

Generated by DALL-E



VOYAGE, VOYAGE

Luísa Coheur

OVERVIEW

- Learning objectives
- Topics
 - Concepts
 - Main challenges
 - Historical overview
- Key takeaways

LEARNING OBJECTIVES

LEARNING OBJECTIVES

- After this class, students should be able to:
 - Define concepts such as Natural Language, paraphrases, ellipsis, co-reference, etc.
 - Explain some of the main challenges in NLP
 - Explain the concepts related with the linguistic approach
 - Explain how NLP move forwards over the last 80 years
 - Explain some NLP tasks

TOPICS

OVERVIEW

- Learning objectives
- Topics
 - Concepts
 - Main challenges
 - Historical overview
- Key takeaways

NATURAL LANGUAGE

- Language
 - Grammatical system
 - Own rules
 - Used to communicate
- Natural
 - Natural evolution due to people communication
 - New words everyday!
 - Yet, some control in adding them to dictionaries...

ACTIVE LEARNING MOMENT: THINK-PAIR-SHARE



EXERCISE

Can you identify some new words in your native language (or in English)?

Can you identify words that were removed from our vocabulary?

EXAMPLES OF NATURAL LANGUAGES

- Portuguese, English

- Portuguese Sign Language

- ...

- What about Esperanto?

NATURAL LANGUAGES

- Esperanto is a language designed Dr Ludwig Lazarus Zamenhof (1887) to be easy to learn and promote communication
 - Example (morphology):
 - All nouns end in “o”
 - noun -> corresponding adjective: replace the “o” with an “a”
 - To make a plural, add a “j” to the end

EXERCISE



The Esperanto alphabet

a b c ç d e f g ã h ĥ i j ã k l m n o p r s ã t u ù v z

ESPERANTO
Saluton
Ĝis
Dankon
Nomo
Amiko
Hundo
Kato

ESPERANTO
Jaro
Patro
Patrino
Viro
Virino
Doktoro
Tago

ESPERANTO
Tempo
Trinku
Mangi
Dormu
Hejmo
Banejo
Restoracio

Vi estas welcome You're welcome De Nada

NATURAL LANGUAGE PROCESSING

- Natural Language Processing (NLP)
 - Field of Artificial Intelligence and Linguistics concerned with the interactions between computers and human (natural) languages

OVERVIEW

- Learning objectives
- Topics
 - Concepts
 - Main challenges
 - Historical overview
- Key takeaways

MAIN CHALLENGES



LANGUAGE
VARIABILITY

**TO DEVELOP
DIALOGUE
SYSTEMS is a
main
NLP TASK!**

MAIN CHALLENGES

It is so dark...



“Open” the light!

Lumos!

LANGUAGE
VARIABILITY

BY THE WAY

- Concept: two sentences with the same meaning are called **paraphrases**

MAIN CHALLENGES

Can you
turn on the
light,
please?



Yes, I can!

AMBIGUITY

MAIN CHALLENGES

Can you
turn on the
light,
please?



AMBIGUITY/VAGUENESS

BY THE WAY

- As we have seen, a single sentence can have different meanings
- The source of those different meanings can vary
 - There are many different types of ambiguity

BY THE WAY

- Some words have multiple meanings (concept: lexical ambiguity)



And if your dog's name is Step?

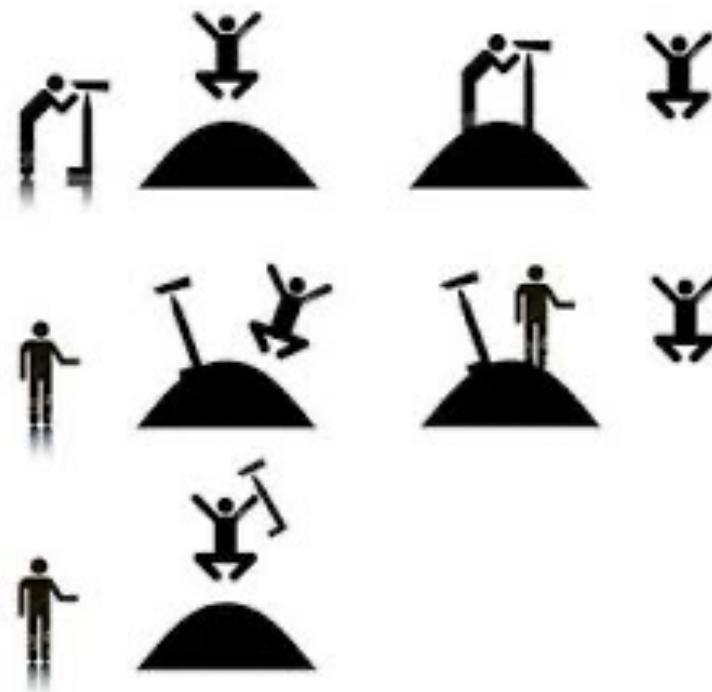
BY THE WAY

- My favorite ambiguity moment



BY THE WAY

- A sentence has multiple parse trees (concept: syntactic ambiguity)
 - I saw the man on the hill with a telescope



BY THE WAY

- All ambiguities “are” semantic, but some are not related with lexical and syntactic ambiguity
 - John and Mary are married.
 - One with the other? With other people?
 - John kissed his wife and Peter too
 - Well...
- And there are many other types...

**Word sense
disambiguation
is an
NLP TASK!**

MAIN CHALLENGES

Can you
turn on the
light,
please?



Now the radio!

ELLIPSIS

Now turn
it off!

CO-REFERENCE

Co-reference
resolution is an
NLP TASK!

MAIN CHALLENGES

Can you
turn on the
light,
please?



Now the radio!

ELLIPSIS

Agora
desliga-a!

CO-REFERENCE

MAIN CHALLENGES

Can you
turn on the
light,
please?



Now the radio!

ELLIPSIS

Agora
desliga-o!

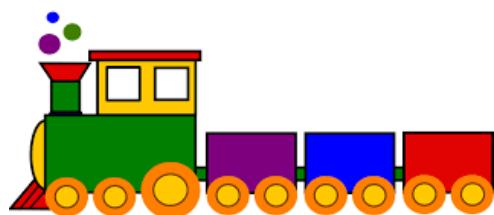
CO-REFERENCE

BY THE WAY

- Concepts:
 - Ellipsis: the omission of a word or words that are able to be understood from contextual clues
 - Co-reference: elements of a text that have the same reference

MAIN CHALLENGES

Can you
turn on the
light,
please?



**There is a course
focused on
Speech in P4**



NOISE

MAIN CHALLENGES

Can you
turn on the
light,
please?



DIFFERENT
ACCENTS

ENGLISH

From <https://www.worldaccent.com/>

MAIN CHALLENGES

DIFFERENT AGES

Can you turn on
the light, please?



Nha nha nhã?



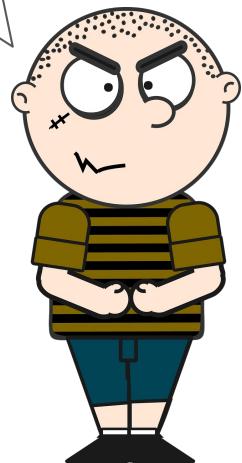
MAIN CHALLENGES

At 4pm I will be waiting
for you outside school!



MAIN CHALLENGES

At 4pm I will be waiting
for you outside school!



CONTEXT

We had a
project about
Cyberbullying
detection!

MAIN CHALLENGES

“Those who believe in telekinetics, raise my hand.” – Kurt Vonnegut

SARCASM AND IRONY

Generated by DALL-E



OVERVIEW

- Learning objectives
- Topics
 - Concepts
 - Main challenges
 - [Historical overview](#)
- Key takeaways

40s

- Machine Translation
 - Warren Weaver (1949) suggested using cryptographic techniques and statistical methods to translate languages



We will have a
class dedicated
to Machine
Translation, a
main NLP task!

RULE-BASED APPLICATIONS

60s



- Daniel G. Bobrow developed Student (PhD. Thesis, 1964)
 - Student solves algebra problems (implemented in LISP)
 - Example:
 - User: “The price of a radio is 69.7 dollars. If the price is 15 percent less than the marked price, find the marked price.”
 - STUDENT: The marked price is 82 dollars

RULE-BASED APPLICATION

60s

- Joseph Weizenbaum developed Eliza (1966)

```
Welcome to
      EEEEEE  LL      IIII  ZZZZZZ  AAAAAA
      EE      LL      II      ZZ  AA  AA
      EEEEEE  LL      II      ZZZ  AAAAAAAA
      EE      LL      II      ZZ  AA  AA
      EEEEEE  LLLLLL  IIII  ZZZZZZ  AA  AA

Eliza is a mock Rogerian psychotherapist.
The original program was described by Joseph Weizenbaum in 1966.
This implementation by Norbert Landsteiner 2005.

ELIZA: Is something troubling you ?
YOU: Men are all alike.
ELIZA: What is the connection, do you suppose ?
YOU: They're always bugging us about something or other.
ELIZA: Can you think of a specific example ?
YOU: Well, my boyfriend made me come here.
ELIZA: Is it important to you that your boyfriend made you come here ?
YOU: He says I'm depressed much of the time.
ELIZA: I am sorry to hear that you are depressed.
YOU: It's true. I am unhappy.
ELIZA: Can you explain what made you unhappy ?
YOU: ■
```



ALSO: RULE-BASED APPLICATION

60s

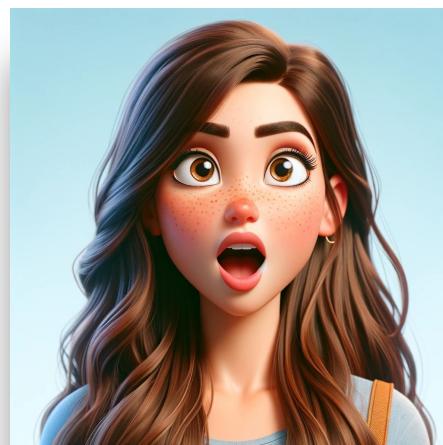
- How did ELIZA work?

USER: I think my cat has powers.

USER: I think my <anything>

ELIZA: Why do you think your <anything>

ELIZA: Why do you think your cat has powers.



80's

RULE-BASED

CHIT-CHAT AND “PERSONA” BOTS

Talk with Captain Kirk



QUESTION/ANSWERING AND
TASK ORIENTED DIALOGUE
SYSTEMS



Who was Fernando
Pessoa?

- I need a ticket to Braga
- ...

**Question/Answer and the development of
Dialogue System are NLP TASKs!**

90's

- Release of linguistic and computational resources



**We will have a
class dedicated
to datasets
(corpora) and
we will use
several
computational
NLP resources
along this course**

90's

- International evaluation fora/campaigns/shared tasks



Image from <https://www.publicdomainpictures.net/>

These campaigns put some order in the evaluation process. They also release new datasets and evaluation metrics.

Besides the class dedicated to datasets, we will have a class dedicated to (evaluation) metrics

90's, 00's

- Advances in Machine Learning



Emergence of the feature-engineering approach!
Ex: SVM (1995)

We will dedicate some time to the (adorable) feature engineering approach!

Since 00's

- More and more computational resources become available
 - Some are more recent, but NLTK, for instance, started to be developed in 2001



NLTK

...



We will use NLTK, spaCy
and Stanza in the labs!

2011

- WATSON (IBM) wins Jeopardy!

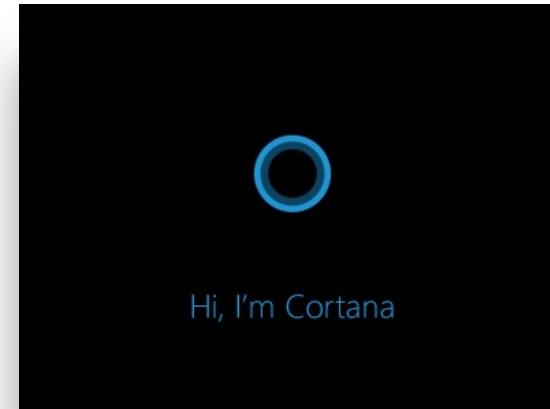
Photo: IBM



“AI wow” (in a long time)

SINCE 2011

- Virtual Assistants started to emerge



2012

- Meet Edgar Smith at the Monserrate Palace (by INESC-ID). It wasn't a global event, but it was really cool!



SINCE 2013

- Deep Learning
 - neural word embeddings
 - Transfer Learning
 - Multi-task Learning
 - ...

We will dedicate many classes to this!



2016

- Chatbots
(again)



Zuckerberg's Facebook Messenger launches 'chat bots' platform

Jessica Guynn | USA TODAY
Published 5:02 PM EDT Apr 12, 2016

SAN FRANCISCO — You already chat with friends on Facebook Messenger. Now Messenger wants you to chat with businesses and get updates from them, too.



Image from <https://eu.usatoday.com/story/tech/news/2016/04/12/facebook-messenger-f8-chat-bots/82919056/>

SINCE 2016 (MAYBE BEFORE)

- Several tools to build chatbots become available
 - Although they called themselves “natural language understanding” frameworks, they “only” take a sentence as input and:
 - Find the intention behind it (**classification**)
 - **Extract named entities**
 - Do something with this information.
 - Some have a module to define the **dialogue flow** (usually handcrafted)



Amazon Lex



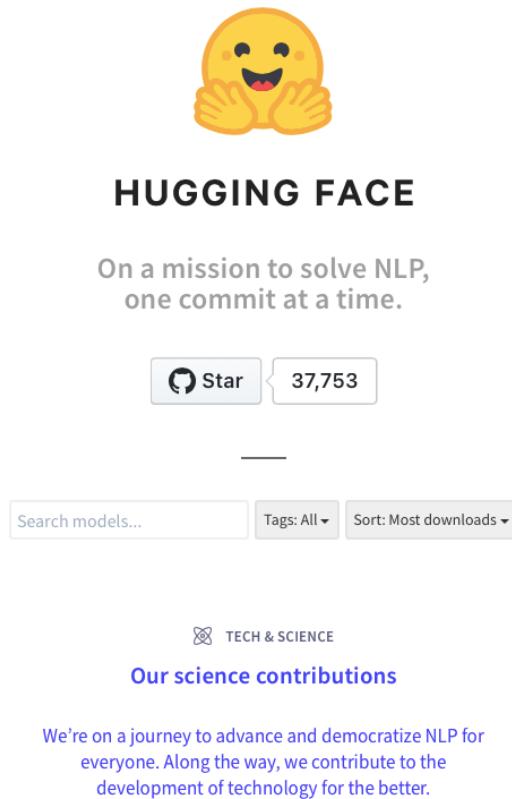
Dialogflow



We will study several classification tasks in NLP and also Named Entity Recognition, which is an NLP task

2016

- More computational stuff (mainly deep learning models)
 - Hugging Face is an NLP-focused startup with a large open-source community



SINCE 2016

- Shared Tasks/Prizes
 - [...] new benchmark [...] requiring a single system to perform ten disparate natural language tasks [...]



