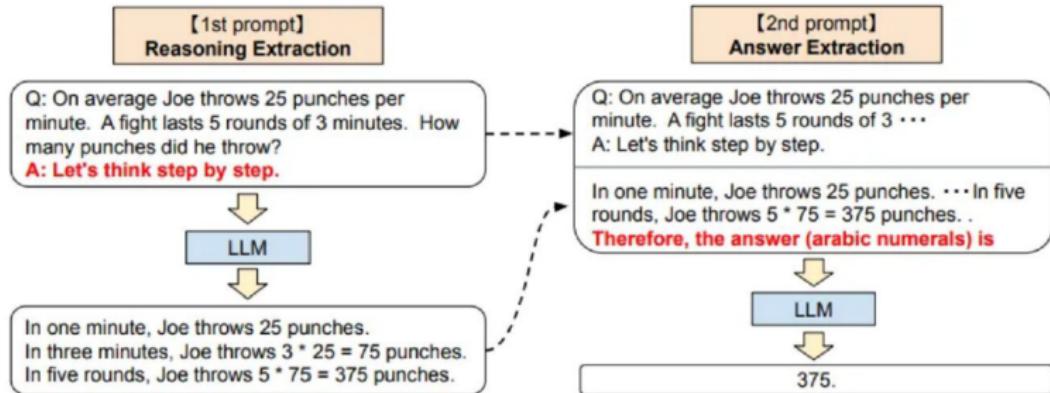
A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls.  $5 + 6 = 11$ . The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

### Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had  $23 - 20 = 3$ . They bought 6 more apples, so they have  $3 + 6 = 9$ . The answer is 9. 

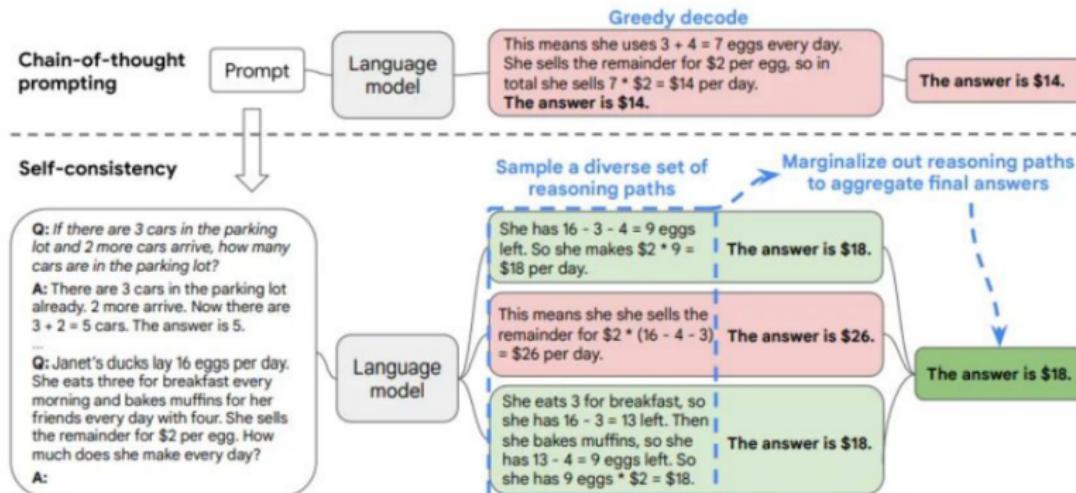
# Dive deep into chain of thought (COT)



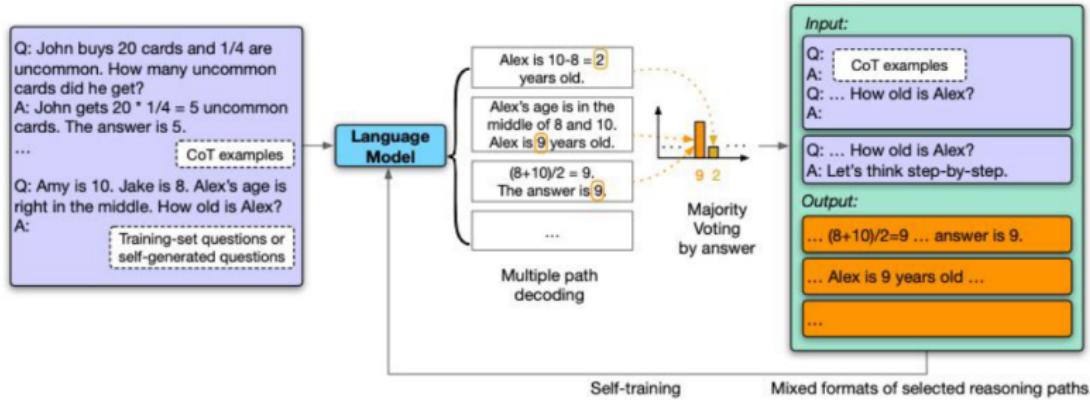
The process involves two steps: first “reasoning prompt extraction” to extract a full reasoning path from a language model, and then use the second “answer prompt extraction” to extract the answer in the correct format from the reasoning text.

# Self-consistency COT

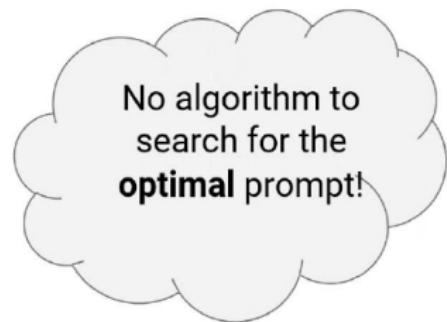
Greedy decoding in COT is replaced by another decoding strategy used in COT prompting named self-consistency COT.



