

# Applied Natural Language Processing (BUAN 6342)

## Lecture 1 Introduction/Overview

Harpreet Singh

University of Texas at Dallas

Spring 2024



# Agenda

---

Introduction

---

Overview of NLP

---

Course Overview

---

Pre-Processing

---

Break

---

Spacy Intro

---

Sparse Embedding



# Me...

---

---

Office location: JSOM 3.430

---

Email: [harpreet@utdallas.edu](mailto:harpreet@utdallas.edu)

---

Emails will be answered within three working days

---

If you send email to [harpreet@utdallas.edu](mailto:harpreet@utdallas.edu): start with “BUAN6342” in the subject title

---

Office hour: By appointment

---

**Student emails and messages will be answered within 3 working days under normal circumstances.**



# You...

---

Please bring your name tags from next class



# TA

---

## Helper

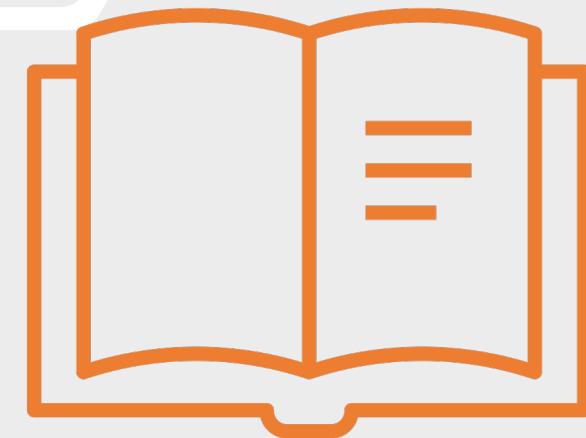
- Debapratim Ghosh ([debapratim.ghosh@utdallas.edu](mailto:debapratim.ghosh@utdallas.edu))
- Contact him for any kind of troubleshooting/doubts
- He will also have dedicated office hours



## Grader

- Shritej Shrikant Chavan  
([shritejshrikant.chavan@utdallas.edu](mailto:shritejshrikant.chavan@utdallas.edu))
- He is responsible for grading
- Contact him only for grading related questions

# Overview of Natural Language Processing (NLP)



# What is NLP

---

Natural language processing (NLP) is a collective term referring to automatic computational processing of human languages.



# Search/Question Answering

what is deep learning



All Videos Images Books News More Tools

About 1,340,000,000 results (0.83 seconds)

<https://www.ibm.com> › Cloud › Cloud Learn

## What is Deep Learning? - IBM

May 1, 2020 — Deep learning is a subset of machine learning, which is essentially a neural network with three or more layers. These neural networks attempt to ...

[What is deep learning?](#) · Deep learning vs. machine... · Deep learning applications

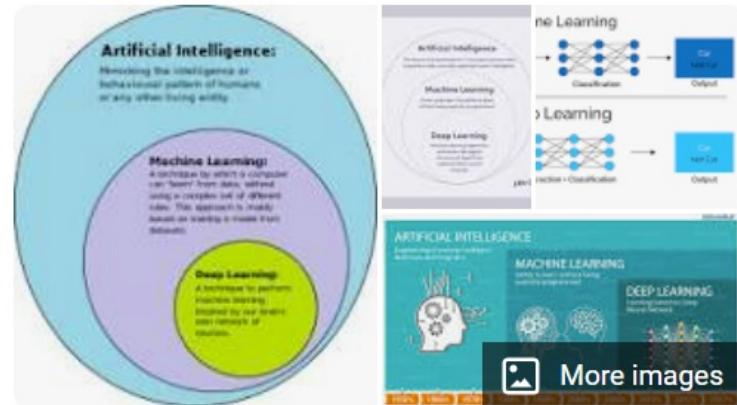
<https://www.techtarget.com> › searchenterpriseai › definition

## What is Deep Learning and How Does It Work? - TechTarget

Deep learning is a type of machine learning and artificial intelligence (AI) that imitates the way humans gain certain types of knowledge. Deep learning is ...

[An Error Occurred](#) · [How Deep Learning Works](#) · Deep Learning Methods

You visited this page on 8/19/22.



Deep learning is part of a broader family of machine learning methods based on artificial neural networks with representation learning. Learning can be supervised, semi-supervised or unsupervised.



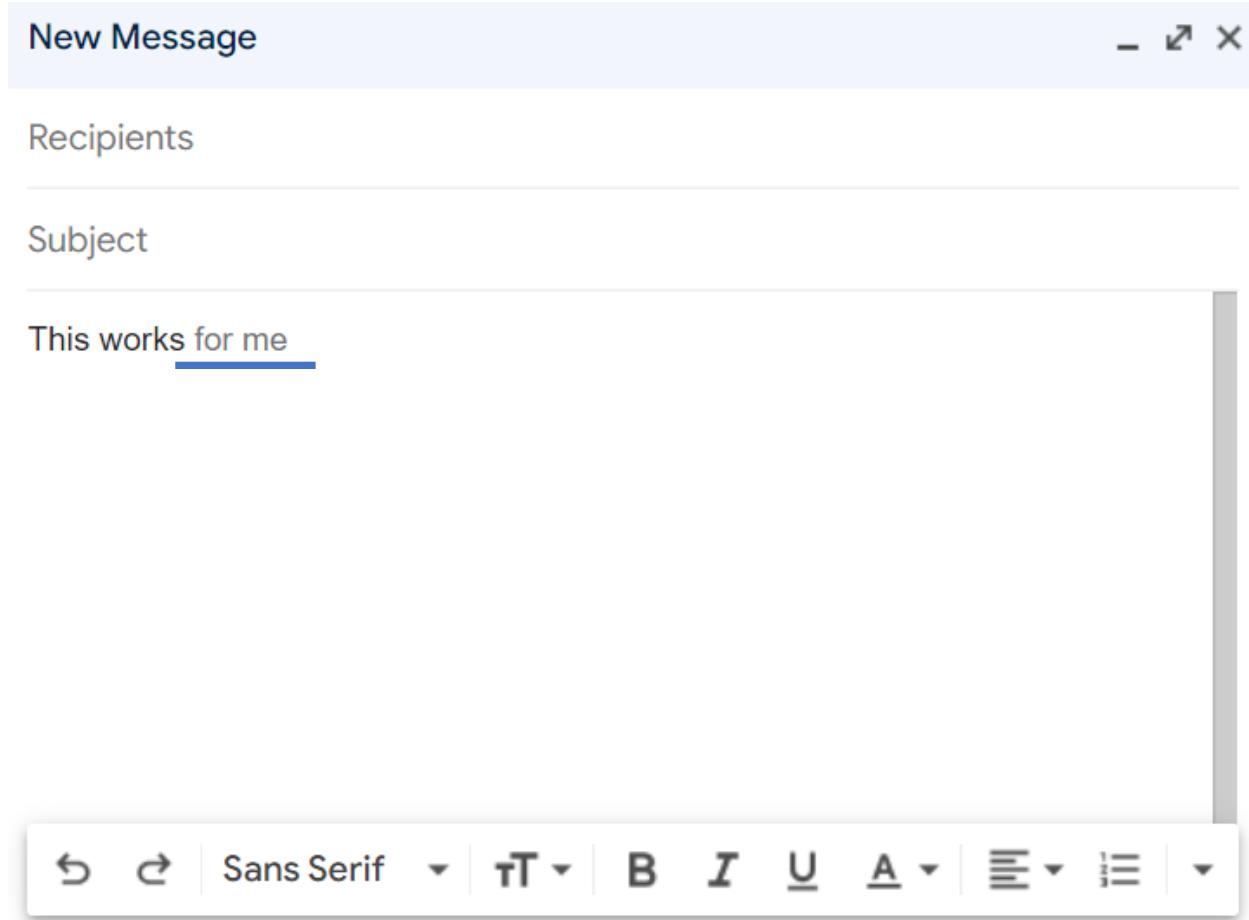
# Machine Translation

English ▾ ↔ Hindi ▾

UTD is a good university × यूटीडी एक अच्छा विश्वविद्यालय है  
yooteedee ek achchha vishvavidyaalay hai



# Predictive Text



# Grammar Checking/paraphrasing



If you have any questions about my rate or if you find it necessary to increase or decrease the scope for this project, please let me know.

Clarity · Rephrase sentence

If you have any questions about my rate or **if you** find it necessary to increase or decrease **the scope** **for this-project****project's scope**, please let me know.