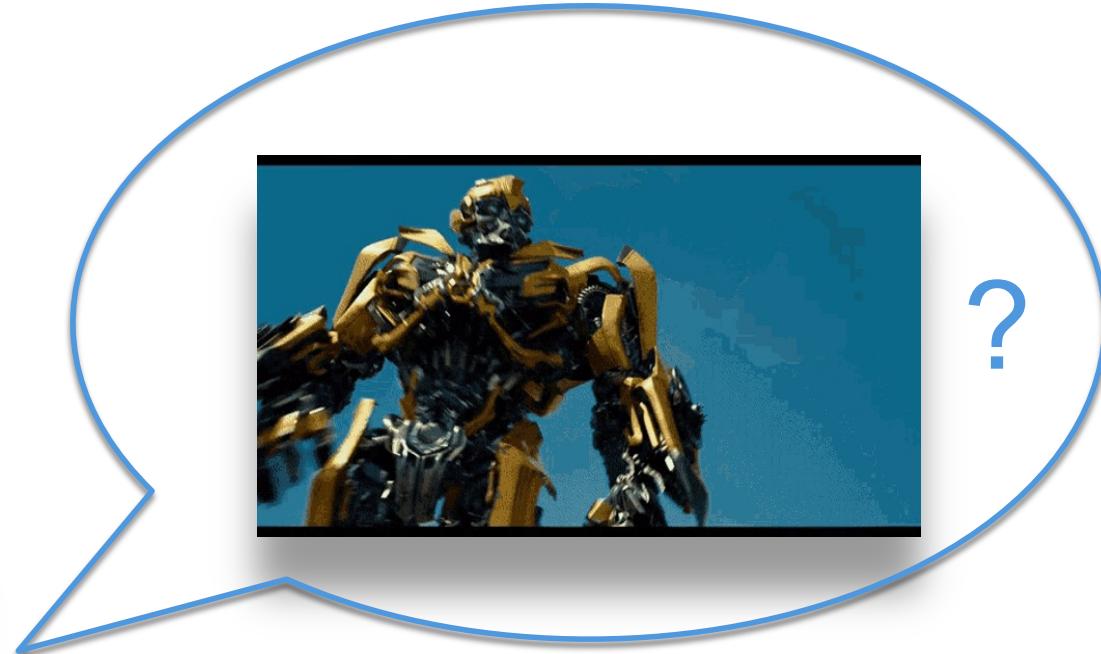
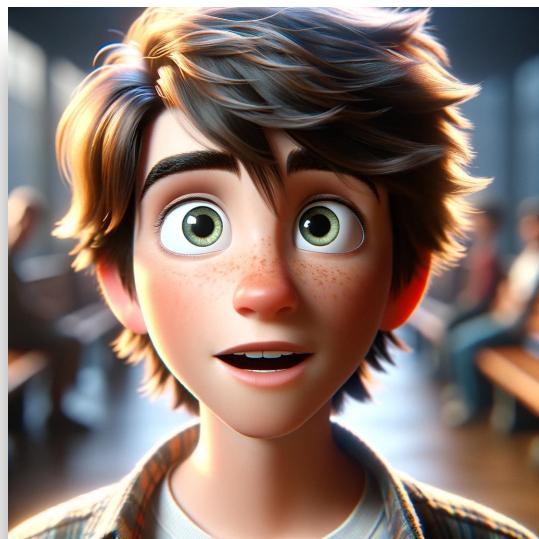
- \$500,000 prize will be awarded to the team that creates the best socialbot. The second- and third-place [...] \$100,000 and \$50,000, respectively.



A Portuguese team from NOVA got the first places twice in the Alexa competitions!

2017

- Transformers

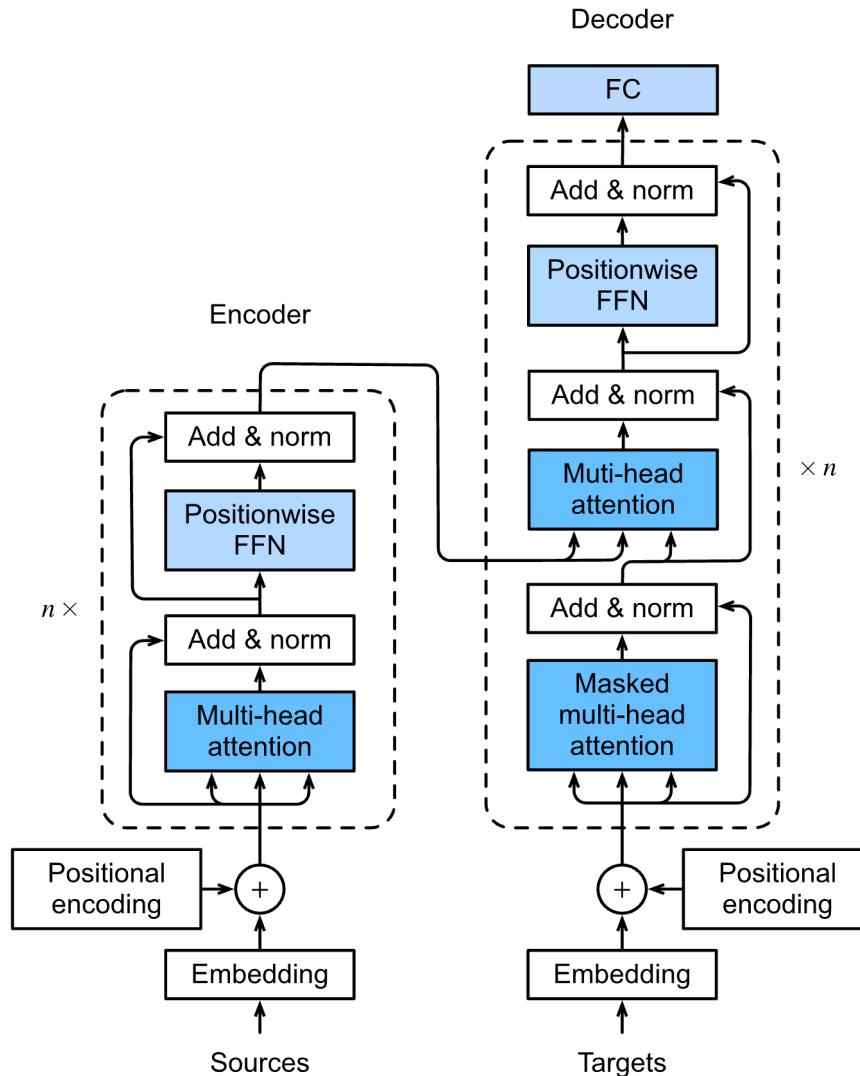


NOPE!
Not those transformers...

2017

- Transformers are a successful deep learning architecture
- Transformers are behind most of the nowadays Large Language Models (LLMs)

We will study this architecture



IN THE MEANWHILE



CRAZY CONFERENCE NUMBERS

CRAZY CONFERENCE NUMBERS

LONG PAPERS ACL

Year	Submitted	Accepted	Rate
1997	264	83	24%
1998 (w/COLING)	550	137	25%
1999	320	80	25%
2000	267	70	26.2%
2001	260	69	27%
2002	256	66	26%
2003	360	71	20%
2004	348	88	25%
2005	423	77	18%
2006 (w/COLING)	630	147	23%
2007	588	131	22.3%
2008	470	119	25%
2009	569	121	21%
2010	638	160	25%
2011	634	164	26%
2012	571	111	19%
2013	664	174	26%
2014	572	146	26.2%
2015	692	173	25.0%
2016	825	231	28.0%
2017	751	195	25%
2018	1045	256	24.5%
2019	1740	447	25.7%
2020	2244	779	22.7%

2021	3350	710 (long and short) for main conference + 457 in "Findings of ACL"	21.2% main / 37.5% findings
2022	3378	701 (long and short) for main conference + 361 in "Findings of ACL"	20.8% main / 31.4% findings

2024	4835	940 + 975 (findings)
2025	> 8300	1700 + 1392 (findings)

CRAZY COMPUTATIONAL COSTS

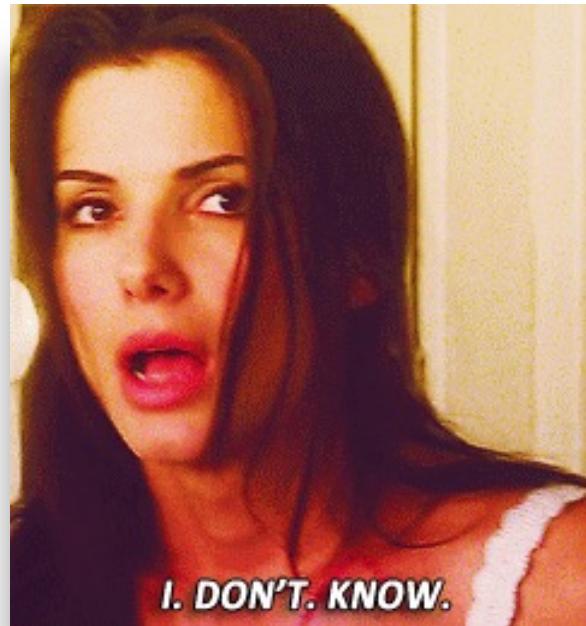
Google
Facebook
OpenAI
...



Rest of the world

NEW MODELS POPPING EVERYDAY

- In an online Transformers class:
chat: what about **decepticons?**



From wifflegif

2019



OpenAI

- GPT-2 (Open-AI)
 - Um modelo que usa transformers
 - O modelo não foi logo libertado por razões éticas

Legolas and Gimli advanced on the orcs, raising their weapons with a harrowing war cry.