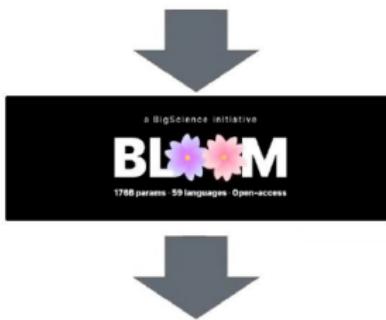
# Self-consistency COT



# Optimal Prompts : Open Problem



Input: That movie was really great if you are dumb beyond imagination.  
Sentiment expressed in the above sentence is



Input: That movie was really great if you are dumb beyond imagination.  
Sentiment expressed in the above sentence is **on the positive side of the range**.

# Optimal Prompts : Open Problem

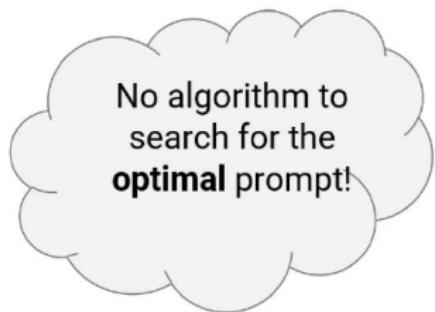


Consider the following sentence: "That movie was ~~really~~ great if you are dumb beyond imagination." The author's opinion expressed towards the movie in this sentence is



Consider the following sentence: "That movie was really great if you are dumb beyond imagination." The author's opinion expressed towards the movie in this sentence is **neutral**.

# Optimal Prompts : Open Problem



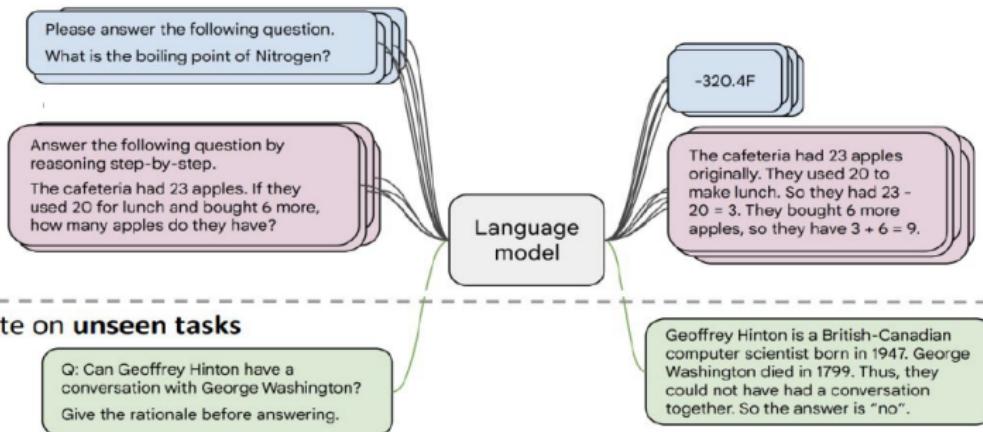
Consider the following sentence: "That movie was really great if you are dumb beyond imagination." In the author's opinion, the movie is



Consider the following sentence: "That movie was really great if you are dumb beyond imagination." In the author's opinion, the movie is **not good**.

# Instruction Fine Tuning

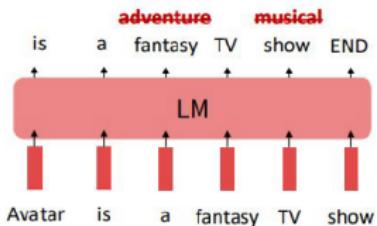
Collect examples of (instruction, output) pairs across many tasks and finetune an LM



Evaluate on **unseen** tasks

# Limitations of instruction finetuning

- One limitation of instruction fine tuning is obvious: it's expensive to collect ground truth data for tasks.
- Problem 1: tasks like open-ended creative generation have no right answer.
- Problem 2: language modeling penalizes all token-level mistakes equally, but some errors are worse than others.
- Can we explicitly attempt to satisfy human preferences?



# Plan of the Tutorial

- 1 Plan of the Tutorial
- 2 Introduction to NLP
- 3 Overview of Distributional Representation Learning for NLP
- 4 Overview of Transformer based Language Model
- 5 Overview of Large Language Models
- 6 Concept of in-context learning and its application
- 7 Conclusion

# Conclusion

- The tutorial has provided a comprehensive understanding of the key concepts and principles that form the foundation of large language models.
- From the basics of natural language processing to the inner workings of advanced language models, the audience has gained a solid grounding in the field.
- Hands-on exercises and examples have equipped the audience with practical skills, allowing them to implement and integrate these models into real-world projects.

## Reference I

- [1] Kenton, J. & Toutanova, L. Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings Of NaacL-HLT.* **1** pp. 2 (2019)
- [2] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, Ł. & Polosukhin, I. Attention is all you need. *Advances In Neural Information Processing Systems.* **30** (2017)
- [3] Mikolov, T., Chen, K., Corrado, G. & Dean, J. Efficient estimation of word representations in vector space. *ArXiv Preprint ArXiv:1301.3781.* (2013)
- [4] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I. & Others Language models are unsupervised multitask learners. *OpenAI Blog.* **1**, 9 (2019)

## Reference II

- [5] McGuffie, K. & Newhouse, A. The radicalization risks of GPT-3 and advanced neural language models. *ArXiv Preprint ArXiv:2009.06807.* (2020)
- [6] Pennington, J., Socher, R. & Manning, C. Glove: Global vectors for word representation. *Proceedings Of The 2014 Conference On Empirical Methods In Natural Language Processing (EMNLP).* pp. 1532-1543 (2014)

# *Thank You!*

**Hands on & Demo:** <https://github.com/payelsantra/FIRE2023tutorial>