Rephrase

Dismiss



# NLP is everywhere

---

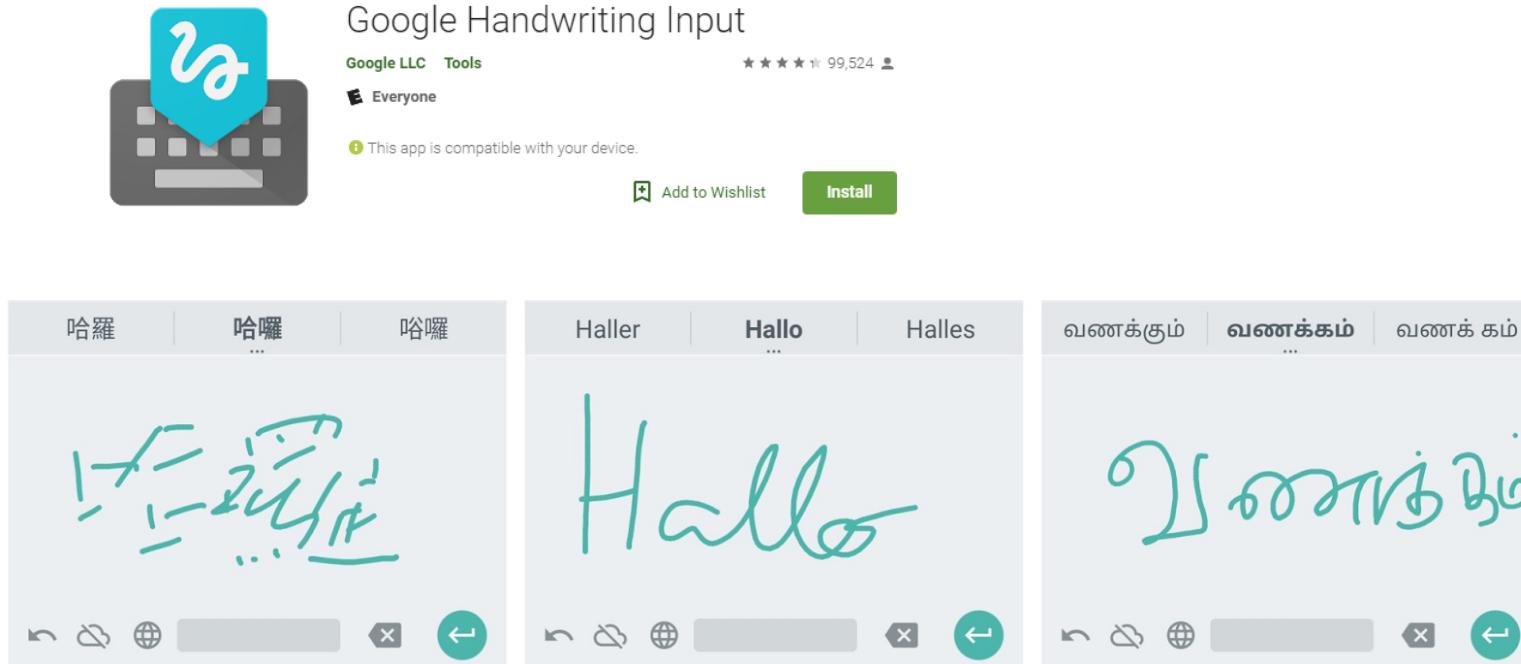
“Alexa, who is the president  
of USA?”



# NLP is everywhere

---

Learning to recognize handwriting



# Reasoning Tasks

---

Pathways Language Model (PaLM): Scaling to 540 Billion Parameters for Breakthrough Performance

Monday, April 4, 2022

## Explaining a joke

### Prompt

Explain this joke:

Joke: Did you see that Google just hired an eloquent whale for their TPU team? It showed them how to communicate between two different pods!

### Model Response

Prediction: TPUs are a type of computer chip that Google uses for deep learning. A "pod" is a group of TPUs. A "pod" is also a group of whales. The joke is that the whale is able to communicate between two groups of whales, but the speaker is pretending that the whale is able to communicate between two groups of TPUs.



The latest from Google Research

# ChatGPT

---



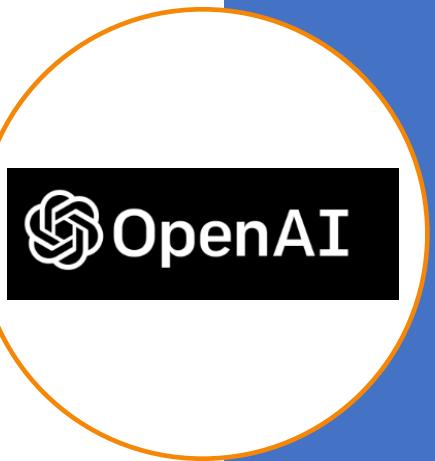
ChatGPT is a variant of GPT (Generative Pre-trained Transformer) that is specifically designed for conversational AI tasks such as dialogue generation and response generation.



It builds on the original GPT model by fine-tuning it on conversational data, which allows it to generate more natural and contextually relevant responses.



Additionally, ChatGPT may also include additional task-specific architectures and training procedures to further improve its performance on conversational AI tasks.



Let us try it out: <https://openai.com/blog/chatgpt/>

# Core Tasks

- Spell Checking
- Keyword Based Information Retrieval
- Topic Modeling
- Text Classification (e.g. Sentiment Analysis)
- Information Extraction
- Closed Domain Conversational Agents
- Text Summarization
- Question Answering
- Machine Translation
- Open Domain Conversation Agent

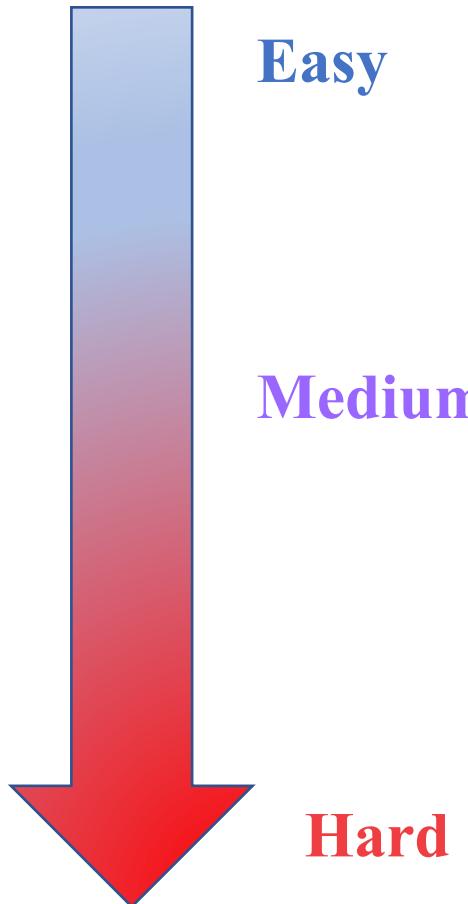
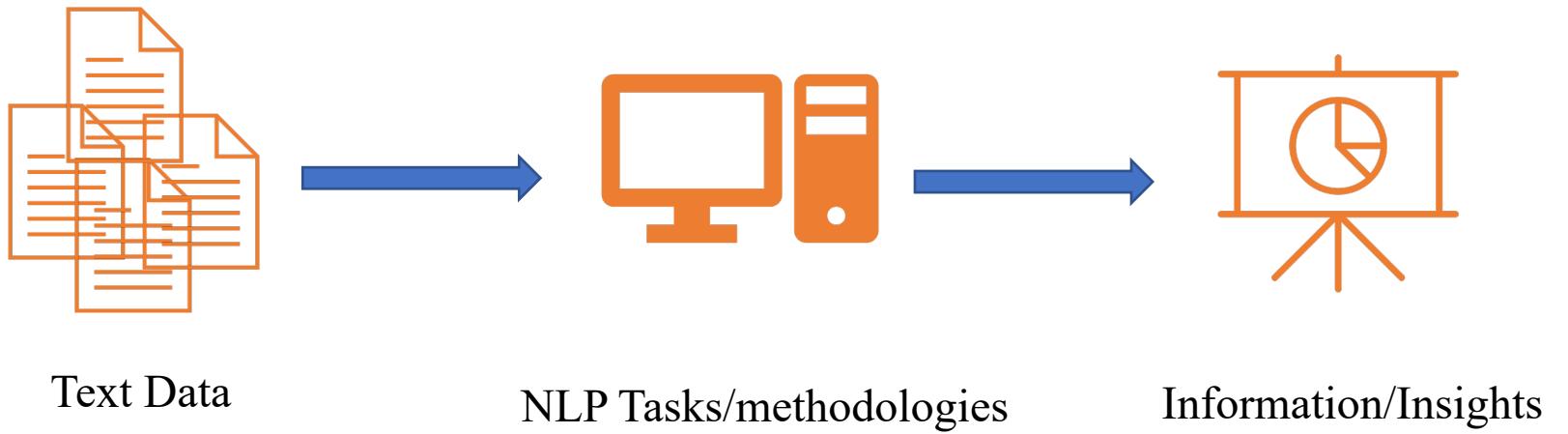


Figure adapted from the book: <https://www.amazon.com/Practical-Natural-Language-Processing-Pragmatic/dp/1492054054>

# Applied NLP



# What makes Language Hard?

---

---

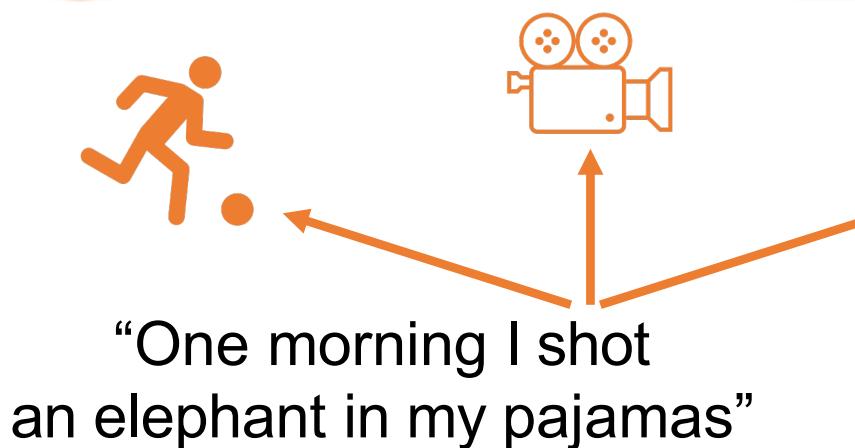
Language is a complex social process

---

Tremendous ambiguity at every level of representation



# Ambiguity



"One morning I shot  
an elephant in my pajamas"