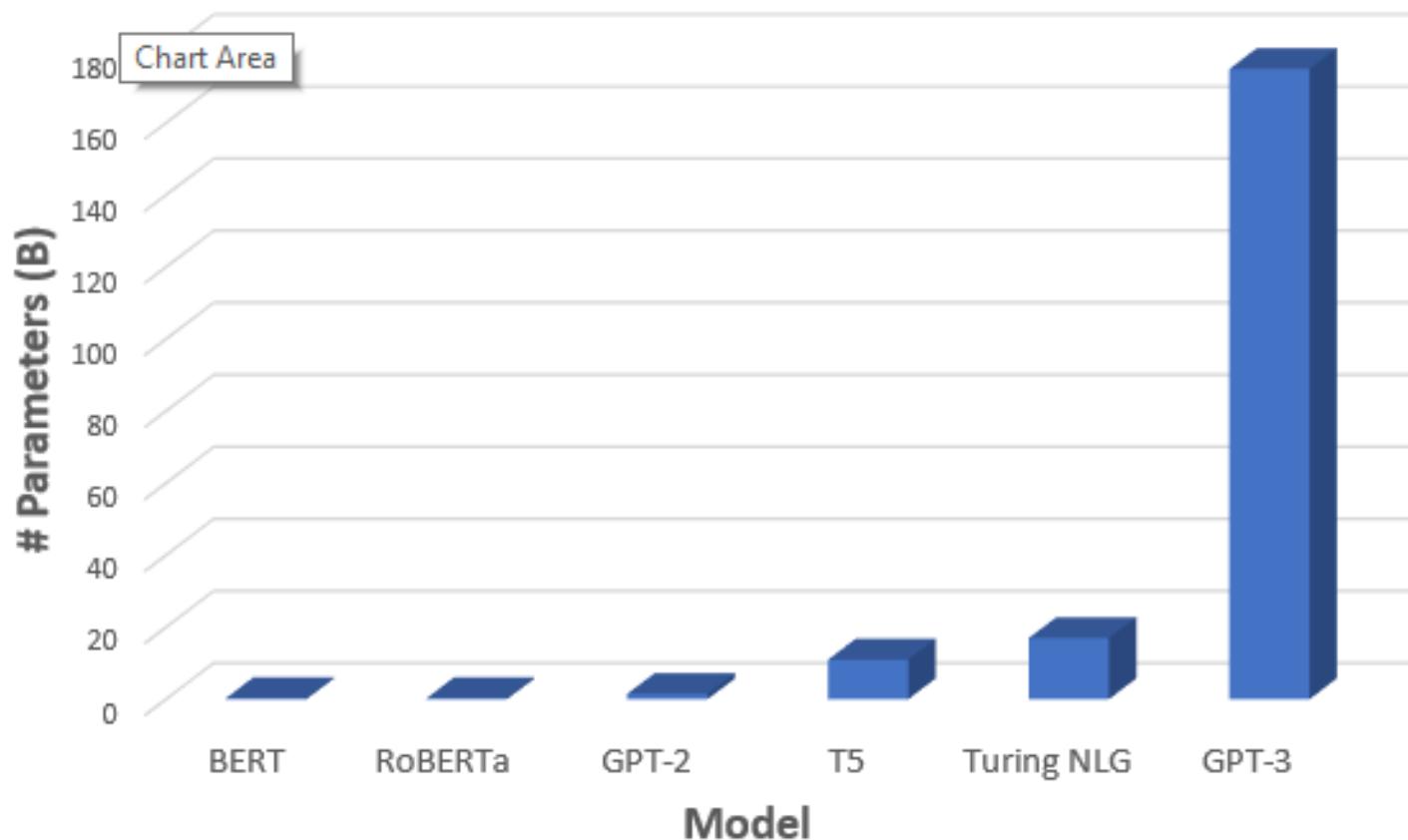
EXAMPLE

2020



OpenAI

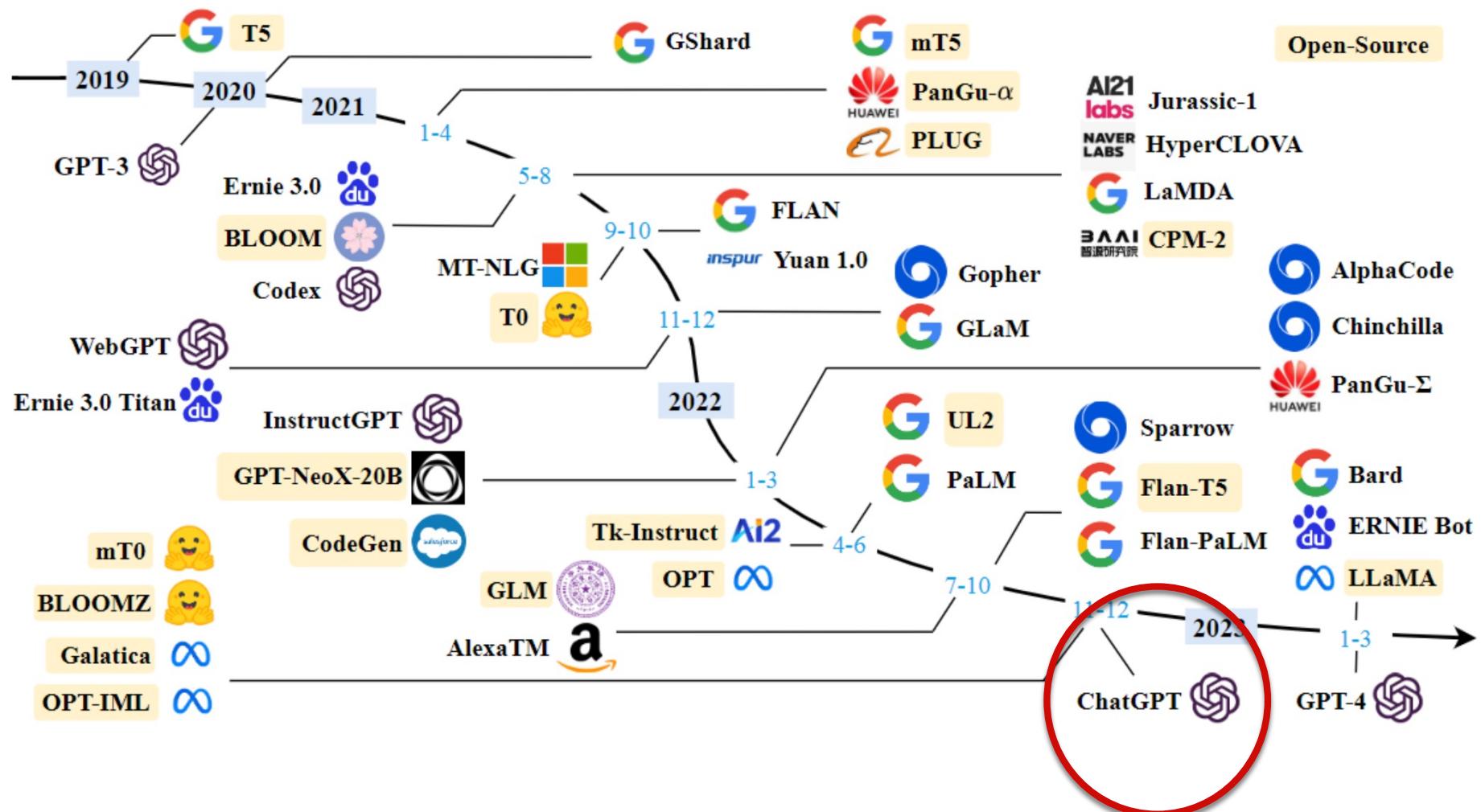
- GPT3
 - 175 mil milhões de parâmetros



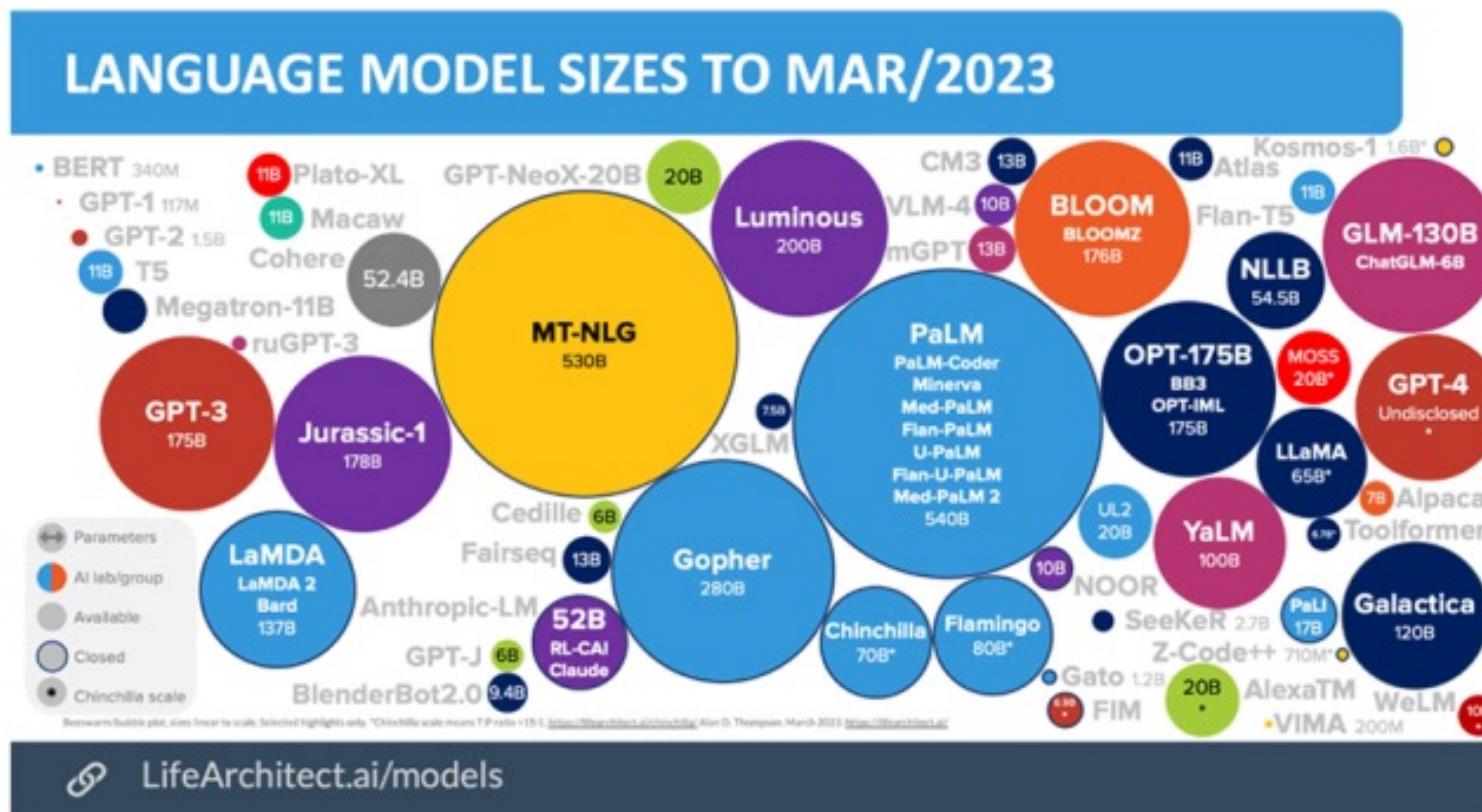
ABOUT GPT-3

- The cost of AI is increasing exponentially. A single training cycle for the 175 Billion parameter model takes about 355 years on a single V100 GPU, or around \$4,600,000 using on-demand Lambda Cloud GPU instances.
- (from <https://lambdalabs.com/blog/demystifying-gpt-3/>)

2022



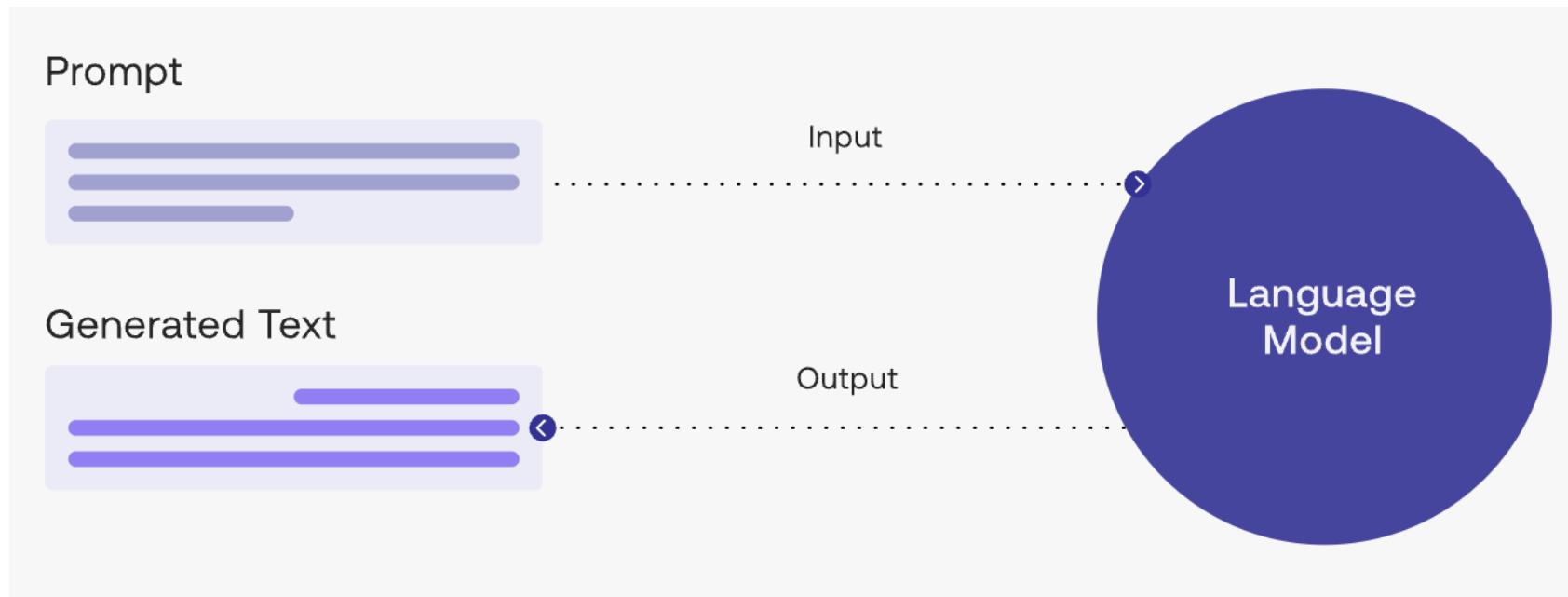
2023



2023

We will play with ChatGPT
and will talk about this

- A new job role has emerged: Prompt Engineer
- The job is dead some months after
 - LLM are also very good prompt engineers



2025

- This is how I see it!



KEY TAKEAWAYS

KEY TAKEAWAYS

- Many challenges, many different approaches (some taking advantage of linguistic knowledge), many different applications, many concepts to understand (see learning goals)
- Long process before the current achievements
- Everything is moving fast nowadays, and we are still trying to understand what we have done

MINI-MAP 1





CORPORA

Luísa Coheur

OVERVIEW

- Learning objectives
- Topics
 - Corpora: motivation, concept and examples
 - Building corpora
 - Agreement between annotators
 - Data splits
 - Data augmentation
 - Data cleaning
 - Toxic data
- Key takeaways
- Suggested readings

LEARNING OBJECTIVES

LEARNING OBJECTIVES

- After this class, students should be able to explain:
 - What is a corpus and how it should be analysed, cleaned and used in experiments
 - Several concepts, such as annotators' agreement or wizard of oz
 - Different types of toxic data
 - Different data augmentation processes

TOPICS

OVERVIEW

- Learning objectives
- Topics
 - Corpora: motivation, concept and examples
 - Building corpora
 - Agreement between annotators
 - Data augmentation
 - Data cleaning
 - Toxic data
 - Using Data: data splits
- Key takeaways
- Suggested readings

MOTIVATION

- The scientific method
 - Observation/Question: Identify a problem and formulate a [research question](#)
 - Hypothesis: Develop a [hypothesis](#) (testable and falsifiable prediction or tentative explanation that addresses the research question)
 - [Data Collection](#): Planning and gathering data
 - Experimentation: Designing and conducting [experiments](#) to test the hypothesis
 - Data Analysis: [check results](#), determine whether they support the hypothesis or not
 - Draw [conclusions](#)

MOTIVATION

- FACT: to train and test our models...



and sometimes
**ANNOTATED
DATA**

CORPORA

- A **corpus** is a collection of texts
 - Corpora (plural); corpus (singular)
- There are many, many, many corpora available with or without annotations
- There are many different types of annotations
 - Example:
 - Reviews are annotated as positive, neutral or negative
 - Words within a text are annotated with the correspondent morpho-syntactic category (verb, noun, etc.)
 - ...

**Sentiment
Analysis is an
NLP TASK!**

**PoS Tagging is an
NLP TASK (and we will
study how to do it)!**

CORPORA

- Some **companies/startups/whatever** gather/produce/sell corpora
 - Examples:
 - ELDA: Evaluations and Language Resources Distribution Agency
 - LDC: Linguistic Data Consortium
 - Kaggle
 - ...
 - Some **conferences** and **journals** are exclusively dedicated to NLP resources, including corpora
 - Examples:
 - Language Resources and Evaluation conference (LREC)
 - Language Resources and Evaluation journal

CORPORA: EXAMPLES

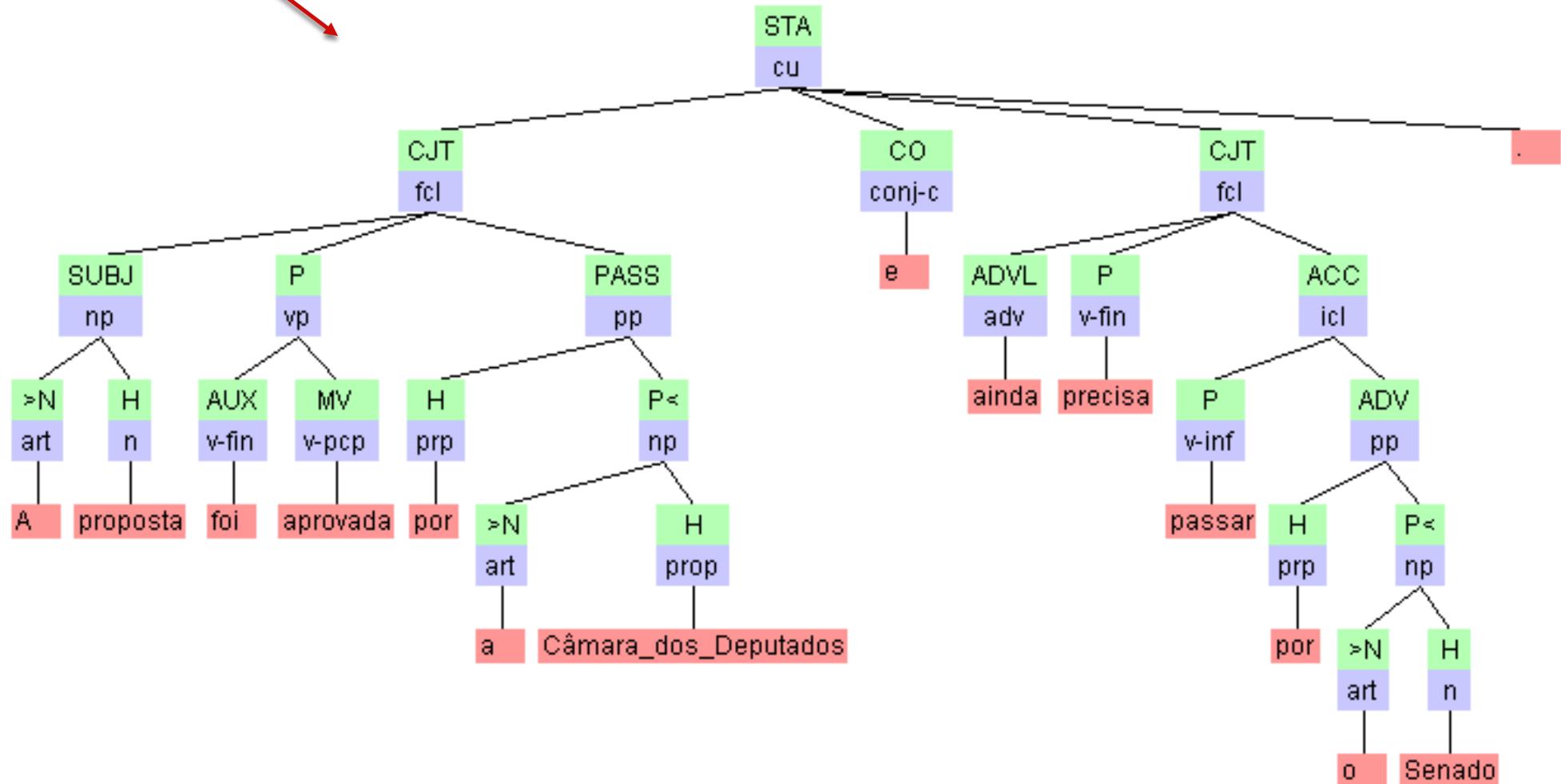
Annotations



What is fibromyalgia ?	DESC:def
What is done with worn or outdated flags ?	DESC:desc
What does cc in engines mean ?	DESC:def
When did Elvis Presley die ?	NUM:date
What is the capital of Yugoslavia ?	LOC:city
Where is Milan ?	LOC:city
What is the speed hummingbirds fly ?	NUM:speed
What is the oldest city in the United States ?	LOC:city
What was W.C. Fields ' real name ?	HUM:ind
What river flows between Fargo , North Dakota and Moorhead , Minnesota ?	LOC:other
What do bats eat ?	ENTY:food
What state did the Battle of Bighorn take place in ?	LOC:state
Who was Abraham Lincoln ?	HUM:desc
What do you call a newborn kangaroo ?	ENTY:termeq
What are spider veins ?	DESC:def
What day and month did John Lennon die ?	NUM:date
What strait separates North America from Asia ?	LOC:other

CORPORA: EXAMPLES

Annotations



Example from Floresta Sintática (Linguateca)

CORPORA: EXAMPLES

The screenshot shows the Linguistic Data Consortium (LDC) website. The header features the LDC logo and the text "Linguistic Data Consortium". A navigation menu on the left includes links for ABOUT, MEMBERS, COMMUNICATIONS, LANGUAGE RESOURCES (with a dropdown for Data), Obtaining Data, Catalog, By Year, Top Ten Corpora (which is highlighted in red), Projects, Search, Memberships, Data Scholarships, Tools, Papers, LR Wiki, DATA MANAGEMENT, and COLLABORATIONS. The main content area displays the "Top Ten LDC Corpora" list. The list includes:

LDC2013T19	OntoNotes Release 5.0
LDC93S1	TIMIT Acoustic-Phonetic Continuous Speech Corpus
LDC2006T13	Web 1T 5-gram Version 1
LDC96L14	CELEX2
LDC99T42	Treebank-3
LDC2008T19	The New York Times Annotated Corpus
LDC93S10	TIDIGITS
LDC97S02	Switchboard-1 Release 2
LDC2006T06	ACE 2005 Multilingual Training Corpus
LDC2011T07	English Gigaword Fifth Edition

A red box highlights the entries for LDC2008T19 and LDC93S10. A red arrow points from the word "Example" to the LDC93S10 entry.

© 1992-2020 Linguistic Data Consortium, The Trustees of the University of Pennsylvania. All Rights Reserved.

CORPORA: EXAMPLES

Updates

As of April, 2015, TIDIGITS is also available in flac compressed wav. This package is available to licensees as an additional download. Not included in this version are the folders relating to handling the shortened sphere files of the original corpus.

Copyright

Portions © 1993 Trustees of the University of Pennsylvania

Available Media

- Web Download

Fees

\$0.00 1993 Member
\$500.00 Non-Member
\$250.00 Reduced-License
\$0.00 Extra Copy
[Login for the applicable fee](#)

CORPORA: EXAMPLES

Featured Code Competition

Quora Insincere Questions Classification

Detect toxic content to improve online conversations

Quora · 4,037 teams · 2 years ago

Overview Data Notebooks **Discussion** Leaderboard Rules New Topic



Dieter

22nd place

Augmentation for text

Posted in [quora-insincere-questions-classification](#) 2 years ago

117

Of course there are a lot of augmentation techniques for images, but what about text? Let's discuss some techniques:

1. Exchanging words with synonyms (see e.g. <https://arxiv.org/pdf/1502.01710.pdf>)
2. noising in RNN (<https://arxiv.org/pdf/1703.02573.pdf>)
3. Translation to other language and back (<https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/discussion/48038>)

CORPORA: EXAMPLES

The screenshot shows a dataset page for "Friends Series Dataset". At the top, there's a search bar, a "Sign In" button, and a "Register" button. Below the search bar, the title "Friends Series Dataset" is displayed in bold, with a red oval highlighting it. A subtitle "Data about all 236 episodes of Friends Series" follows. A profile picture of Mohammad Reza Ghari and the text "Mohammad Reza Ghari • updated 5 months ago" are shown. Below this, a navigation bar includes "Data" (which is underlined), "Tasks", "Notebooks (6)", "Discussion (1)", "Activity", and "Metadata". To the right are "Download (56 KB)", "New Notebook", and a more options menu. The main content area has sections for "Usability" (rating 10.0), "License" (Data files © Original Authors), and "Tags" (arts and entertainment). Below these are "Description", "Context", and "Content" sections. The "Context" section contains text about the dataset being useful for social interactions like votes and reviews. The "Content" section describes the dataset as having 235 rows and 8 columns representing IMDB-indexed episodes.

Search

Sign In Register

Friends Series Dataset

Data about all 236 episodes of Friends Series

Mohammad Reza Ghari • updated 5 months ago

Data Tasks Notebooks (6) Discussion (1) Activity Metadata Download (56 KB) New Notebook :

Usability 10.0 License Data files © Original Authors Tags arts and entertainment

Description

Context

Most of the times there are a lot of interesting insights behind the popular subjects online, specially the ones that are involved social interactions like votes, reviews and user-generated contents. Popular TV shows are in this category. Friends Sitcom TV show, is one of the most loved series and I found it useful to have a dataset online.

Content

This dataset consisting of 235 row each representing an episode of the show and 8 columns that are features of each episodes indexed on IMDB. I'll try to add more features in the future.

Overview

- Learning objectives
- Topics
 - Corpora: motivation, concept and examples
 - [Building corpora](#)
 - Agreement between annotators
 - Data splits
 - Data augmentation
 - Data cleaning
 - Toxic data
- Key takeaways
- Suggested readings

BUILDING CORPORA

- Time-consuming and expensive
- Sometimes **experts** are needed to label the data
- Sometimes **the crowd** is used
 - Check, for instance, Amazon Mechanical Turk
 - Check the work of the Portuguese DefinedCrowd
(currently Defined.ai)



DefinedCrowd®

DEFINED.AI

ACTIVE LEARNING MOMENT: THINK-PAIR-SHARE



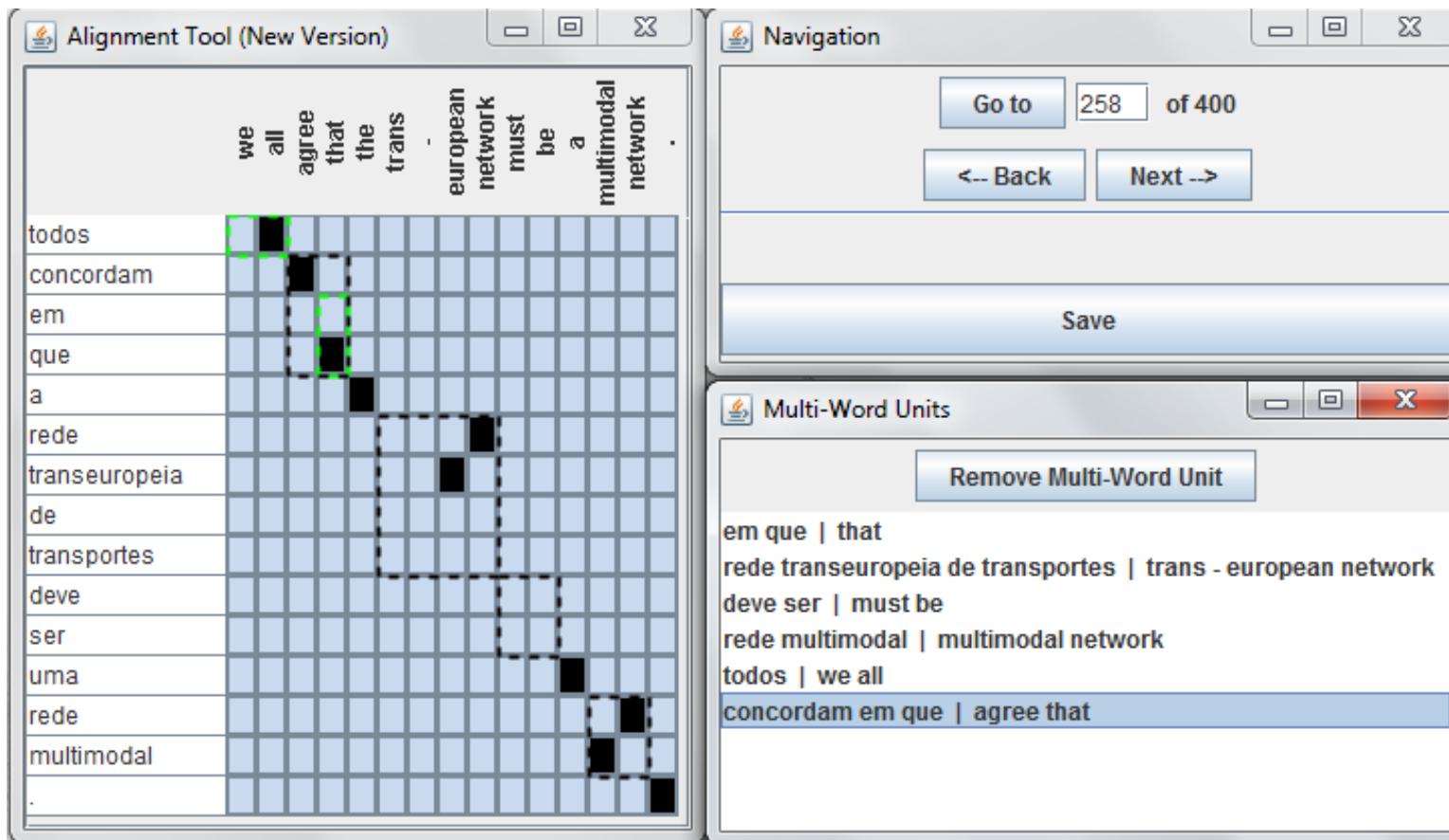
THINK-PAIR-SHARE



Find examples of NLP tasks that require annotated data where the annotators do not need to be experts

BUILDING CORPORA: EXAMPLES DATA ALIGNED WITH CLUE-ALIGNER (Anabela Barreiro's project)

Text alignment is an NLP TASK!



BUILDING CORPORA: EXAMPLES TEXT SEGMENTATION (Pedro Mota's PhD)

2: Kinematics: Describing motion.

Our first goal is understanding the motion of objects.

The first step is simple: merely DESCRIBING the motion of things.

1) We'll only talk about "particles": point like objects, whose structure is irrelevant.

2) We'll work in one dimension, e.g. a train moving back and forth on a straight track.

To describe motion, we need a few basic concepts, quantities, and definitions.

We'll use English language words but define them mathematically when possible.

You'll see that words like "velocity, acceleration, force, energy, momentum (which are often sloppy), are, in physics, totally distinct and well defined.

(A1, A2, A3)

1) POSITION: Where is the object?

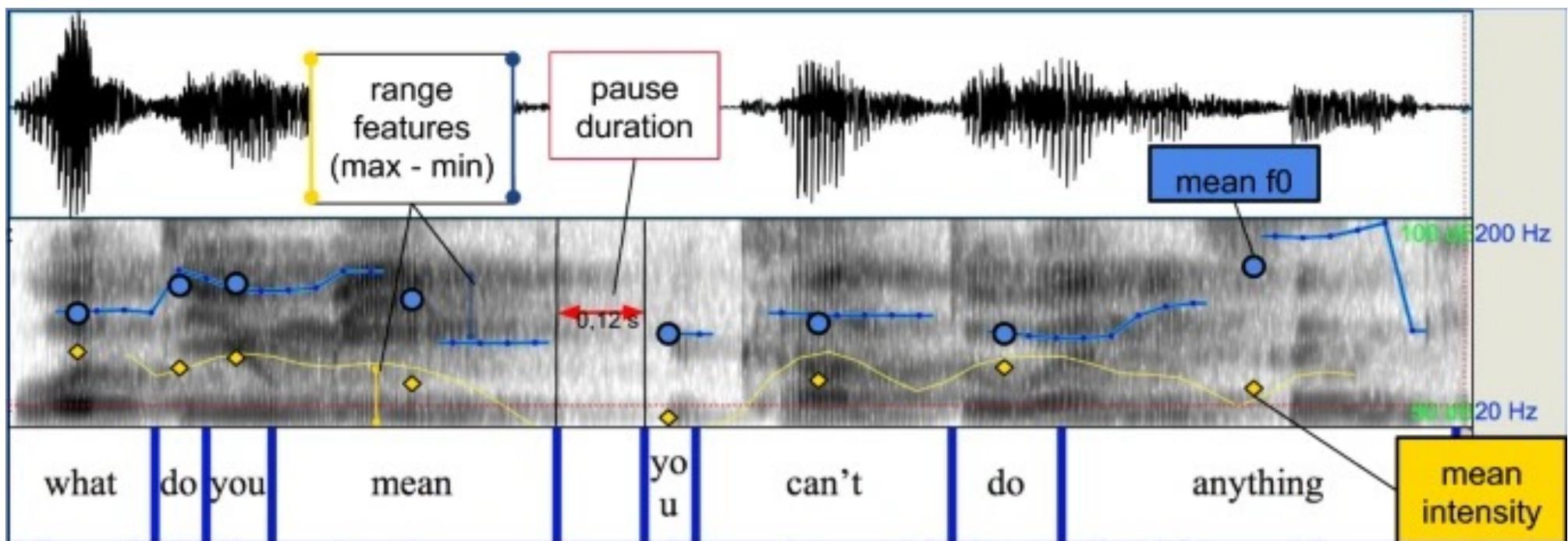
You need a reference frame to describe position.

A reference frame means a choice of axis and coordinate system: where is the origin, what units will we use to measure length, which direction will we call positive?

Text
segmentation is
also an
NLP TASK!

BUILDING CORPORA: EXAMPLES SPEECH RECOGNITION

Speech
Recognition: a
course in P4!



<https://link.springer.com/article/10.1007/s10579-021-09556-2>

BUILDING CORPORA: EXAMPLES SIGN LANGUAGES

Arquivo Editar Anotação Trilha Tipo Buscar Visualizar Opções Janela Ajuda

The screenshot shows a video editing application window. At the top, there's a menu bar with Portuguese labels: Arquivo, Editar, Anotação, Trilha, Tipo, Buscar, Visualizar, Opções, Janela, and Ajuda. Below the menu is a toolbar with various icons for video editing. The main area consists of two main sections: a video preview on the left and a control panel on the right.

Control Panel (Right Side):

- Volume:** A slider scale from 0 to 100, currently set at 100.
- Video_140_294.mp4:** A second volume slider for the video track, also set at 100. It includes options for Mute and Solo.
- Velocidade:** A slider scale from 0 to 200, currently set at 70.

Video Preview (Left Side):

A woman with dark hair, wearing a blue t-shirt with a "Love" graphic, is standing in front of a chalkboard. She is gesturing with her right hand. The timestamp at the bottom of the preview is 00:00:01.260.

Timeline and Transcription (Bottom):

The timeline shows several tracks labeled on the left:

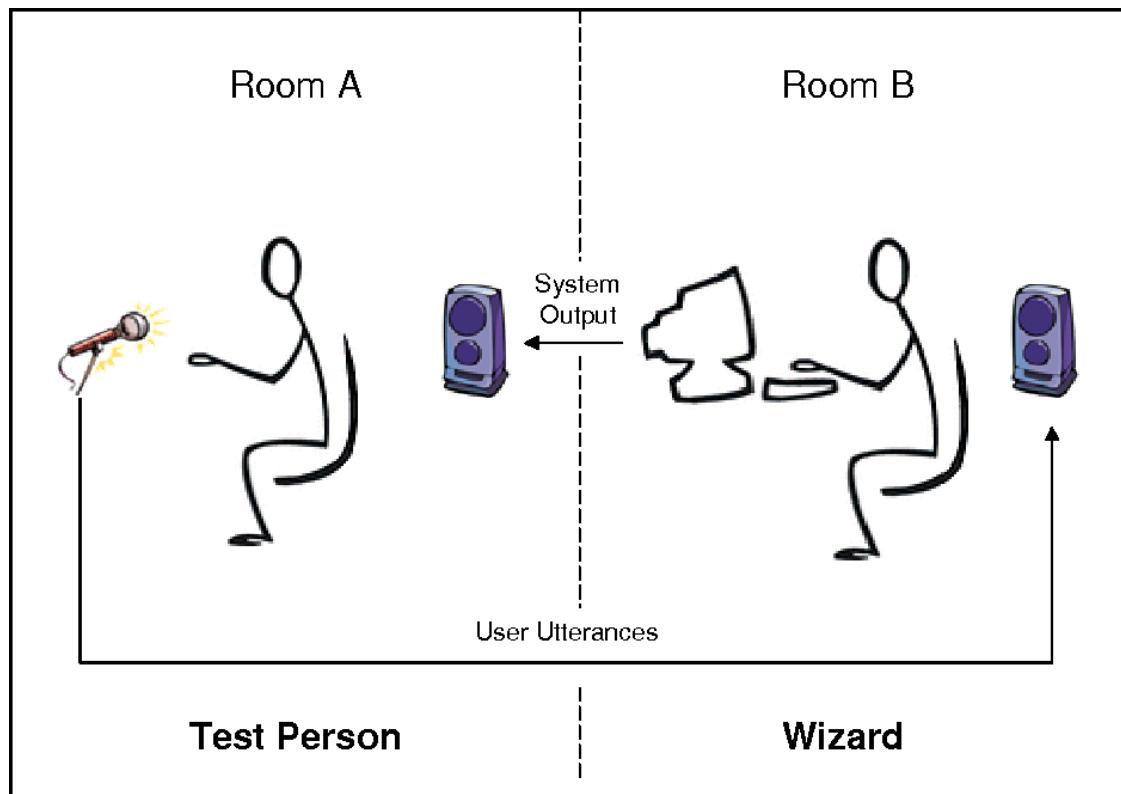
- LP_P1 transcrição livre [37]
- Come_P1Literal [29]
- LGP_P1Trans_Literal [101]
- GLOSAS_P1 [456]
- GLOSA_P1-M1 [451]
- GLOSA_P1-M2 [159]
- M2_ClassGram [144]
- M1_ClassGram

On the right side of the timeline, there are several text entries corresponding to the transcription tracks:

- Cultura, Arte, Teatro.
- CULTURA ARTE TEATRO
- CULTURA
- CULTURA

BUILDING CORPORA: EXAMPLES AND IF WE HAVE NO CORPUS?

- If you don't have data that allows you to understand how your system will be used, try a [Wizard of Oz](#).



Wizard of Oz: experiment in which subjects interact with a computer that they believe to be autonomous, although it is being operated by an unseen human

Overview

- Learning objectives
- Topics
 - Corpora: motivation, concept and examples
 - Building corpora
 - [Agreement between annotators](#)
 - Data splits
 - Data augmentation
 - Data cleaning
 - Toxic data
- Key takeaways
- Suggested readings

INTER-ANNOTATOR AGREEMENT

- How good are the annotations?
 - A way to check their quality is to see the **agreement among annotators**. If the agreement is very low:
 - We have bad guidelines, or
 - The task is very difficult, or
 - The annotators did not pay much attention to their job
- Notice that if two humans don't agree... well... the machine will not work a miracle

INTER-ANNOTATOR AGREEMENT

- Different tasks need different metrics for annotator's agreement
- For each task, several metrics might exist
- Examples:
 - Cohen's kappa coefficient (two annotators), ← Sebenta and now
 - Fleiss Kappa (several annotators),
 - Window Difference (ex: text segmentation)
 - ...