# NLP is interdisciplinary

---

---

Artificial Intelligence

---

Machine Learning (ca. 2000—today); statistical models, neural networks

---

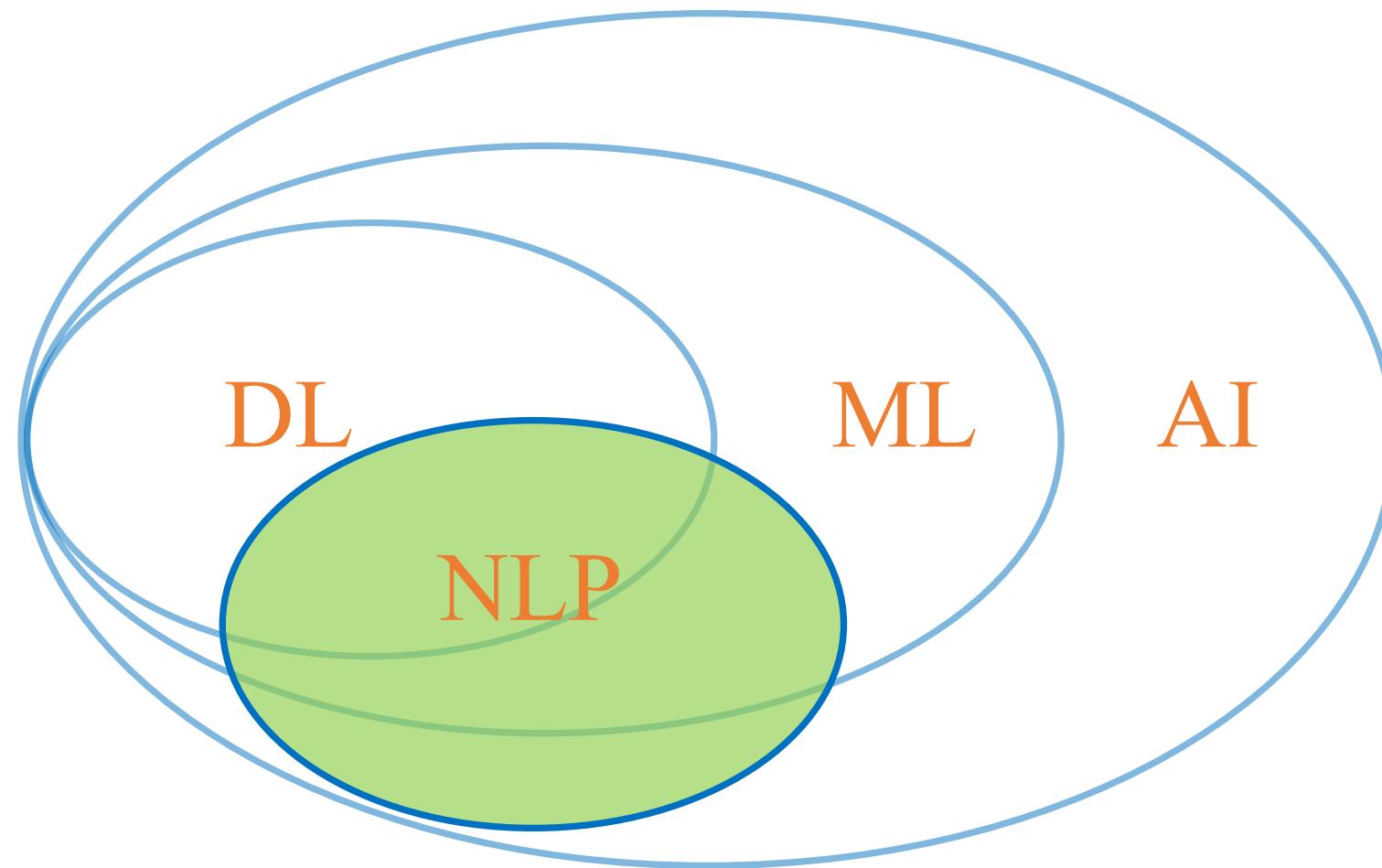
Linguistics (representation of Language)

---

Social sciences/humanities(models of language at use in culture/society)



# Overlap

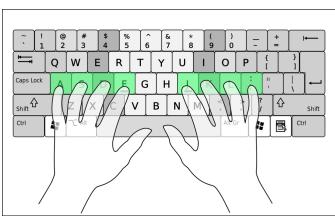


# DL Vs. ML

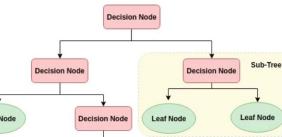
## Machine Learning



Input



Manual Feature Extraction



Classification



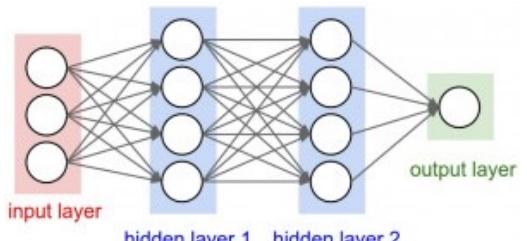
Sentiment



Output



Input



Feature Extraction + Classification

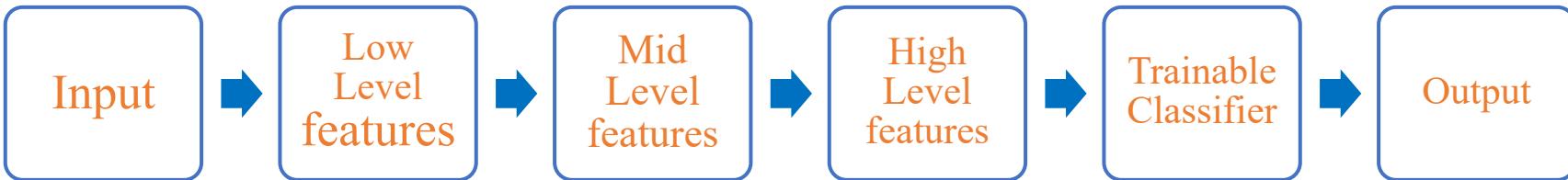


Sentiment

Output

# What is Deep Learning (DL)?

## Deep Learning is Hierarchical Feature Learning

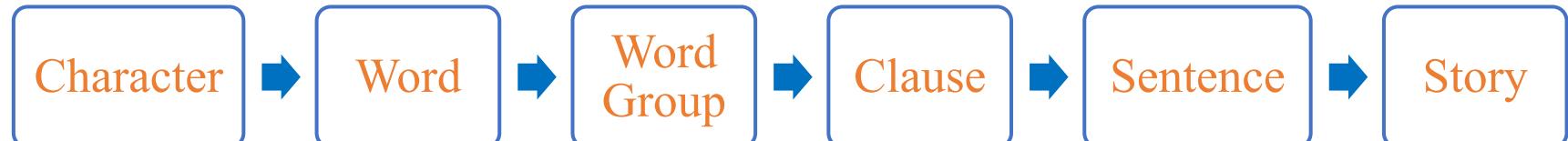


Hierarchy of representations with increasing level of abstraction

- Image Recognition



- Text

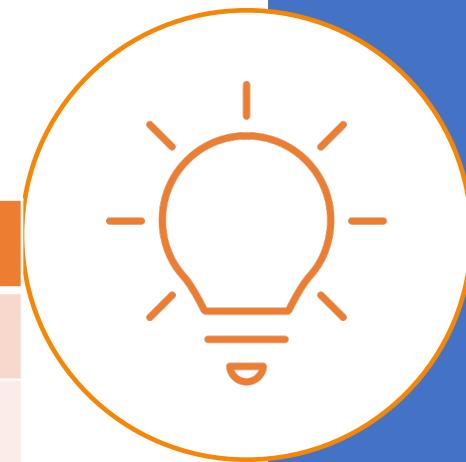


# Hardware

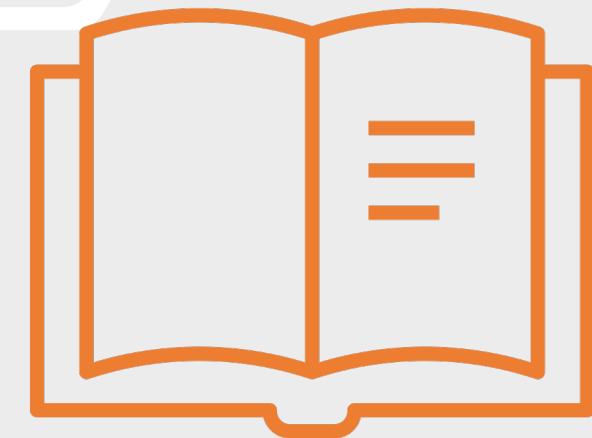
$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \text{ Input matrix of } 6 \times 6 \quad * \quad \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \text{ Kernel of } 3 \times 3 \quad = \quad \begin{pmatrix} 3 & 0 & 2 & 2 \\ 0 & 3 & 2 & 3 \\ 3 & 1 & 5 & 4 \\ 0 & 3 & 2 & 3 \end{pmatrix} \text{ Activation map}$$

- Numerous matrix computations are required for deep learning.