ACTIVE LEARNING MOMENT



EXERCISE: ANNOTATORS AGREEMENT

- LABELS:
 - GEOGRAPHY, MUSIC, LITERATURE, HISTORY, SCIENCE
- Guidelines:
 - GOAL: Tag questions with the given labels according with their main focus
 - Examples:
 -  Stevie Nicks: "Edge Of ___" Seventeen
 -  Germany has Worms & this country that borders Germany has a district called Wormerveer
The Netherlands (Holland)
 -  In this 17th C. novel, Sancho Panza is ...
Don Quixote

EXERCISE: ANNOTATORS AGREEMENT

1. Annotate the corpus in the next slide (use your knowledge!)
2. Choose a colleague near you
3. Go to:



4. Calculate the agreement between you and your colleague
5. Discuss the results

TAGS: GEOGRAPHY, MUSIC, LITERATURE, HISTORY, SCIENCE



1	This book by Virginia Woolf inspired Michael Cunningham's novel "The Hours"	Mrs. Dalloway
2	The "amiable" former name of the Tongan archipelago	the Friendly Islands
3	The Rhine Valley occupies one-third of this 62-square-mile country; the Alps cover the rest	Liechtenstein
4	PBS fans know that "Evening at Pops" refers to this city's Pops	Boston
5	In 1996 he simultaneously published "The Regulators" as Richard Bachman & "Desperation" under this name	Stephen King
6	In 1843 Congress allocated \$30,000 to string one between Baltimore & Washington; it was completed in 1844	a telegraph wire
7	According to Chuck Jones, whenever possible, this force of nature was to be Wile E. Coyote's greatest enemy	gravity
8	This 1940 Disney film featured the music of Bach, Beethoven, Stravinsky, Schubert & Mussorgsky	Fantasia
9	The Babylonians kept abreast of the times using a form of this instrument seen here:	Sundial
10	Dying in 2009 at age 113, British WWI vet Henry Allingham was the last original surviving member of this group, formed 1918	the Royal Air Force
11	-273 Celsius	absolute zero
12	Perhaps the greatest violinist ever, this Italian could play a whole piece on just one string	Niccolo Paganini
13	Blink-182: "That's about the time she walked away from me, nobody likes you when you're ___"	23
14	Rolf Gruber & Mother Abbess of Nonnberg Abbey	The Sound of Music
15	In this song, David Bowie instructs, "Put on your red shoes and dance the blues"	Let's Dance
16	Longfellow wrote, "Tell me not" that "life is but an empty" this	dream
17	His 1543 book "Concerning the Revolutions of the Celestial Spheres" started an astronomical revolution	Nicholas Copernicus
18	Baby, boudoir & concert are 3 sizes of this type of piano	grand piano
19	Welcome MCR, this alt-rock group, to "The Black Parade", its 2006 concept album	My Chemical Romance
20	Iron filings are often used to demonstrate the presence of this field	magnetic field

Annotate and find the agreement **between you and your colleague**

EXERCISE: ANNOTATORS AGREEMENT

	ME					
	GEO	MUS	LIT	HIST	SCI	TOTAL
GEO	2					2
MUS	1	7				8
LIT			2	1		3
HIST				2		2
SCI			1		4	5
TOTAL	3	7	3	3	4	*20*

REFERENCE

1	LITERATURE
2	GEOGRAPHY
3	GEOGRAPHY
4	MUSIC
5	LITERATURE
6	HISTORY
7	SCIENCE
8	MUSIC
9	SCIENCE
10	HISTORY
11	SCIENCE
12	MUSIC
13	MUSIC
14	MUSIC
15	MUSIC
16	LITERATURE
17	SCIENCE
18	MUSIC
19	MUSIC
20	SCIENCE

Quantify agreement with kappa results

	A	B	C	D	E	Total
A	2	0	0	0	0	2
B	1	7	0	0	0	8
C	0	0	2	1	0	3
D	0	0	0	2	0	2
E	0	0	1	0	4	5
Total	3	7	3	3	4	20

Number of observed agreements: 17 (85.00% of the observations)

Number of agreements expected by chance: 4.9 (24.25% of the observations)

Kappa= 0.802

SE of kappa = 0.103

95% confidence interval: From 0.601 to 1.000

The calculations above only consider exact matches between observers. If the categories (A, B, C...) are ordered, you may also wish to consider close matches. In other words, if one observer classifies a subject into group B and the other into group C, this is closer than if one classifies into A and the other into D. The calculation of weighted kappa, below, assumes the categories are ordered and accounts for how far apart the two raters are. This calculation uses linear weights.

EXERCISE: ANNOTATORS AGREEMENT

- Discussion:
 - His 1543 book "Concerning the Revolutions of the Celestial Spheres" started an astronomical revolution
 - LITERATURE? SCIENCE? HISTORY?
 - Unbalanced corpus (8 MUSIC in 20)
 - Consequences?

Generated by DALL-E



Overview

- Learning objectives
- Topics
 - Corpora: motivation, concept and examples
 - Building corpora
 - Agreement between annotators
 - [Data splits](#)
 - Data augmentation
 - Data cleaning
 - Toxic data
- Key takeaways
- Suggested readings

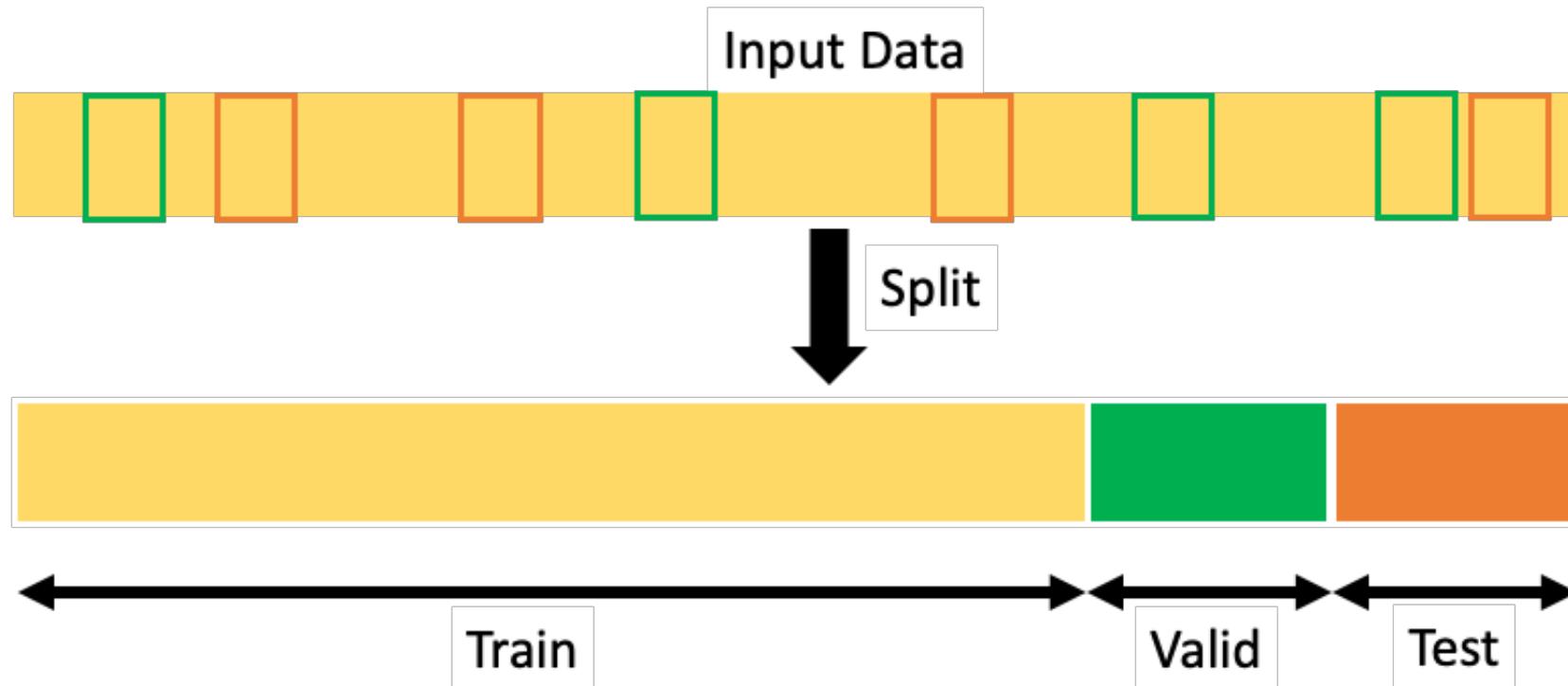
DATA SPLITS

- **Train set:** used to train the model
- **Test set:** used to evaluate the performance of the model after training
 - Tests the model's ability to generalize to new, unseen data
 - Results on the test set gives an indication of how the model will perform with real-world data
 - The test set is **NOT** used during training
 - **Data hygiene** (keeping the training data separate from the test) in the case of large language models (LLM) can be difficult. How to guarantee the separation train/test?

DATA SPLITS

- **Validation set:** used during the tuning process
 - It helps adjust hyperparameters (e.g., learning rate, depth, regularization) **without touching the test set**
 - It must remain independent from the test set to ensure an unbiased final evaluation
 - It is **not** used for training the model's weights
 - **Development set:** used like a validation set, but sometimes also for preliminary assessment of the models
 - Sometimes development set and validation set are used interchangeably

DATA SPLITS



From <https://towardsdatascience.com/how-to-split-data-into-three-sets-train-validation-and-test-and-why-e50d22d3e54c>

DATA SPLITS

- **Reference:** refers to a set of data used as a benchmark or standard when comparing different models or when validating the outputs of a model
- **Gold standard or gold collection:** dataset that has been meticulously curated and is of highest quality. It often serves as a benchmark

DATA SPLITS

- Data Split
 - Usually: 90% train – 10% test or 80% train – 20% test
 - BigData: 99% train – 1% test
- Validation/development (dev) set:
 - Usually 10% of the training set

DATA SPLITS

- K-fold cross validation
 - Divide the dataset into K equal parts (folds/splits) and run K experiments
 - Each fold is used once as a test set while the remaining K-1 folds form the training set
 - For instance, for K = 10
 - divide the corpus in 10 parts (randomly. Why?)
 - train your system with 9 parts
 - evaluate in the remaining one
 - average over the rounds at the end
- Notice that in the deep learning days it might be complicated to train K models

Overview

- Learning objectives
- Topics
 - Corpora: motivation, concept and examples
 - Building corpora
 - Agreement between annotators
 - Data splits
 - **Data augmentation**
 - Data cleaning
 - Toxic data
- Key takeaways
- Suggested readings

DATA AUGMENTATION

- Some techniques:
 - Synonym replacement: substitutes words in sentences with their *synonyms*
 - Paraphrasing: rewrite sentences or paragraphs differently
 - Rule-based augmentation: apply linguistic rules to sentences, such as changing the voice from active to passive
 - Example: The student wrote the report (active voice) vs. The report was written by the student (passive voice)
 - Back translation: translates text to another language and then back to the original language
 - Random swap: randomly swaps the position of words within sentences to create slight variations.

DATA AUGMENTATION

- Some techniques (cont.):
 - Text expansion: enrich the content with additional relevant text, such as explanatory clauses or descriptive phrases
 - Noise injection: introduce typos, spelling mistakes, or grammatical errors to mimic real-world imperfections in text
 - Entity substitution: replace named entities (like names, locations, and organizations) with other entities of the same type

ACTIVE LEARNING MOMENT: THINK-PAIR-SHARE



THINK-PAIR-SHARE

Try to match the following sentences with the possible techniques – [Synonym replacement](#), [Paraphrasing \(rule-based, back translation, random swap\)](#), [text expansion](#), [noise injection](#), [entity substitution](#) – used to generate them from [*Princess Mary entered the palace*](#):

1. Princess Marry entered the palace, the royal residence
2. Into the palace, princess Mary entered
3. Princess Mary walked into the palace
4. Princess Mary entered the place
5. The royal lady Mary made her way into palace
6. Princess Mary has entered the palace
7. Princess Joana entered the palace

Overview

- Learning objectives
- Topics
 - Corpora: motivation, concept and examples
 - Building corpora
 - Agreement between annotators
 - Data splits
 - Data augmentation
 - Data cleaning
 - Toxic data
- Key takeaways
- Suggested readings

DATA CLEANING

- Data cleaning (= denoising) involves the detection and rectification of errors and inconsistencies in datasets

Overview

- Learning objectives
- Topics
 - Corpora: motivation, concept and examples
 - Building corpora
 - Agreement between annotators
 - Data splits
 - Data augmentation
 - Data cleaning
 - Toxic data
- Key takeaways
- Suggested readings

THINK-PAIR-SHARE

How hard can be the work of annotators of toxic datasets?

TOXIC DATA

- Toxicity in NLP: various forms of harmful, offensive, or inappropriate content that can manifest in text
- Examples:
 - Profanity: use of swear/curse words
 - Threats, Insults:
 - Cyberbullying: repeated online behaviours that intimidate or upset individuals

TOXIC DATA

- Examples (cont.):
 - **Misinformation and disinformation:** the spread of false or misleading information, either unintentionally (misinformation) or deliberately (disinformation)
 - **Stereotyping and generalizations:** statements that apply a generalized belief or opinion to all members of a group
 - **Hate speech:** communication that demeans a person or group based on characteristics such as race, religion, ethnic origin, sexual orientation, disability, or gender

ACTIVE LEARNING MOMENT: THINK-PAIR-SHARE



THINK-PAIR-SHARE

If I have a dataset that might contain hate speech,
whom should I hire?

Who should decide what is hate speech?

KEY TAKEAWAYS

KEY TAKEAWAYS

- Understand the importance of having corpora for NLP tasks
- Understand the importance of having good annotations
- Concepts associated with corpora, including the ones related with toxic language, and techniques of data augmentation

SUGGESTED READINGS

READINGS

- This slides
- Sebenta:
 - Methodology, corpora and evaluation
 - notice that it does not covers all these slides



EVALUATION

Luís Coheur

OVERVIEW

- Learning Objectives
- Topics
 - Prelude: comparing strings
 - Automatic evaluation
 - Baseline
 - Some evaluation metrics
 - Human evaluation
 - Automatic vs. Human Evaluation
 - Intrinsic vs. extrinsic evaluation
 - Evaluation forums/shared tasks
- Key takeaways
- Suggested readings

LEARNING OBJECTIVES

LEARNING OBJECTIVES

- After this class, the student should be able to:
 - Explain some evaluation measures widely used in NLP
 - Define several concepts, such as evaluation fora, human evaluation, intrinsic/extrinsic evaluations, etc.
 - Apply some similarity/distance/evaluation metrics

TOPICS

OVERVIEW

- Learning Objectives
- Topics
 - Prelude: comparing strings
 - Automatic evaluation
 - Baseline
 - Some evaluation metrics
 - Human evaluation
 - Automatic vs. Human Evaluation
 - Intrinsic vs. extrinsic evaluation
 - Evaluation forums/shared tasks
- Key takeaways
- Suggested readings

COMPARING STRINGS

- Comparing strings is at the basis of an NLP evaluation
 - We compare words, sentences, paragraphs and even documents
 - These comparisons might be lexical or semantic
 - Example:
 - "The discovery of DNA structure was a revolutionary achievement in science."
- vs.
- "The revelation of the DNA structure was a revolutionary milestone in science."

COMPARING STRINGS

- Comparing strings can be useful in other tasks:
 - Spell Checking:
 - Example: "The brown fox jumped over the lazy dog."
 - Data Cleaning: Matching and merging duplicate records
 - Ex: "John Fitzgerald Kennedy", "Jonh F. Kennedy", "Jonh Kennedy" => "John Kennedy"
 - Plagiarism Detection:
 - Ex: "The discovery of DNA structure was a revolutionary achievement in science." vs "The revelation of the DNA structure was a revolutionary milestone in science."
 - Code Similarity Detection
 - ... (and many, many more)

BEFORE WE MOVE ON: DISTANCE VS. SIMILARITY

- Pay attention to the metric you use: some are similarity metrics, some are distance/difference metrics
- You may need to normalize them



COMPARING STRINGS

- Edit-based metrics allow to quantify how dissimilar two strings are to one another, by **counting the minimum number of operations** required to transform one string into the other
 - The **Levenshtein distance** is an Edit-based metric that calculates the minimum number of **insertions, deletions or substitutions** required to change one sequence into the other

```

 $C_1, C_2, C_3 \leftarrow 1$ 
if  $n = 0$  then
    return m
end if
if  $m = 0$  then
    return n
else
    Build matrix M, with  $m+1$  lines and  $n+1$  columns
     $j \leftarrow 1$ 
    while  $j \neq n + 1$  do
         $i \leftarrow 1$ 
        while  $i \neq m + 1$  do
            if  $s[i] = t[j]$  then
                 $M[i, j] = M[i - 1, j - 1]$  // Take the diagonal value if characters
                match
            else
                 $M[i, j] = \min(M[i - 1, j] + C_1, M[i, j - 1] + C_2, M[i - 1, j - 1] + C_3)$ 
            end if
             $i \leftarrow i + 1$ 
        end while
         $j \leftarrow j + 1$ 
    end while
    return M[m, n]
end if

```

	0	1	2
	•	M	E
0	•	0 ← 1 ← 2	
1	M	1	0 ← 1
2	Y	2	1

- An empty string
- ← DELETION OPERATION
- ↑ INSERTION OPERATION
- DO NOTHING (both letters are equal)
- ↖ SUBSTITUTION OPERATION

ACTIVE LEARNING MOMENT



EXERCISE

- What is the MED between MORDOR and LORD?

	M	O	R	D	O	R
0	1	2	3	4	5	6
1						
2						
3						
4						

EXERCISE

- What is the MED between MORDOR and LORD?

	M	O	R	D	O	R
0	1	2	3	4	5	6
L	1	1	2			
O	2	2	1			
R	3	3				
D	4	4				

EXERCISE

- What is the MED between MORDOR and LORD?

	M	O	R	D	O	R
0	1	2	3	4	5	6
L	1	1	2	3	4	5
O	2	2	1	2	3	4
R	3	3	2	1	2	3
D	4	4	3	2	1	2

MORE EDIT DISTANCES

- Other Edit-based metrics:
 - The [Longest Common Subsequence \(LCS\) distance](#) allows as operations only insertion and deletion, not substitution
 - The [Hamming distance](#) allows only substitution (it only applies to strings of the same length)
 - The [Damerau–Levenshtein distance](#) allows insertion, deletion, substitution, and the transposition (swapping) of two adjacent characters
 - The [Jaro distance](#) allows only transposition
- Just to name a few...

COMPARING STRINGS

- Jaccard and Dice are other metrics used to compare strings (but they operate on sets)
 - While the MED says how distant two strings are (the highest the value, the less similar they are), Jaccard and Dice are similarity metrics: the highest the value, the more similar they are
- Let us see Jaccard and Dice as sets and not bags
 - that is, no repetitions; a set does not have repeated elements
- Both are examples of token-based similarity-metrics
- There are many, many more token-based metrics

COMPARING STRINGS

- $\text{Jaccard}(s, t) = |s \cap t| / |s \cup t|$
- $\text{Dice}(s, t) = 2 \times |s \cap t| / (|s| + |t|)$
- ...
- $\text{Overlap}(s, t) = |s \cap t| / \min(|s|, |t|)$ ← just another example
- ...

Remember: consider s and t as sets!
(each token appears only once)

COMPARING STRINGS

- Notice that you can pre-process strings and compare them afterwards
- For instance, you can “translate” your string to a form that represents how it sounds

COMPARING STRINGS (NOW AT THE SOUND LEVEL)

- Soundex (but also others, such as Metaphone, ...)

1. Retain the first letter of the name;
2. Drop all occurrences of a, e, i, o, u, y, h, w (unless they appear in the first position).
3. Replace consonants by digits, as follows (after the first letter):

(a) b, f, p, v → 1	Luisa
(b) c, g, j, k, q, s, x, z → 2	L
(c) d, t → 3	Ls
(d) l → 4	L2
(e) m, n → 5	L200
(f) r → 6	
4. Two adjacent letters with the same number are coded as a single number (ex: 55 → 5)
5. Continue until you have one letter and three numbers. If you run out of numbers, add zeros until there are three numbers (ex: L2 → L200); if you have too much numbers drop them after the third one (ex: L2345 → L234).

ACTIVE LEARNING MOMENT



SOUNDEX – EXAMPLE

- Can soundex help you to understand the following joke?
 - How does superman likes his milk?
 - Claro quente (vs Clark Kent)

1. Retain the first letter of the name;
2. Drop all occurrences of a, e, i, o, u, y, h, w (unless they appear in the first position).
3. Replace consonants by digits, as follows (after the first letter):
 - (a) b, f, p, v → 1
 - (b) c, g, j, k, q, s, x, z → 2
 - (c) d, t → 3
 - (d) l → 4
 - (e) m, n → 5
 - (f) r → 6

C460 Q530 vs C462 K530
+ Levenshtein (for instance)
And if it was Claroquente vs. Clarkkent?
4. Two adjacent letters with the same number are coded as a single number (ex:
 $55 \rightarrow 5$)
5. Continue until you have one letter and three numbers. If you run out of numbers, add zeros until there are three numbers (ex: L2 → L200); if you have too much numbers drop them after the third one (ex: L2345 → L234).

COMPARING STRINGS

- We have seen how to apply these metrics to words
 - MED(Monserrate, Moncerrate) = ...
 - Jaccard(Olá, Ola) = ...
 - ...
 - But we can also apply them to sentences (although they are not very effective):
 - MED("Niagara Falls is viewed by thousands of tourists every year.", "Each year, thousands of people visit Niagara Falls.")
 - Jaccard("She was a successful author and speaker.", "She found success as a public speaker and writer.")
 - ...
- Can be used to detect paraphrases!
(at this moment, still at the lexical level – no semantics)

COMPARING SENTENCES DRAWBACKS

(example from Read more: <http://www.city-data.com/forum/writing/1115620-two-sentences-have-same-words-but-2.html>)

- Only he told his mistress that he loved her. (Nobody else did)
- He only told his mistress that he loved her. (He didn't show her)
- He told only his mistress that he loved her. (Kept it a secret from everyone else)
- He told his only mistress that he loved her. (Stresses that he had only ONE!)
- He told his mistress only that he loved her. (Didn't tell her anything else)
- He told his mistress that only he loved her. ("I'm all you got, sweetie--nobody else wants you.")
- He told his mistress that he only loved her. (Not that he wanted to marry her.)
- He told his mistress that he loved only her. (Yeah, don't they all...).
- He told his mistress that he loved her only. (Similar to above one).

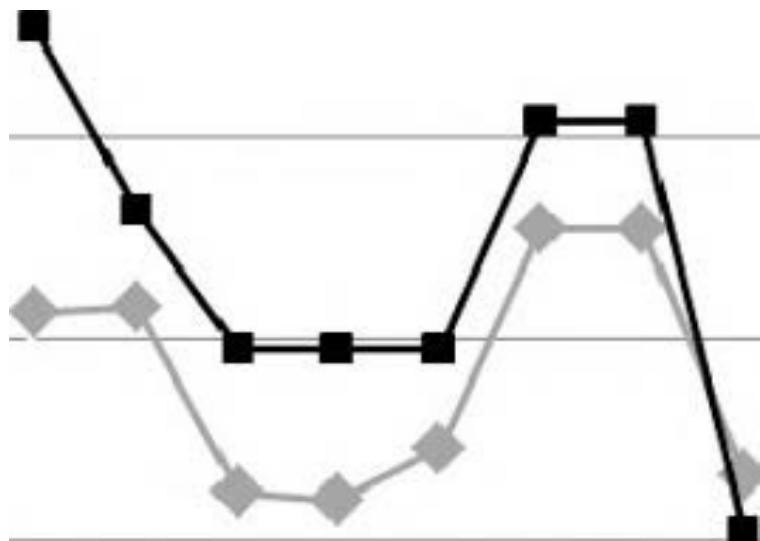
MED is different, but considering the other measures we talked about, they all end in the same sets!

OVERVIEW

- Learning Objectives
- Topics
 - Prelude: comparing strings
 - **Automatic evaluation**
 - Baseline
 - Some evaluation metrics
 - Human evaluation
 - Automatic vs. Human Evaluation
 - Intrinsic vs. extrinsic evaluation
 - Evaluation forums/shared tasks
- Key takeaways
- Suggested readings

AUTOMATIC EVALUATION

- Baseline:
 - a known starting configuration(s) against which results are compared



Hopefully:
Baseline (grey) vs Your System

You should always compare your system with others (at least with a (random) baseline)!!!

AUTOMATIC EVALUATION

- Metrics:
 - Precision (and macro- and micro-precision)
 - Recall (and macro- and micro-recall)
 - Accuracy
 - F-measure and F1-measure – use Precision and Recall

AUTOMATIC EVALUATION

- Consider:
 - True Positives (TP)
 - True Negatives (TN)
 - False Positives (FP)
 - False Negatives (FN)
- Precision (P) = $TP/(TP+FP)$
- Recall (R) = $TP/(TP + FN)$
- $F1 = 2PR/(P+R)$

ACTIVE LEARNING MOMENT



The capital of Portugal, Lisbon (Portuguese: Lisboa) has experienced a renaissance in recent years, with a contemporary culture that is alive and thriving and making its mark in today's Europe. Lisbon lacks a defined “~~downtown~~”, but the vast Praça do Comércio, facing the river at the base of the pedestrianized grid of Baixa (lower town), occupies a central position. Further northwest from Baixa stretches Lisbon's “Main Street”, Avenida da Liberdade, a broad boulevard resplendent in leafy trees, chic hotels and upscale shops, terminating at the circular Praça de Marques de Pombal. To the east are old neighborhoods of Mouraria and Alfama, both relatively spared during the Great Earthquake (as they are on a firmer rock) and therefore both retaining the charm of the winding alleys and azulejo-covered crumbling walls (further north lie relatively boring residential quarters).

Named Entity Recognition is an NLP TASK!

$$\text{Precision (P)} = \text{TP}/(\text{TP}+\text{FP})$$
$$\text{Recall (R)} = \text{TP}/(\text{TP} + \text{FN})$$
$$F_1 = 2\text{PR}/(\text{P}+\text{R})$$

- Reference:
Portugal, Lisbon, Lisboa, Europe, Praça do Comércio, Baixa, Avenida da Liberdade, Praça de Marquês de Pombal, Mouraria, Alfama
- System A: Portugal, Lisboa, Avenida da Liberdade, Alfama
- System B: Portugal, Lisbon, Portuguese, Lisboa, Praça do Comércio, Europe, Main Street, Alfama, Great Earthquake

Which is the best system?

- System A
 - $TP = 4$, $FP = 0$, $FN = 6$
 - $P = 4 / (4+0) = 1$
 - $R = 4/(4+6) = 0.4$
 - $F1 = 2PR/(P+R) = 2*0.4/1.4 = 0.57$
- System B
 - $TP = 6$, $FP = 3$ (Portuguese, Main Street, Great Earthquake), $FN = 4$
 - $P = 6 / (6+3) = 0.667$
 - $R = 6/(6+4) = 0.6$
 - $F1 = 0.63$



AUTOMATIC EVALUATION

- More metrics:
 - BLEU – Machine Translation
 - METEOR – Machine Translation
 - ROUGE – initially summarization
 - Perplexity – to evaluate Language Models
 - ...
 - COMET – trained measure (deep learning model)

Recently: LLMs are used to evaluate systems

- Most of these metrics are now used in generative tasks such as dialogue systems
 - And they are awful!

ACTIVE LEARNING MOMENT



EXERCISE

- Consider the evaluation of a
 - Sentiment analysis task
 - Humanity is great – positive/negative
 - Translation task
 - Humanity is great
 - A humanidade é maravilhosa
 - A humanidade é porreira
 - A humanidade é fantástica
 - Dialogue task
 - [You] Hi!
 - [Bot] Hi, how are you?
 - [You] Fine, and you?
 - [Bot] Miserable.

Remember the previous class?

- How should we build gold collections for these tasks?

Discuss the difficulty of evaluating a model that performs each task

AUTOMATIC EVALUATION

- To conclude, automatic evaluation:
 - Allows for more agile development cycles in NLP
 - Evaluate a vast number of language samples quickly and consistently, saving time and resources;
 - thus, reduces the need for expensive and time-consuming manual evaluation
 - Experiments are easily reproduced
 - Provide a uniform standard, reducing the subjectivity and potential bias that might come with (more) human evaluations;
 - Although the comparison is almost always against some human reference

AUTOMATIC EVALUATION

- But...
 - Some metrics are just lexical
 - Most of the metrics do not provide insights into subjective aspects like fluency, coherency, and readability
 - Sometimes do not correlate well with human evaluations

OVERVIEW

- Learning Objectives
- Topics
 - Prelude: comparing strings
 - Automatic evaluation
 - Baseline
 - Some evaluation metrics
 - Human evaluation
 - Automatic vs. Human Evaluation
 - Intrinsic vs. extrinsic evaluation
 - Evaluation forums/shared tasks
- Key takeaways
- Suggested readings

HUMAN EVALUATION

- Allows for:
 - **Nuanced understanding:** Humans can grasp context, irony, humor, and cultural references in ways that automated metrics cannot
 - **Quality assessment:** human judges can assess subjective qualities like readability, coherency, and engagement of the text
 - **Error identification:** while automatic metrics can indicate that an error has occurred, human evaluators can provide detailed insights into the nature of the error
 - **Ground truth benchmarking:** human judgment often serves as the gold standard

HUMAN EVALUATION

- But...
 - It is necessary to ensure evaluator expertise and consistency
 - Human evaluation is expensive
 - Human evaluation is time-consuming (and can be boring)



HUMAN EVALUATION

- Some types of human evaluation:
 - Direct Assessment:
 - Human evaluators rate the output on a fine-grained scale
 - Example:
 - 1-10, 1-100
 - Rank-Based Evaluation:
 - Multiple outputs are ranked in order of preference or quality
 - Enables evaluators to assess relative performance
 - Paired Comparison (a subcase of rank-based evaluation):
 - Human evaluators are presented with two outputs and asked to choose the better one

ACTIVE LEARNING MOMENT



EXERCISE

- Consider the following synopses of Harry Potter books:
 - S1: Harry go to magic school and do magic stuff, then he fight bad guy and win at the end.
 - S2: Harry goes to school, learns magic, and beats the villain.
 - S3: Harry embarks on a thrilling journey at Hogwarts, uncovering magical secrets and battling the dark forces of Voldemort in an epic struggle for the fate of the wizarding world.
- (1) Consider that you want to evaluate the English quality of each synopsis. Use a Direct Assessment (0-10).
- (2) Consider that you want to evaluate how engaging is each synopsis. Use a Rank-Based Evaluation.

BY THE WAY...



Connor McCartan

@MILKCARTAN_

 Follow

#ExplainAFilmPlotBadly; Noseless guy has an unhealthy obsession with a teenage boy



From: 64 Times People Explained Movies So Badly It Was Good

OVERVIEW

- Learning Objectives
- Topics
 - Prelude: comparing strings
 - Automatic evaluation
 - Baseline
 - Some evaluation metrics
 - Human evaluation
 - [Automatic vs. Human Evaluation](#)
 - Intrinsic vs. extrinsic evaluation
 - Evaluation forums/shared tasks
- Key takeaways
- Suggested readings

AUTOMATIC vs. HUMAN EVALUATION

- The best of two worlds:
 - COMET (Rei, 2019): Automatic metric used in Machine Translation that takes advantage of human evaluation (COMET is trained on human annotations – COMET is a trained metric – deep learning)
 - We will talk about this in one of the next classes

AUTOMATIC vs. HUMAN EVALUATION

- What is the best metric?
 - We also need to evaluate evaluation metrics
 - To do so, we check **how they relate** with human evaluation

AUTOMATIC vs. HUMAN EVALUATION

- Pearson, Spearman, and Kendall (remember?) are three statistical methods used to measure the strength and direction of association between two variables
 - They are used to find the correlation between human scores and metric scores. The higher the correlation with human scores, the better the metric is

AUTOMATIC vs. HUMAN EVALUATION

1. Pearson correlation coefficient (r):

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2} \sqrt{\sum(y_i - \bar{y})^2}}$$

where x_i and y_i are the values of the two variables, \bar{x} and \bar{y} are the means of those variables, respectively.