CPU	GPU/TPU
Few complex cores	Hunderd of simpler cores
Focuses on accomplishing one task in the most effective manner feasible	Focuses on performing multiple tasks simultaneously.
Used for general purpose tasks	Used for graphics processing or matrix multiplication
<i>Intel® Core™ i7-10510U Processor</i> - 4 cores	NVIDIA GTX 1080 Ti - 3584 cores



# Why should you take NLP Course ?



# Why should you take NLP Course?

Figure 2. AI & ML VC DEAL ACTIVITY

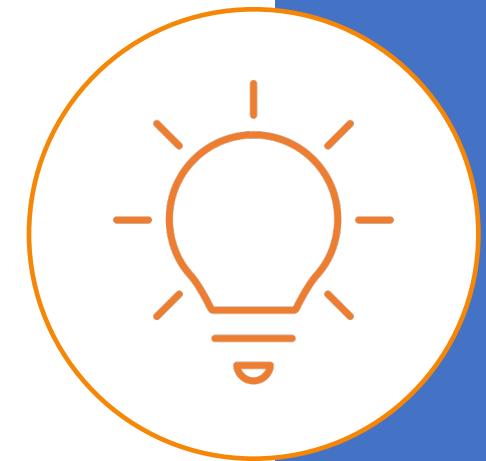
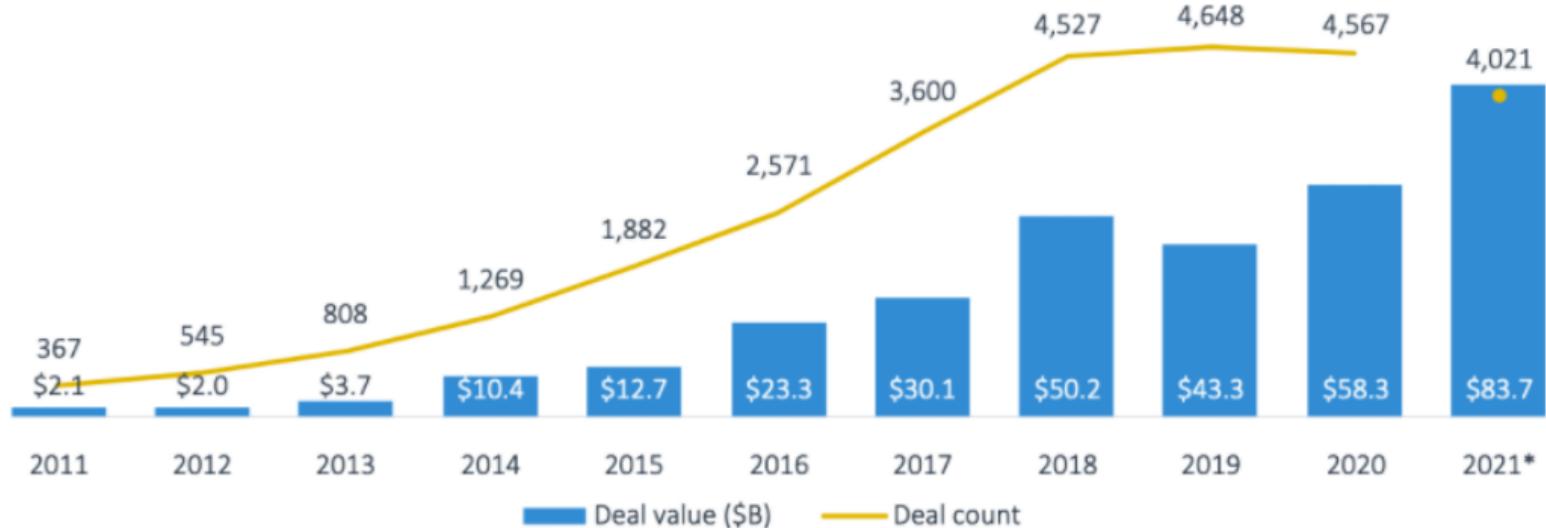
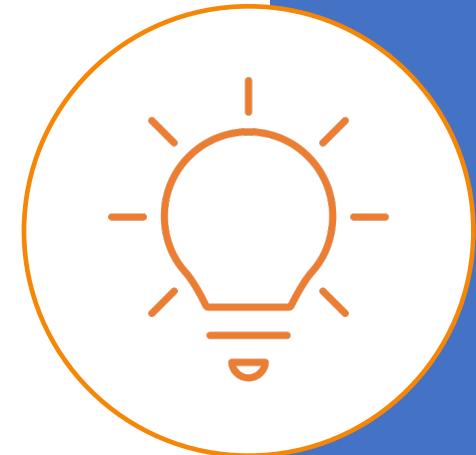


Chart Credit: Pitchbook 2021, Q3, Emerging Tech Report

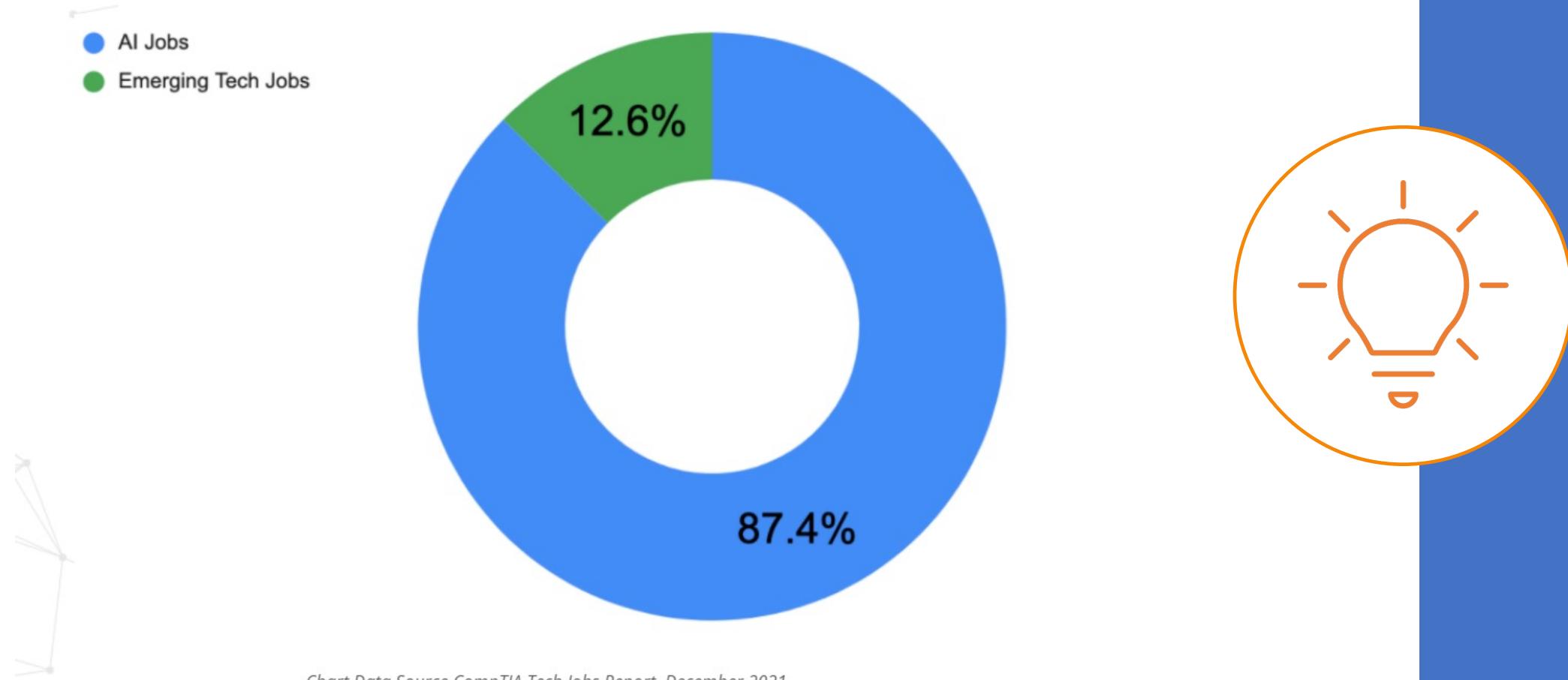
# Why should you take NLP Course?



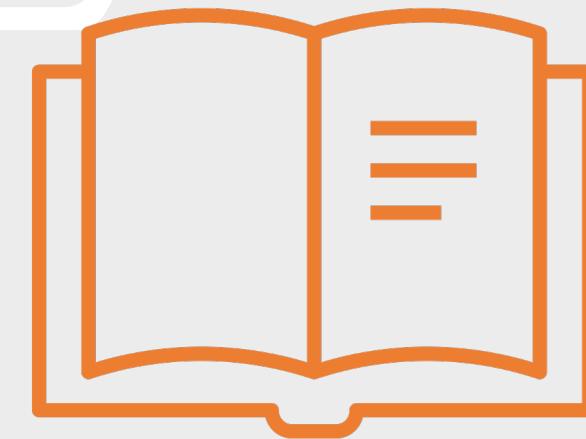
- 7% We don't use AI currently but are looking into it
- 14% We have tested a few proofs of concepts with limited success
- 21% We have a few promising proofs of concepts and are looking to scale
- 25% We have processes fully enabled by AI with widespread adoption
- 33% We have started implementing limited AI use cases



# Why should you take NLP Course?



# Course Scope



# Topics Covered

---

## Models/Methods

- Text Classification with ML
- Feed-forward networks
- Word Embedding Models
- Transformer Models
- Transfer Learning/Fine-tuning
- Handling few/No label Cases
- Parameter efficient fine-tuning
- Semantic Search
- RAG

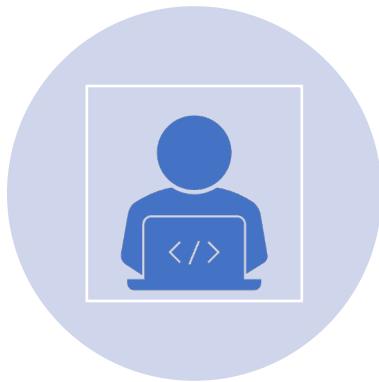
## Applications

- Sequence Classification
  - Sentiment Analysis
  - Stack Exchange Question Tagging
- Token Classification
  - Name Entity Recognition
- Sequence-to-Sequence
  - Question/Answering
  - Summarization
  - Machine Translation