2. Spearman's rank correlation coefficient (ρ):

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

where d_i is the difference between the ranks of corresponding variables and n is the number of observations.

3. Kendall's tau (τ):

$$\tau = \frac{2}{n(n-1)} \sum_{i < j} \text{sign}(x_i - x_j) \text{sign}(y_i - y_j)$$

where n is the number of observations, x_i and x_j are the ranks of the x variables, y_i and y_j are the ranks of the y variables, and sign is the sign function.

An example of calculating Spearman's correlation

To calculate a Spearman rank-order correlation on data without any ties we will use the following data:

	Marks									
English	56	75	45	71	62	64	58	80	76	61
Maths	66	70	40	60	65	56	59	77	67	63

We then complete the following table:

English (mark)	Maths (mark)	Rank (English)	Rank (maths)	d	d^2
56	66	9	4	5	25
75	70	3	2	1	1
45	40	10	10	0	0
71	60	4	7	3	9
62	65	6	5	1	1
64	56	5	9	4	16
58	59	8	8	0	0
80	77	1	1	0	0
76	67	2	3	1	1
61	63	7	6	1	1

EVALUATION OF EVALUATION METRICS

- We can also evaluate two metrics considering how they rank two systems for each sample:
 - considering the scores given by two metrics to two systems, check if these metrics share the same sign that humans do

$$\frac{|\text{sign}(\text{metric}\Delta) = \text{sign}(\text{human}\Delta)|}{\text{all system pairs}}$$

Sample	Human		Sign	M1		Sign	M2		Sign
	Sys1	Sys2		Sys1	Sys2		Sys1	Sys2	
	7	6.5	+	5	4	+	7	7.3	-

M1 will gain 1 point because it ranked the same way as Humans this sample; M2 will not

ACTIVE LEARNING MOMENT



EXERCISE

- Consider that system X returns 10 translations that are scored by humans as SysXHuman and by metric Y as SysXMetricY. The values are:
 - Sys1Human = [6, 8, 9, 7, 9, 9, 7, 7, 6, 6] = S1 S1 S1 = S2 S2 S2 S2 =
 - Sys2Human = [6, 7, 5, 5, 9, 10, 8, 10, 7, 6]
 - Sys1Metric1 = [5, 7, 8, 6, 8, 9, 7, 7, 6, 6]
 - Sys2Metric1 = [6, 8, 5, 7, 7, 6, 7, 7, 7, 9] S2 S2 S1 S2 S1 S1 == S2 S2
 - Sys1Metric2 = [6, 8, 9, 7, 9, 10, 8, 8, 7, 8] S2 S2 S1 S1 = S1 S2 S2 S1 S2
 - Sys2Metric2 = [7, 9, 4, 6, 9, 9, 9, 10, 4, 9]
- Which metric better correlates with the human evaluation according with this metric?

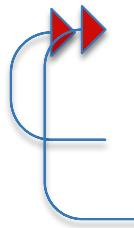
$$\frac{|\text{sign}(\text{metric}\Delta) = \text{sign}(\text{human}\Delta)|}{\text{all system pairs}}$$

Metric1 Accuracy: 0,2

Metric2 Accuracy: 0,5

EXERCISE

- In detail:
 - Comparing M1 and M2 with H



	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
H	=	S1	S1	S1	=	S2	S2	S2	S2	=
M1	S2	S2	S1	S2	S1	S1	=	=	S2	S2
M2	S2	S2	S1	S1	=	S1	S2	S2	S1	S2

M1 Accuracy: 2 in 10 = 0,2
M2 Accuracy: 5 in 10 = 0,5

OVERVIEW

- Learning Objectives
- Topics
 - Prelude: comparing strings
 - Automatic evaluation
 - Baseline
 - Some evaluation metrics
 - Human evaluation
 - Automatic vs. Human Evaluation
 - [Intrinsic vs. extrinsic evaluation](#)
 - Evaluation forums/shared tasks
- Key takeaways
- Suggested readings

EXTRINSIC VS. INTRINSIC EVALUATION

- Intrinsic: evaluate your system alone
- Extrinsic: evaluate your system as a component of a more complex system
- Example:
 - Question Classification (QC) vs. Question/Answer (QA)
 - A QC system can be evaluated “per se” and have a score X (intrinsic evaluation)
 - The same system might replace an existing QC system in a QA system, an improve (or not) the QA system (extrinsic evaluation of the QC system)



Kidney Stones



OVERVIEW

- Learning Objectives
- Topics
 - Prelude: comparing strings
 - Automatic evaluation
 - Baseline
 - Some evaluation metrics
 - Human evaluation
 - Automatic vs. Human Evaluation
 - Intrinsic vs. extrinsic evaluation
 - [Evaluation forums/shared tasks](#)
- Key takeaways
- Suggested readings

EVALUATION FORA (PL. OF FORUM)

- CLEF (QA, ...)
- IWSLT (Translation)
- SEM-EVAL (Semantics)
- SENSEVAL (Semantics)
- ...
- There are even forums for shared tasks to evaluate evaluation metrics (translation)



EVALUATION FORA (PL. OF FORUM)

- As we have seen:
 - Recent evaluation campaigns:
 - [...] new benchmark [...] requiring a single system to perform ten disparate natural language tasks [...]



EVALUATION FORA (PL. OF FORUM)

- As we have seen:
 - Recent evaluation campaigns:
 - \$500,000 prize will be awarded to the team that creates the best socialbot. The second- and third-place [...] \$100,000 and \$50,000, respectively.



KEY TAKEAWAYS

KEY TAKEAWAYS

- There are many different metrics to apply to different NLP tasks. Some tasks are properly evaluated; some do not
 - Both human and automatic evaluations have they pros and cons
 - Concepts:
 - Baseline, similarity measures, distance measures, evaluation measures, evaluation fora, automatic evaluation, human evaluation, intrinsic/extrinsic evaluations, evaluation fora, etc.

SUGGESTED READINGS

READINGS

- Sebenta:
 - Methodology, corpora and evaluation
 - notice that it does not covers all these slides
 - <https://aclanthology.org/2021.wmt-1.57.pdf>

<https://www.bbc.co.uk/programmes/b00n7sf5>



And now, for something...

N-GRAMS

Luísa Coheur

Overview

- Learning objectives
- Topics
 - Concepts
 - Language Models
 - N-grams
 - Word prediction
 - Sentence probability
 - Evaluation of N-grams
 - Challenges
 - Smoothing
- Key takeaways
- Suggested readings

LEARNING OBJECTIVES

LEARNING OBJECTIVES

- After this class, students should be able to:
 - Explain what a N-gram is
 - Understand how to model language, with N-grams
 - Apply N-grams to
 - Predict the next word of a given sentence, and
 - Calculate the probability of a sentence
 - Explain the concept of smoothing
 - Apply Laplace-smoothing

TOPICS

Overview

- Learning objectives
- Topics
 - Concepts
 - Language Models
 - N-grams
 - Word prediction
 - Sentence probability
 - Evaluation of N-grams
 - Challenges
 - Smoothing
- Key takeaways
- Suggested readings

LANGUAGE MODELS

- Models at the basis of Natural Language Generation
- **Language models** (notice that I am not using the word **large**) learn the probability distribution of words, that is, how words can be organized to create meaningful and grammatically correct sentences
- With **Language Models** we can:
 - Predict the next word within a text; and
 - Find how **likely (probable)** is a sequence of words

N-GRAMS

- Example 1: complete the following sentences:
 - And now for something...
 - Once upon a...
 - Spoiler...
 - Stranger...

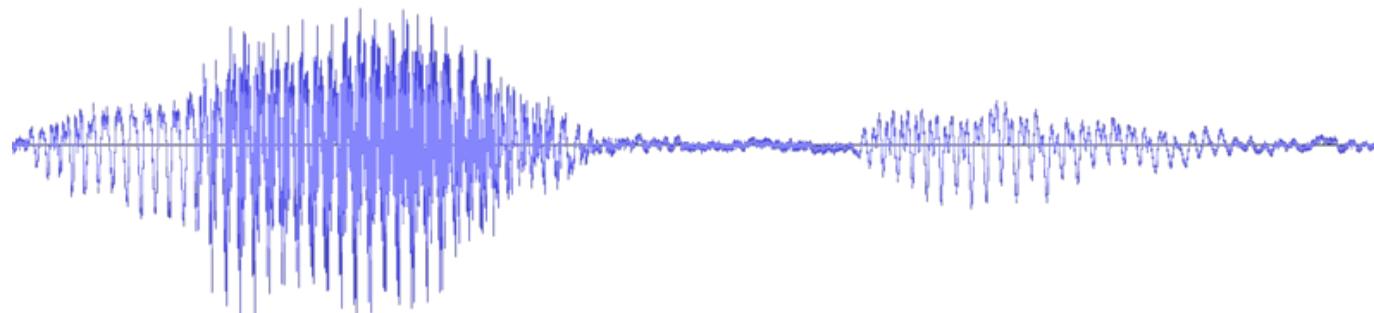
N-GRAMS

- Example 1: complete the following sentences:
 - And now for something completely different
 - Once upon a time
 - Spoiler alert
 - Stranger things

With N-grams we can make word prediction!!!!

N-GRAMS

- Example 2: consider the possible outputs of an Automatic Speech Recognizer (ASR):



- olá edgar
- ou lá apagar
- ó lá edgar
- ...

<https://commons.wikimedia.org/wiki/File:Signal-speech-martin-de.png>

Which sentence is the most likely?

N-GRAMS

- Example 3: consider the possible outputs of a Machine Translation System:

- Input: It is raining cats and dogs
- Possible translations:
 - Chovem cães e gatos
 - Chove a potes
 - Chovem potes
 - ...

Which sentence is the most likely?

Ok, but what is a N-gram?



N-GRAMS

- N-gram = sequence of N tokens
 - N = 1 => unigrams
 - N = 2 => bigrams
 - N = 3 => trigrams
 - ...
- A token can be
 - a word (ola, Maria, hello, ...)
 - a character (o, l, M, a, r, ...)
 - a set of sequences of characters (ol, la, Mar, ari, ...)

N-GRAMS

- Let us see now how to apply them to
 - Make word prediction
 - Calculate a sentence probability

Overview

- Learning objectives
- Topics
 - Concepts
 - Language Models
 - N-grams
 - Word prediction
 - Sentence probability
 - Evaluation of N-grams
 - Challenges
 - Smoothing
- Key takeaways
- Suggested readings

WORD PREDICTION

- Input: H (or history $H = W_1 \dots W_{N-1}$)
 - Task: find what is the probability of W ($= W_N$) being the next word, that is, we want to find:
 - $P(W | H)$
 - Notation:
 - $W_{1:N-1} = W_1 \dots W_{N-1}$
- Example:
 $H = \text{Once upon a}$
 $W = \text{time}$
 $\text{?? } P(\text{time} | \text{Once upon a})??$

WORD PREDICTION

- Hypothesis 1:

$$\begin{aligned} P(W \mid H) &= \text{count}(HW) / \sum_k \text{count}(HK) \\ &= \text{count}(HW) / \text{count}(H) \end{aligned}$$

- Example:

- H = Once upon a
- W = time
- $P(\text{time} \mid \text{Once upon a}) =$

$$= \text{count}(\text{once upon a time}) / \text{count}(\text{once upon a})$$

- Problem:

- Some sequences were never seen, thus, you might not have all these values

WORD PREDICTION

- Markov Assumption:
 - It is possible to calculate the probability of a future event without having to look to the entire history
- Let's do some approximations!!

WORD PREDICTION

- Hypothesis 2 (based on Markov assumption)
 - To calculate $P(W \mid H) = P(W_N \mid W_1 \dots W_{N-1})$:
 - $P(W_N \mid W_1 \dots W_{N-1}) \cong P(W_N \mid W_{N-1})$ (use bigrams)
 - $P(W_N \mid W_1 \dots W_{N-1}) \cong P(W_N \mid W_{N-2} W_{N-1})$ (use trigrams)

ACTIVE LEARNING MOMENT



EXERCISE

- Corpus (<s> for beginning of the sentence and </s> for the end):
 - <s>Eu adoro a Maria</s> (I adore Maria)
 - <s>A Maria eu adoro</s> (Maria I adore)
 - <s>Adoro bolachas Maria</s> ((I) adore cookies (named) Maria)

If I say “eu adoro” (I adore), what is the most probable next word: eu, a, Maria, adoro, bolachas or </s>?

- Use:
 - $P(W_N \mid W_1 \dots W_{N-1}) \cong P(W_N \mid W_{N-1})$ (use bigrams)
 - $P(W_N \mid W_1 \dots W_{N-1}) \cong P(W_N \mid W_{N-2} W_{N-1})$ (use trigrams)

EXERCISE: BIGRAMS

- First, some pre-processing:
 - <s>eu adoro a maria</s>
 - <s>a maria eu adoro</s>
 - <s>adoro bolachas maria</s>

What you need to know:

$$P(W_N | W_1 \dots W_{N-1}) \approx P(W_N | W_{N-1})$$

$$P(W | H) = \frac{\text{count}(HW)}{\text{count}(H)}$$

EXERCISE: BIGRAMS

- First, some pre-processing:
 - <s>eu adoro a maria</s>
 - <s>a maria eu adoro</s>
 - <s>adoro bolachas maria</s>
- Using bigrams: $P(W_N | W_1 \dots W_{N-1}) \cong P(W_N | W_{N-1})$
 - $P(\text{eu} | \text{adoro}) = \text{count}(\text{adoro eu})/\text{count}(\text{adoro}) = 0$
 - $P(\text{a} | \text{adoro}) = \text{count}(\text{adoro a})/\text{count}(\text{adoro}) = 1/3$
 - $P(\text{Maria} | \text{adoro}) = \text{count}(\text{adoro Maria})/\text{count}(\text{adoro}) = 0$
 - $P(\text{adoro} | \text{adoro}) = \text{count}(\text{adoro adoro})/\text{count}(\text{adoro}) = 0$
 - $P(\text{bolachas} | \text{adoro}) = \text{count}(\text{adoro bolachas})/\text{count}(\text{adoro}) = 1/3$
 - $P(</s> | \text{adoro}) = \text{count}(\text{adoro } </s>)/\text{count}(\text{adoro}) = 1/3$

What you need to know:

$$P(W_N | W_1 \dots W_{N-1}) \cong P(W_N | W_{N-1})$$

$$P(W | H) = \frac{\text{count}(HW)}{\text{count}(H)}$$

EXERCISE: TRIGRAMS

- First, some pre-processing:
 - <s>eu adoro a maria</s>
 - <s>a maria eu adoro</s>
 - <s>adoro bolachas maria</s>

What you need to know:

$$P(W_N | W_1 \dots W_{N-1}) \approx P(W_N | W_{N-2} W_{N-1})$$

$$P(W | H) = \frac{\text{count}(HW)}{\text{count}(H)}$$

EXERCISE: TRIGRAMS

What you need to know:

- First, some pre-processing:
 - <s>eu adoro a maria</s>
 - <s>a maria eu adoro</s>
 - <s>adoro bolachas maria</s>
- Using trigrams: $P(W_N | W_1 \dots W_{N-1}) \cong P(W_N | W_{N-2} W_{N-1})$
 - $P(\text{eu} | \text{eu adoro}) = \text{count}(\text{eu adoro eu})/\text{count}(\text{eu adoro}) = 0$
 - $P(\text{a} | \text{eu adoro}) = \text{count}(\text{eu adoro a})/\text{count}(\text{eu adoro}) = 1/2$
 - $P(\text{Maria} | \text{eu adoro}) = \text{count}(\text{eu adoro Maria})/\text{count}(\text{eu adoro}) = 0$
 - $P(\text{adoro} | \text{eu adoro}) = \text{count}(\text{eu adoro adoro})/\text{count}(\text{eu adoro}) = 0$
 - $P(\text{bolachas} | \text{eu adoro}) = \text{count}(\text{eu adoro bolachas})/\text{count}(\text{eu adoro}) = 0$
 - $P(</s> | \text{eu adoro}) = \text{count}(\text{eu adoro } </s>)/\text{count}(\text{eu adoro}) = 1/2$