# Prerequisites

---



PROGRAMMING,  
ESPECIALLY PYTHON



BUAN 6341: MACHINE  
LEARNING



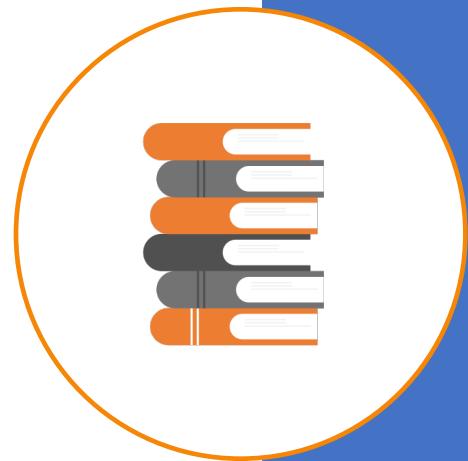
# Syllabus



# Syllabus - NLP

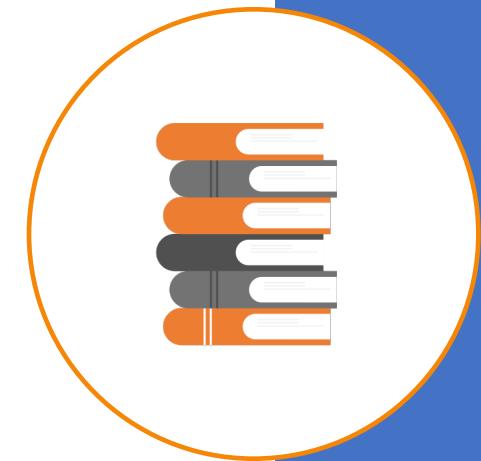
Date	Content
15 <sup>th</sup> Jan	- Martin Luther King Holiday
22 <sup>nd</sup> Jan	- Intro Course - Spacy and Pre-Processing - HW1: ( <del>PyTorch</del> Tensors- Watch video and complete HW)
29 <sup>th</sup> Jan	- Sparse Embeddings ( <del>tfidf</del> ) - Sentiment Analysis with Spacy and <del>sklearn</del> - Into Neural Networks - I (Theory) - HW2: Classification using Spacy and <del>sklearn</del>
5 <sup>th</sup> Feb	- Into Neural Networks - II (Theory) - <del>PyTorch</del> Layers - Numeric Prediction with <del>PyTorch</del> - Classification with <del>PyTorch</del> - Introduction to Embeddings - IMDB Sentiment Analysis with simple DL ( <del>PyTorch</del> ) - HW3 (Stack Exchange Multi-Class (Imbalanced dataset) with <del>PyTorch</del> (skip connection model) - Two experiments.
12 <sup>th</sup> Feb	- Intro to Hugging Face - IMDB Sentiment Analysis with simple DL ( <del>PyTorch</del> + Hugging Face) - Word2Vec/Language Models/RNN/LSTM (Theory) - HW 4 (Stack Exchange Multi-Label with Hugging Face Trainer)
19 <sup>th</sup> Feb	- Encoder-Decoder/Attention/Transformer (Theory) - Understanding BERT/GPT/T5 (Theory) - IMDB Sentiment Analysis using <u>BERT</u> - Revisit Sentiment analysis (Domain Adaptation – BERT) - HW5 (Emotion Detection with Flan T5)
26 <sup>th</sup> Feb	- Finding duplicates in a given pair of Quora questions (Sentence pair classification using BERT) - Stack exchange multi-label classification using <u>BERT</u> - Revisit IMDB sentiment Analysis (Handling Longer Sequences) - Named Entity Recognition (Token Classification) - HW6 (NER – Extracting Brand Names)
26 <sup>th</sup> Feb	- <u>Kaggle Competition</u>
4 <sup>th</sup> march	- Current Models (GPT4, LLAMA, Mistral) - Parameter Efficient fine tuning (LLAMA) - OpenAI Finetuning - HW7 – Finetuning with LORA
11 <sup>th</sup> march	- Break
18 <sup>th</sup> – 22 <sup>nd</sup> March	- Exam1

**HWs are Tentative and will most likely change. We will probably have less than the schedule here**



# Syllabus - NLP

25 <sup>th</sup> March	<ul style="list-style-type: none"><li>- Intro to Sentence-BERT (Sentence Embeddings)</li><li>- Finding top-k similar questions</li><li>- <u>BerTopic</u> (Topic Modelling using Bert)</li><li>- HW 8 Topics Modeling with <u>BERTopic</u></li></ul>
1 <sup>st</sup> April	<ul style="list-style-type: none"><li>- Few to No Label (Zero Shot classification, data augmentation)</li><li>- HW9 (Zero-Shot Classification)</li></ul>
8 <sup>th</sup> April	<ul style="list-style-type: none"><li>- Revisit the Language model (focus on inference)</li><li>- Beam Search/Greedy Decoding</li><li>- Sampling in Text Generation (Top-k, Top-p, Temperature, Greedy, Random)</li><li>- Sequence to Sequence Tasks</li><li>- Translation/Summarization/Simple Question-Answer</li><li>- HW10 (Dialogue Summarization)</li></ul>
15 <sup>th</sup> April	<ul style="list-style-type: none"><li>- Intro to Prompt Engg (PE), Semantic Search (SS), RAG, <u>LangChain</u> (LC), Vector Database (VD) -I</li><li>- <b>Kaggle Competition Report – Mid report/Mandatory First Submission</b></li></ul>
22 <sup>nd</sup> April	<ul style="list-style-type: none"><li>- PE, SS, RAG, LG, VD -II</li><li>- HW11</li></ul>
29 <sup>th</sup> April	<ul style="list-style-type: none"><li>- PE, SS, RAG, LG, VD -III</li></ul>
6 <sup>th</sup> May – 10 <sup>th</sup> May	<b>Exam2</b>
10 <sup>th</sup> May	<b>Final Kaggle Competition Report/ No more submissions allowed</b>



# Grading

Class Participation	10
Homework Assignments	50
Exam 1	20
Exam 2	20
Total	100

Your course grade will depend on your overall score relative to your peers.

- 1) The students with scores in the 80th percentile and above will get an A grade.
- 2) The students with scores between the 80<sup>th</sup> and the 60<sup>th</sup> percentile will get an A- grade.
- 3) The students with scores between the 60<sup>th</sup> and the 40<sup>th</sup> percentile will get a B+ grade.
- 4) The students with scores between the 40<sup>th</sup> and the 20<sup>th</sup> percentile will get a B grade.
- 5) The students with scores between the 20<sup>th</sup> and the 10<sup>th</sup> percentile will get a B- grade.
- 6) The instructor will decide the students with scores below the 10th percentile.



# Exams

---

Exams will be proctored at the UTD Testing Center.

**Students will need to make reservation using “Reserve Your Seat” tool at least 72 hours prior to the exam time.**



# Late work Policy for HWs

---

Homework assignments will be turned in on eLearning and are due at 11:59pm CT on their assigned due date.

---

**Each student has a total of three free late days to use on assignments throughout the semester.**

---

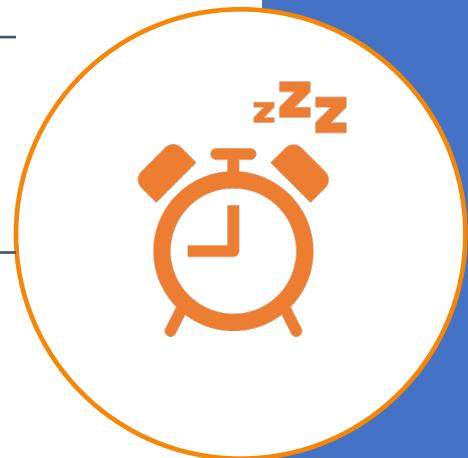
There is no restriction on how you can use the late days. You can use the late days on one assignment or across multiple assignments.

---

You do not need permission to use free late days. We will keep track of late days that you have used.

---

After you have exhausted free late days, you will receive a penalty of 25% per late day. You do not need to take any permissions. The minimum score in the assignment will be 0.



# Expectations

---

Individual Assignments

Impromptu class quizzes – No prior announcement- part of HW grade

5 – 10 hours/week – HWs/ Readings

The lectures will also be streamed on Teams

- Please use this only if you cannot attend in person.
- I am not responsible for the quality of streaming/videos or recordings

No Plagiarism – First and last warning



# Coding Environment: Google Colab

---

Programming in the browser

---

Codes, instructions, and outputs are displayed “in-line”

---

Free Access to GPUs (required hardware for Deep Learning)

---

Useful for writing codes that tells a story

---

Used by scientists and researchers

---

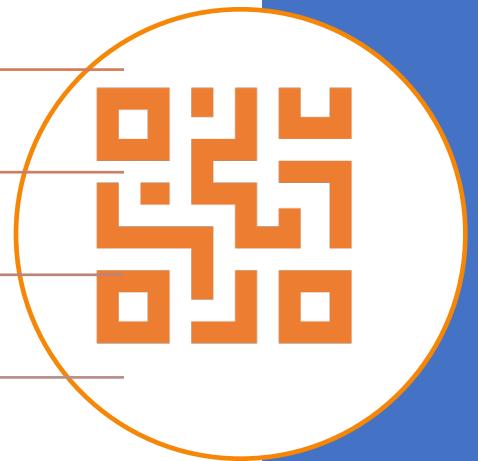
Integration with Google Drive

---

No installation

---

**Recommendation: Get a pro version (\$10/month)**



# Framework

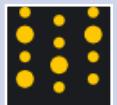
---



PyTorch (Deep Learning)



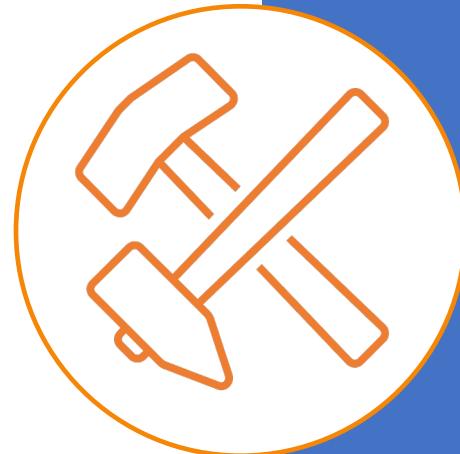
Huggingface Trainer (High Level API)



wandb (Logging and saving model artifacts)

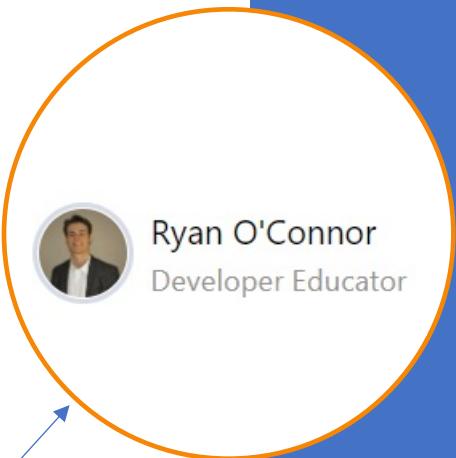
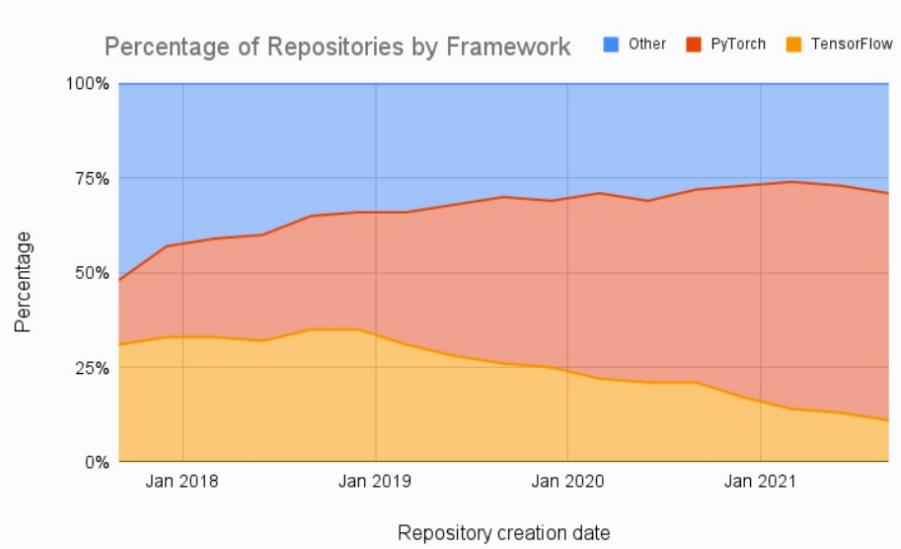


Spacy/NLTK/Hugging face (Pre-processing)



# Why PyTorch?

PyTorch	TensorFlow 2.0
<ul style="list-style-type: none"><li>• Clean API</li><li>• Easy dynamic graphs for debugging</li><li>• Hard / inefficient to use on TPUs</li><li>• Not easy to deploy on mobile</li></ul>	<ul style="list-style-type: none"><li>• API is little confusing</li><li>• Dynamic graphs by default</li><li>• Easier to use on TPUs</li><li>• Standardized High level API – Keras</li></ul>

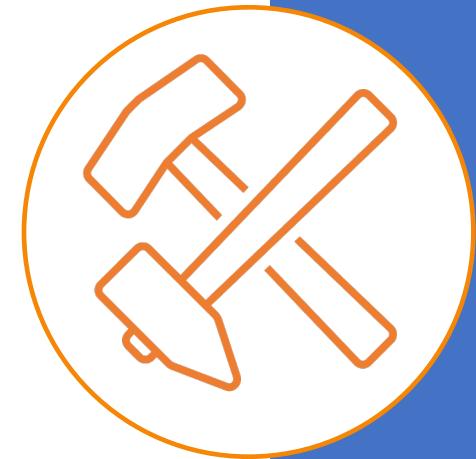


# Why PyTorch?

---

PyTorch strengthens its governance by joining the Linux Foundation (Sep 2022)

Keras Core: Keras for TensorFlow, JAX, and PyTorch (stable version coming in next few months. (Aug 2023)



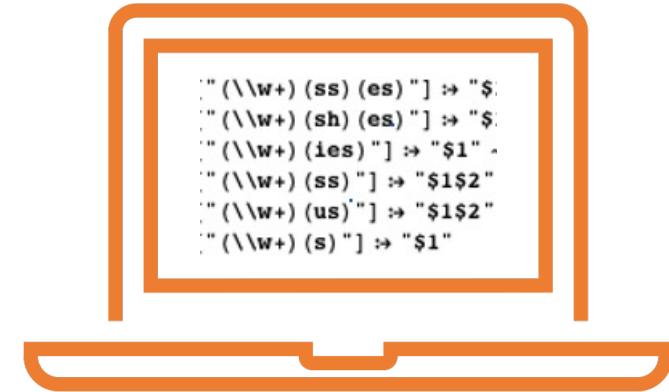
# How to get maximum out of this course?

---

- Write as much code as you can. Your understanding will improve immensely. The rule is “If you can't implement it, then you don't understand it. “
- Even if you find a solution online - do not use copy and paste- always type your code.
- Use ChatGPT to get a better understanding but try to code yourself



# Preprocessing



```
:"(\w+) (ss) (es)"] => "$1  
:"(\w+) (sh) (es)"] => "$1  
:"(\w+) (ies)"] => "$1$2  
:"(\w+) (ss)"] => "$1$2"  
:"(\w+) (us)"] => "$1$2"  
:"(\w+) (s)"] => "$1"
```

# Unstructured Data

---

ML

- Fixed number of features
- Contiguous
- Categorical

NLP

- No pre-defined features



# Unstructured Data

Free string – but not “words”



	country	fullName	Id	nationalPoliticalGroup	politicalGroup
0	Sweden	Lars ADAKTUSSON	124990	Kristdemokraterna	Group of the European People's Party (Christia...
1	Italy	Isabella ADINOLFI	124831	Movimento 5 Stelle	Europe of Freedom and Direct Democracy Group
2	Italy	Marco AFFRONTI	124797	Movimento 5 Stelle	Group of the Greens/European Free Alliance
3	Italy	Laura AGEA	124811	Movimento 5 Stelle	Europe of Freedom and Direct Democracy Group
4	United Kingdom	John Stuart AGNEW	96897	United Kingdom Independence Party	Europe of Freedom and Direct Democracy Group



Categorical (If you're lucky)



Named Entities



# Text Data is unstructured

---

@amankedia I'm going to buy a Rolexxxxxxxx watch!!!  
:-D #happiness #rolex <3



\*\*May Contain Spoilers\*\*<br /><br />A dude in a dopey-looking Kong suit (the same one used in KING KONG VS. GODZILLA in 1962) provides much of the laughs in this much-mocked monster flick. Kong is resurrected on Mondo Island and helps out the lunkhead hero and other good guys this time around. The vampire-like villain is named Dr. Who-funny, he doesn't look like Peter Cushing! Kong finally duked it out with Who's pride and joy, a giant robot ape that looks like a bad metal sculpture of Magilla Gorilla. Like many of Honda's flicks this may have had some merit before American audiences diddled around with it and added new footage. The Rankin/Bass animation company had a hand in this mess. They should have stuck to superior children's programs like The Little Drummer Boy.

# Preparing data for model

---

## Preprocessing

- Combining/merging Data
- Removing unnecessary text (e.g. html tags)
- Tokenization
- Stemming
- Lemmatization
- Stop Words

## Vectorization

- Sparse Embeddings
- Dense Embeddings



# Definitions

Words

- Sequence of characters with a meaning and/or function.

Sentence

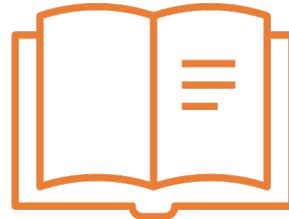
- Sequence of words.

Document

- One or more sentences e.g. Tweet, Review.

Corpus:

- A collection of documents.



# Text Normalization

Remove unwanted formatting (e.g. HTML)

Segment structure (e.g., sentences)

Tokenise words

Normalise words

Remove unwanted words

Stop Words

“<p> Hi All. I’m your  
instructor. </p>”

["Hi All.",  
 "I'm your  
 instructor."]

[["hi", "all", "."],  
 ["i", "am", "your", "instructor", "."]]

“Hi All. I’m your  
instructor.”

[["Hi", "All", "."],  
 ["I", "m", "your", "instructor", "."]]

[[],["instructor"]]



# Tokenization-Whitespace

---

```
text.split(" ")
```

Regular expressions are extremely useful in extracting information from text such as code, log files, spreadsheets, or even documents. And while there is a lot of theory behind formal languages, the following lessons and examples will explore the more practical uses of regular expressions so that you can use them as quickly as possible



# Tokenization-Whitespace

---

Regular expressions are extremely useful in extracting information from text such as **code**, **log files**, **spreadsheets**, or even documents. And while there is a lot of theory behind formal **languages**, the following lessons and examples will explore the more practical uses of regular expressions so that you can use them as quickly as possible

languages,
languages?
languages.
languages,"
languages!
language-of
Language--is



# Punctuation

Punctuations can be useful

- Punctuation signals boundaries e.g., end of sentence.
- Can convey speaker's intention - exclamation points (!) and question marks (?)
- Emoticons are strong signals of e.g. sentiment.



# Tokenization: English

Another Naïve approach: separate out alphanumeric strings (`\w+`)

- Abbreviations (*U.S.A.*)
- Hyphens (*merry-go-round* vs. *well-respected* vs. *yes-but*)
- Numbers (*1,000,00.01*)
- Dates (*3/1/2016*)
- Clitics (*n't* in *can't*)
- Internet language (<http://www.google.com>, #metoo, :-))
- Multiword units (New Zealand)



# spaCy Tokenizer-Rule based

Iterate over space-separated substrings.

Check whether we have an explicitly defined special case for this substring. If we do, use it.

Look for a token match. If there is a match, stop processing and keep this token.

Check whether we have an explicitly defined special case for this substring. If we do, use it.

Otherwise, try to consume one prefix. If we consumed a prefix, go back to #3, so that the token match and special cases always get priority.

If we didn't consume a prefix, try to consume a suffix and then go back to #3.

If we can't consume a prefix or a suffix, look for a URL match.

If there's no URL match, then look for a special case.

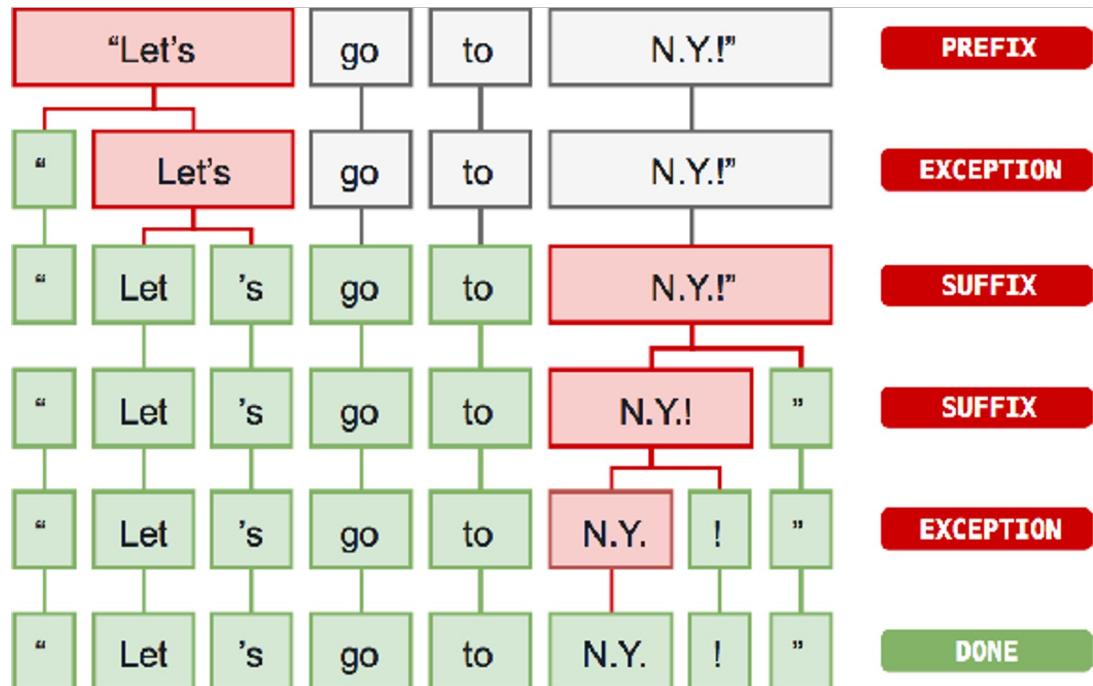
Look for “infixes” – stuff like hyphens etc. and split the substring into tokens on all infixes.

Once we can't consume any more of the string, handle it as a single token.

Make a final pass over the text to check for special cases that include spaces or that were missed due to the incremental processing of affixes.



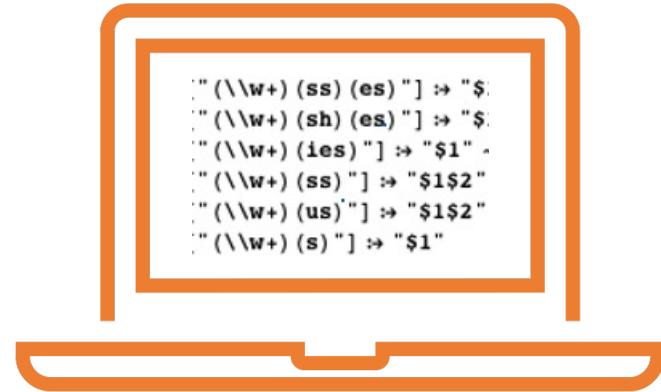
# spaCy Tokenizer



- Split text on white space
- Look for a token match
- Try to consume prefix,
- Try to consume suffix
- Check for exceptions like N.Y. , Let's
- Check if it is an URL
- Check for infixes like hyphen , slash

<https://spacy.io/usage/spacy-101#annotations-token>

# Feature Engineering



# Feature Engineering

- Basic Text Features
  - Document Length
  - Word Length
  - Character Count
  - Sentence Count
  - Punctuation Count
- Lexical Features
  - Part-of-speech (POS) tags: Count of nouns, verbs, adjectives, etc.
  - Named Entity Recognition (NER): Count of named entities, and possibly categorizing them into person, organization, location, etc.
- Sparse Embeddings: Frequency of specific words or phrases



# Parts of Speech Tags

## Universal POS Tags

- These tags mark the core part-of-speech categories

	<b>Tag</b>	<b>Description</b>	<b>Example</b>
Open Class	<b>ADJ</b>	Adjective: noun modifiers describing properties	<i>red, young, awesome</i>
	<b>ADV</b>	Adverb: verb modifiers of time, place, manner	<i>very, slowly, home, yesterday</i>
	<b>NOUN</b>	words for persons, places, things, etc.	<i>algorithm, cat, mango, beauty</i>
	<b>VERB</b>	words for actions and processes	<i>draw, provide, go</i>
	<b>PROPN</b>	Proper noun: name of a person, organization, place, etc..	<i>Regina, IBM, Colorado</i>
	<b>INTJ</b>	Interjection: exclamation, greeting, yes/no response, etc.	<i>oh, um, yes, hello</i>
Closed Class Words	<b>ADP</b>	Adposition (Preposition/Postposition): marks a noun's spacial, temporal, or other relation	<i>in, on, by under</i>
	<b>AUX</b>	Auxiliary: helping verb marking tense, aspect, mood, etc.,	<i>can, may, should, are</i>
	<b>CCONJ</b>	Coordinating Conjunction: joins two phrases/clauses	<i>and, or, but</i>
	<b>DET</b>	Determiner: marks noun phrase properties	<i>a, an, the, this</i>
	<b>NUM</b>	Numerical	<i>one, two, first, second</i>
	<b>PART</b>	Particle: a preposition-like form used together with a verb	<i>up, down, on, off, in, out, at, by</i>
	<b>PRON</b>	Pronoun: a shorthand for referring to an entity or event	<i>she, who, I, others</i>
	<b>SCONJ</b>	Subordinating Conjunction: joins a main clause with a subordinate clause such as a sentential complement	<i>that, which</i>
Other	<b>PUNCT</b>	Punctuation	<i>;, ,0</i>
	<b>SYM</b>	Symbols like \$ or emoji	<i>\$, %</i>
	<b>X</b>	Other	<i>asdf, qwfg</i>

**Figure 8.1** The 17 parts of speech in the Universal Dependencies tagset (Nivre et al., 2016a). Features can be added to make finer-grained distinctions (with properties like number, case, definiteness, and so on).



# Parts of Speech Tags

## Detailed POS tags

- Division of Universal Tags into multiple tags

Tag	Description	Example	Tag	Description	Example	Tag	Description	Example
CC	coord. conj.	<i>and, but, or</i>	NNP	proper noun, sing.	<i>IBM</i>	TO	"to"	<i>to</i>
CD	cardinal number	<i>one, two</i>	NNPS	proper noun, plu.	<i>Carolinas</i>	UH	interjection	<i>ah, oops</i>
DT	determiner	<i>a, the</i>	NNS	noun, plural	<i>llamas</i>	VB	verb base	<i>eat</i>
EX	existential 'there'	<i>there</i>	PDT	predeterminer	<i>all, both</i>	VBD	verb past tense	<i>ate</i>
FW	foreign word	<i>mea culpa</i>	POS	possessive ending	<i>'s</i>	VBG	verb gerund	<i>eating</i>
IN	preposition/ subordin-conj	<i>of, in, by</i>	PRP	personal pronoun	<i>I, you, he</i>	VBN	verb past partici- ple	<i>eaten</i>
JJ	adjective	<i>yellow</i>	PRP\$	possess. pronoun	<i>your, one's</i>	VBP	verb non-3sg-pr	<i>eat</i>
JJR	comparative adj	<i>bigger</i>	RB	adverb	<i>quickly</i>	VBZ	verb 3sg pres	<i>eats</i>
JJS	superlative adj	<i>wildest</i>	RBR	comparative adv	<i>faster</i>	WDT	wh-determ.	<i>which, that</i>
LS	list item marker	<i>1, 2, One</i>	RBS	superlatv. adv	<i>fastest</i>	WP	wh-pronoun	<i>what, who</i>
MD	modal	<i>can, should</i>	RP	particle	<i>up, off</i>	WP\$	wh-possess.	<i>whose</i>
NN	sing or mass noun	<i>llama</i>	SYM	symbol	<i>+, %, &amp;</i>	WRB	wh-adverb	<i>how, where</i>

Figure 8.2 Penn Treebank part-of-speech tags.



# Parts-of-Speech (POS) Tagging

Process of assigning different labels known as POS tags

TEXT	LEMMA	POS	TAG
Apple	apple	PROPN	NNP
is	be	AUX	VBZ
looking	look	VERB	VBG
at	at	ADP	IN
buying	buy	VERB	VBG
U.K.	u.k.	PROPN	NNP
startup	startup	NOUN	NN



# Why do POS Tagging?

---



**Named Entity Recognition (NER):** POS tags as a back-off method for NER. Recognizing proper nouns aids in finding named entities like people, places, or organizations.



**Machine Translation:** POS tags preserve grammatical structure during language translation.



**Text-to-Speech (TTS):** POS tags help generate natural-sounding speech, e.g., "How do you pronounce 'lead'?"



**Regular Expressions and Phrase Identification:** POS tags simplify extracting structured information by using patterns to identify syntactic structures like noun phrases, verb phrases, or adjective phrases



# Named Entities

A named entity is, roughly speaking, anything that can be referred to with a proper name: a person, a location, an organization

Citing high fuel prices, [ORG United Airlines] said [TIME Friday] it has increased fares by [MONEY \$6] per round trip on flights to some cities also served by lower-cost carriers. [ORG American Airlines], a unit of [ORG AMR Corp.], immediately matched the move, spokesman [PER Tim Wagner] said. [ORG United], a unit of [ORG UAL Corp.], said the increase took effect [TIME Thursday] and applies to most routes where it competes against discount carriers, such as [LOC Chicago] to [LOC Dallas] and [LOC Denver] to [LOC San Francisco].



Type	Tag	Sample Categories	Example sentences
People	PER	people, characters	Turing is a giant of computer science.
Organization	ORG	companies, sports teams	The IPCC warned about the cyclone.
Location	LOC	regions, mountains, seas	Mt. Sanitas is in Sunshine Canyon.
Geo-Political Entity	GPE	countries, states	Palo Alto is raising the fees for parking.

Figure 8.5 A list of generic named entity types with the kinds of entities they refer to.

# Named Entity Recognition

Identifying spans of text that correspond to typed entities that are proper names

Two Steps

- Named Entity Extraction
- Named Entity Classification

[tim cook]<sub>PER</sub> is the ceo of [apple]<sub>ORG</sub>



# Domain Specific : Bio Entity Extraction

GENIA corpus of MEDLINE abstracts (biomedical)

We have shown that <cons sem="G#protein">interleukin-1</cons> (<cons sem="G#protein">IL-1</cons>) and <cons sem="G#protein">IL-2</cons> control <cons sem="G#DNA">IL-2 receptor alpha (IL-2R alpha) gene</cons> transcription in <cons sem="G#cell\_line">CD4-CD8-murine T lymphocyte precursors</cons>.

protein  
DNA  
Cell\_line



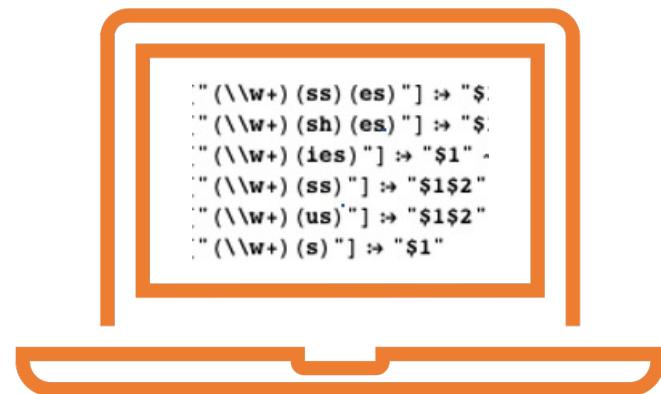
# IO, BIO and BIOES Tagging

Words	IO Label	BIO Label	BIOES Label
Jane	I-PER	B-PER	B-PER
Villanueva	I-PER	I-PER	E-PER
of	O	O	O
United	I-ORG	B-ORG	B-ORG
Airlines	I-ORG	I-ORG	I-ORG
Holding	I-ORG	I-ORG	E-ORG
discussed	O	O	O
the	O	O	O
Chicago	I-LOC	B-LOC	S-LOC
route	O	O	O
.	O	O	O



**Figure 8.7** NER as a sequence model, showing IO, BIO, and BIOES taggings.

# Vectorization



# Corpus

Restaurant Reviews	
R1	Great restaurant and great service !
R2	They can do better to provide better service
R3	Only two thumbs up, worst service ever

Entire Corpus



# Corpus

Set of all the words in the corpus
great
restaurant
and
service
they
can
do
better
to
provide
only
Two
thumbs
up
worst
ever



# Binary Vectorizer

<b>Set of all the words in the corpus</b>	<b>R1: Great Restaurant and great service !</b>	<b>R2: They can do better to provide better service</b>	<b>R3: Only two thumbs up, worst service ever</b>
great	1	0	0
restaurant	1	0	0
and	1	0	0
service	1	1	0
they	0	1	0
can	0	1	0
do	0	1	0
better	0	1	0
to	0	1	0
provide	0	1	0
only	0	0	1
Two	0	0	1
thumbs	0	0	1
up	0	0	1
worst	0	0	1
ever	0	0	1

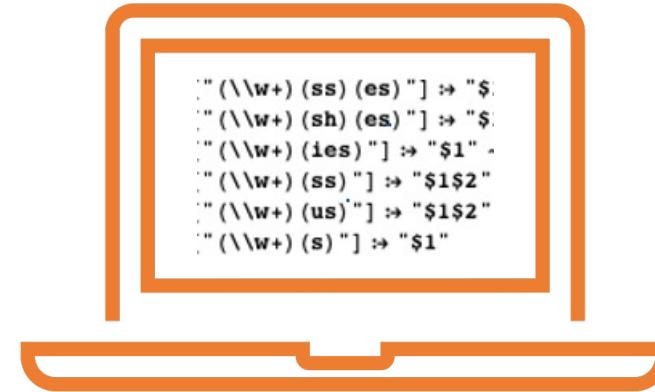


# Binary Vectorizer

Set of all the words in the corpus	R1: Great Restaurant and great service !	R2: They can do better to provide better service	R3: Only two thumbs up, worst service ever
great	2	0	0
restaurant	1	0	0
and	1	0	0
service	1	1	0
they	0	1	0
can	0	1	0
do	0	1	0
better	0	2	0
to	0	1	0
provide	0	1	0
only	0	0	1
Two	0	0	1
thumbs	0	0	1
up	0	0	1
worst	0	0	1
ever	0	0	1



# Adjusting Vocab Size



# Lemmatization

Systematic process for reducing token to its lemma

- *Running, ran, run, runs → run*
- *am, is, ate → be*

In English, there are irregularities that prevent a trivial solution:

- *stopping → stop (not stopp)*

A lexicon of lemmas needed for accurate lemmatization

Running, ‘verb’ → run

Running, ‘noun’ → running

Slower than Stemming



# Stemming

Elementary rule-based process of removing suffixes, leaving a *stem*

Often not an actual lexical item

- *automate, automatic, automation* → *automat*
- *Compressed, compression* → *compress*

Even less lexical sparsity than lemmatization

Popular in information retrieval



# Porter's algorithm

---



One of the most common - and effective - stemming tools is Porter's Algorithm developed by Martin Porter in 1980.



The algorithm employs five phases of word reduction, each with its own set of mapping rules.



# Porter's algorithm

## Step 1a

sses → ss	caresses → caress
ies → i	ponies → poni
ss → ss	caress → caress
s → ø	cats → cat

## Step 1b

(*v*)ing → ø	walking → walk
	sing → sing

In the first phase, simple suffix mapping rules are defined.



The \*v\* symbol is a placeholder for a vowel. This means the rule applies to any word that has a vowel before the -ing suffix.

# Porter's algorithm

## Step 2 (for long stems)

ational → ate

relational → relate

Izer → ize

digitizer → digitize

Ator → ate

operator → operate

...

## Step 3 (for longer stems)

al → ø

revival → reviv

able → ø

adjustable → adjust

ate → ø

activate → activ

...

More sophisticated phases consider the length/complexity of the word before applying a rule



# Snowball stemmer

Snowball is the name of a stemming language also developed by Martin Porter.

The algorithm used here is more accurately called the "English Stemmer" or "Porter2 Stemmer".

It offers a slight improvement over the original Porter stemmer, both in logic and speed.

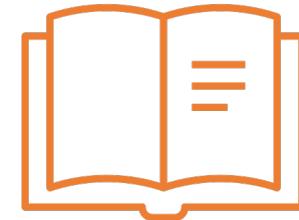


# Fixing Spelling errors

---

Why fix them?

- Spelling errors create new, rare types
- Very common in internet corpora
- In web search, particularly important in queries



How to fix them?

- String distance (Levenshtein, etc.)

# Other word Normalizations

---

## Expanding

- Expanding abbreviations
  - US, U.S. → United States
  - imho → in my humble opinion

## Spelling

- Spelling variations
  - Normalize → Normalise (or vice versa)
  - U r so coool! → you are so cool



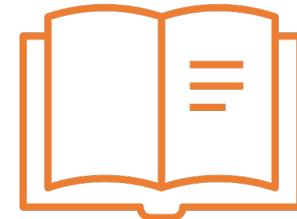
# Stop Words

Definition: a list of words to be removed from the document

- Typical in bag-of-word (BOW) representations
- Not appropriate when sequence is important

How to choose them?

- All *closed-class* or *function* words
  - E.g. *the, a, of, for, he, ...*
- Any high frequency words



# Stop Words

## Remove Stop Words

- Text Classification
- Caption Generation
- Auto-Tag Generation

## Do Not Remove Stop Words

- Machine Translation
- Language Modeling
- Text Summarization
- Question-Answering Problems



# A final word

---



Preprocessing  
unavoidable in text  
analysis



Can have a major  
effect on downstream  
applications



Exact steps may vary  
depending on corpus,  
task



Simple rule-based  
systems work well,  
but rarely perfectly



# Things To Do Before Next Class

---

- HW1: Pytorch Tensors (Similar to Numpy, Video will be posted tonight)
  - Auto Grad
  - GPUs
- Basic Machine Learning
  - Overfitting/Underfitting
  - Need of Data splits
  - Data leakage (notebook will be posted)
  - Pipelines in sklearn (notebook will be posted)
  - Gradient Descent (We will revise in class\_