Generated by DALL-E



Overview

- Learning objectives
- Topics
 - Concepts
 - Language Models
 - N-grams
 - Word prediction
 - Sentence probability
 - Evaluation of N-grams
 - Challenges
 - Smoothing
- Key takeaways
- Suggested readings

SENTENCE PROBABILITY

- Consider now that you want to know how probable is a sentence $W_1 \dots W_{N-1} W_N$

SENTENCE PROBABILITY

- We can use the chain rule (of probability):

$$\begin{aligned} P(w_1^N) &= P(w_1 | \langle s \rangle) * P(w_2 | \langle s \rangle w_1) * \dots * P(w_N | w_1^{N-1}) \\ &= \prod_{k=1}^N P(w_k | w_1^{k-1}) \end{aligned}$$

- We will have the same problem as before => some sequences were never seen. So, once again let us use the Markov assumption:

$$P(w_1^N) \cong \prod_{k=1}^N P(w_k | w_{k-1}) \text{ (use bigrams)}$$

$$P(w_1^N) \cong \prod_{k=1}^N P(w_k | w_{k-2} w_{k-1}) \text{ (use trigrams)}$$

ACTIVE LEARNING MOMENT



	I	Want	To	Eat	Chinese	Food	lunch
I	8	1087	0	13	0	0	0
Want	3	0	786	0	6	8	6
To	3	0	10	860	3	0	12
Eat	0	0	2	0	19	2	52
Chinese	2	0	0	0	0	120	1
Food	19	0	17	0	0	0	0
Lunch	4	0	0	0	0	1	0

Exercise:
What is the probability of the sentence “I eat Chinese food”

What you need to know:

$$P(w_1^N) \cong \prod_{k=1}^N P(w_k | w_{k-1})$$

$$P(W | H) = \text{count}(HW) / \text{count}(H)$$

I	Want	To	Eat	Chinese	Food	Lunch
3437	1215	3256	938	213	1506	459

	I	Want	To	Eat	Chinese	Food	lunch
I	8	1087	0	13	0	0	0
Want	3	0	786	0	6	8	6
To	3	0	10	860	3	0	12
Eat	0	0	2	0	19	2	52
Chinese	2	0	0	0	0	120	1
Food	19	0	17	0	0	0	0
Lunch	4	0	0	0	0	1	0

Exercise:
What is the probability of the sentence “I eat Chinese food”

What you need to know:

$$P(w_1^N) \cong \prod_{k=1}^N P(w_k | w_{k-1})$$

$$P(W | H) = \text{count}(HW) / \text{count}(H)$$

I	Want	To	Eat	Chinese	Food	Lunch
3437	1215	3256	938	213	1506	459

$P(\text{I eat Chinese food}) = P(\text{I} | <\text{s}>) * P(\text{eat} | \text{I}) * P(\text{Chinese} | \text{eat}) * P(\text{food} | \text{Chinese}) * P(</\text{s}> | \text{food})$ Assumindo que não se sabe $P(\text{I} | <\text{s}>)$ e $* P(</\text{s}> | \text{food})$, então $= C(\text{I eat})/C(\text{I}) * C(\text{eat Chinese})/C(\text{eat}) * C(\text{Chinese food})/\text{count}(\text{Chinese}) = 13/3437 * 19/938 * 120/213 = ...$

Overview

- Learning objectives
- Topics
 - Concepts
 - Language Models
 - N-grams
 - Word prediction
 - Sentence probability
 - **Evaluation of N-grams**
 - Challenges
 - Smoothing
- Key takeaways
- Suggested readings

EVALUATION OF N-GRAMS

- Perplexity:
 - Still used
 - “Train” set T:
 - Calculate:
 - Model_1 = unigrams in T
 - Model_2 = bigrams in T
 - ...

EVALUATION OF N-GRAMS

- Perplexity:
 - Test set: $W=w_1 w_2 \dots w_N$,
 - Calculate perplexity $PP(W)$ (for instance – different formulas):

$$PP(W) = P(w_1 w_2 \dots w_n)^{-\frac{1}{N}} = \sqrt[N]{\frac{1}{P(w_1 w_2 \dots w_N)}}$$

- There will be a different $PP(W)$ for **each model**:

$$PP(W) = \sqrt[N]{\prod_{i=1}^N \frac{1}{P(w_i | w_1 \dots w_{i-1})}}$$

- Lower value of $PP(W) \Rightarrow$ better model (less "perplex")

Overview

- Learning objectives
- Topics
 - Concepts
 - Language Models
 - N-grams
 - Word prediction
 - Sentence probability
 - Evaluation of N-grams
 - Challenges
 - Smoothing
- Key takeaways
- Suggested readings

CHALLENGES

- N-gram models are biased to the training corpus
- N-grams are not appropriate to deal with long distance dependencies
 - Gollum loves in a very sick way his precious
- Data sparseness
 - Bigger N (N-grams) => sparse data
- How to deal with 0 counts?
 - Smoothing is the answer

Overview

- Learning objectives
- Topics
 - Concepts
 - Language Models
 - N-grams
 - Word prediction
 - Sentence probability
 - Evaluation of N-grams
 - Challenges
 - Smoothing
- Key takeaways
- Suggested readings

SMOOTHING

- Techniques that allow to deal with the fact that some sequences were never seen or have not been seen many times
 - These techniques will change estimations/probability mass (and we need to guarantee that the counts still make sense => Robin Hood)



SMOOTHING

- Laplace or Add-one smoothing:
 - Add 1 to all the counts (and recalculate counts)

SMOOTHING

- Laplace or Add-one smoothing:
 - Example with bigrams:
 - Previously (Maximum Likelihood Estimation – MLE):
 - $P_{MLE}(W_N | W_{N-1}) = \text{count}(W_{N-1} W_N) / \text{count}(W_{N-1})$
 - Now:
 - $P_{Laplace}(W_N | W_{N-1}) = (\text{count}(W_{N-1} W_N) + 1) / (\text{count}(W_{N-1}) + |\mathcal{V}|)$
 - ($|\mathcal{V}|$ is the number of words in the vocabulary \mathcal{V})

SMOOTHING

- Laplace or Add-one smoothing:

- Example:

- $|V| = 100.000$ words

- $\text{count}(w_2) = 10, \text{count}(w_2 w_3) = 9,$

- Previously:

- $P_{MLE}(W_3 | W_2) = \text{count}(W_2 W_3)/\text{count}(W_2) = 9/10 = 0.9$

- Now:

- $P_{Laplace}(W_3 | W_2) = (\text{count}(W_2 W_3)+1)/(\text{count}(W_2) + |V|) = 10/100.010$

Problem:

If $\text{count}(w_1) = 10$, and $\text{count}(w_1 w_3) = 0$,

Then:

$$P_{MLE}(W_3 | W_1) = 0, P_{Laplace}(W_3 | W_1) = 1/100.010$$

Too close

SMOOTHING

- There are many more smoothing techniques
 - Good-Turing Discounting
 - In order to estimate the probabilities of things that occur c times, it uses the counts of things that occurred $(c+1)$ times (and then you will have to adjust everything again).
 - ...

KEY TAKEAWAYS

KEY TAKEAWAYS

- Understand concepts such as of N-grams, Markov assumptions and smoothing and Language Model
- Be able to apply N-grams to estimate the probability of a sentence or of a word, given a previous sequence of words

SUGGESTED READINGS

READINGS

- Sebenta: chapter about N-grams
- Jurafsky: 3.1, 3.3 and 3.6.1

Image from ChatGPT



PRE-PROCESSING

Luísa Coheur

Overview

- Learning objectives
- Topics
 - Tokenization
 - Concept
 - Character-level
 - Word-level
 - Subword-level
 - Sentence/word manipulation
 - Tips
- Key takeaways
- Suggested readings

LEARNING OBJECTIVES

LEARNING OBJECTIVES

- After this class, students should be able to:
 - Identify some pre-processing techniques
 - Be aware that they don't work in certain scenarios nowadays

TOPICS

Overview

- Learning objectives
- Topics
 - Tokenization
 - Concept
 - Character-level
 - Word-level
 - Subword-level
 - Sentence/word manipulation
 - Tips
- Key takeaways
- Suggested readings

TOKENIZATION

- Tokenization is the process of breaking down a stream of text into smaller, manageable units called **tokens**
- The goal is to create tokens that retain meaningful linguistic information while making the text more accessible for computational models

TOKENIZATION

- Consider the word “cats”. Which is the best input to a machine?
 - c + a + t + s
 - cats
 - cat + s
 - ...

Overview

- Learning objectives
- Topics
 - Tokenization
 - Concept
 - Character-level
 - Word-level
 - Subword-level
 - Sentence/word manipulation
 - Tips
- Key takeaways
- Suggested readings

TOKENIZATION: CHARACTER-LEVEL

- Character-level tokenization tokenizes text by splitting it into individual characters
- Example:
 - Input: hello
 - Output: ['h', 'e', 'l', 'l', 'o']
- + Can manage out-of-vocabulary (OOV) elements
- - Each token (character) carries very little context

Overview

- Learning objectives
- Topics
 - Tokenization
 - Concept
 - Character-level
 - Word-level
 - Subword-level
 - Sentence/word manipulation
 - Tips
- Key takeaways
- Suggested readings

TOKENIZATION: WORD-LEVEL

- Word-level tokenization splits text into individual words based, for instance, on **spaces and punctuation**
- Example:
 - Mr. Smith finished his Ph.D on the 28th April.
 - ['Mr.', 'Smith', 'finished', 'his', 'Ph.D', 'on', 'the', '28th', 'April', '.']
- + It is intuitive, as it as words are natural linguistic units
- - Requires language-specific rules (ex: Ph.D)
- - Struggles with unknown words, typos, or words not present in the training vocabulary

EXERCISE

- Can you think of tokens you wouldn't want to split based on punctuation?
 - Examples:
 - Sr.
 - 55.5 or 55,5
 - www.google.com
 - FT-34-56
 - ...
 - Rock 'n' roll
 - Toys'r us
 - U.S.A

ABOUT WORD-LEVEL TOKENIZATION

- It is possible that we also want to find sequences of words (**compounds**), that is, sequences of words that have some unified linguistic meaning
- Example:
 - Ice cream

ABOUT WORD-LEVEL TOKENIZATION

- Other scenarios:
 - Chinese:
 - don't have a white space between words
 - EN:
 - Nowherefast???? (best music ever! ☺)
 - Agglutinative languages:
 - Words are sequences of morphemes (such as Turkish)
(we will talk about this in a next class)

Overview

- Learning objectives
- Topics
 - Tokenization
 - Concept
 - Character-level
 - Word-level
 - Subword-level
 - Sentence/word manipulation
 - Tips
- Key takeaways
- Suggested readings

TOKENIZATION: SUBWORD-LEVEL

- Subword-level tokenization splits words into smaller meaningful units, such as prefixes, suffixes, or frequent subword patterns
 - Example:
 - Input: unhappiness
 - Output: ['un', 'happi', 'ness']
 - + Combines the benefits of character- and word-level tokenization by breaking down OOV words into known subwords
 - + It is widely used nowadays
 - - Requires more sophisticated algorithms to split text into subwords
 - - May break down words in a way that loses meaningful context

Overview

- Learning objectives
- Topics
 - Tokenization
 - Concept
 - Character-level
 - Word-level
 - Subword-level
 - Sentence/word manipulation
 - Tips
- Key takeaways
- Suggested readings

SENTENCE/WORD'S MANIPULATION

- We call **normalization** to the preprocessing step that transforms raw text into “standardized” format to reduce noise and linguistic variability (and, thus, data sparseness)
- There are many manipulations we can do to “normalize” text

SENTENCE/WORD'S MANIPULATION

- Remove stop words
 - Stop words (mainly functional words)
 - Examples:
 - a, de, para, ...
 - the, a, before, thus,
 - Problem: authorship identification!

There are lists of stopwords available.
Check them before you use them!!!!!!

SENTENCE/WORD'S MANIPULATION

- Remove punctuation
 - This can be ok, but it can also be a problem:
 - Os assassinos de D. Carlos, Afonso Costa e Buiça, foram...
 - Os assassinos de D. Carlos, Afonso Costa e Buiça foram...
 - The assassins of D. Carlos, Afonso Costa and Buiça, were...
 - The assassins of D. Carlos, Afonso Costa and Buiça were...

SENTENCE/WORD'S MANIPULATION

- Lowercasing
 - Avoid **data sparseness**
 - The dog is nice vs. I like **the** dog
 - Problem:
 - Us vs. us, Windows vs. windows, Figo vs. figo

SENTENCE/WORD'S MANIPULATION

- Normalization of dates, numbers, names, ...
 - Examples:
 - 8th-Feb
 - 8-Feb-2013
 - 02/08/13
 - February 8th, 2013
 - Feb 8th
 - ...

SENTENCE/WORD'S MANIPULATION

- Lemmatization: the process of reducing a word to its base or dictionary form (lemma) based on its meaning and context
- Example:
 - running → run
 - studies → study
 - went → go

SENTENCE/WORD'S MANIPULATION

- **Stemming:** the process of reducing a word to its root form by removing suffixes or prefixes, often without considering meaning
- Example:
 - running → run
 - studies → stud
 - went → went

Overview

- Learning objectives
- Topics
 - Tokenization
 - Concept
 - Character-level
 - Word-level
 - Subword-level
 - Sentence/word manipulation
 - [Tips](#)
- Key takeaways
- Suggested readings

TIP

- If you pre-process the training set, pre-process in the exact same manner the test set
 - Pre-processing is probably not a good idea if you are using LLMs
 - Pre-processing does not guarantee better results (so, sorry)



KEY TAKEAWAYS

KEY TAKEAWAYS

- Understand that corpora probably needs some pre-processing, although traditional operations, such as lemmatization or stop words removal make no sense nowadays in most cases

SUGGESTED READINGS

READINGS

- Sebenta: chapter 4 (4.1 and 4.2)
- Jurafsky: 2.5, 2.6, ...



Floppydisksareop • 4y ago

Regex is amazing. You can learn it for the first time every time you need it.



157



Award



Share

...

REGULAR EXPRESSIONS

Luís Coheur

Overview

- Learning objectives
- Topics
 - Regular Expressions
- Key takeaways
- Suggested readings

LEARNING OBJECTIVES

LEARNING OBJECTIVES

- After this class, students should be able to know how to apply Regular Expressions and understand how useful they can be

TOPICS

Overview

- Learning objectives
- Topics
 - Regular Expressions
- Key takeaways
- Suggested readings

APPROACH: RULE-BASED

- First NLP systems were rule-based (hand-crafted)
- There are many frameworks that allow us to craft rules, which can be very sophisticated
- We will focus on Regular Expressions

REGULAR EXPRESSIONS

- Regular expressions, sometimes known as regex or re, can be used for:
 - searching
 - matching, and
 - manipulating text
- First NLP systems, as ELIZA, were totally based on regular expressions
- Currently, regular expressions can be combined with more sophisticated techniques, but are still useful

REGULAR EXPRESSIONS

- RE are case sensitive
- RE always match the biggest string
- Characters inside braces [] specify a disjunction of characters to match:
 - Example:
 - [ola]pp means “o”, “l” or “a” and will match “opp”, “lpp” and “app” and not “olapp”;
 - Instead of [ABCDEFG], you can use [A–G] and instead of [0123...9] you can use [0-9]

REGULAR EXPRESSIONS

- means any character
 - Example:
 - $a.c$ represents abc , aac , acc , adc , $a1c$, a^*c , etc.
- Considering what appears before:
 - ? means zero or one
 - * means zero or more (the wild card)
 - + means one or more
 - Example:
 - $a?$ represents ϵ (empty string) and a
 - a^* represents ϵ , a , aa , aaa , $aaaa$, ...
 - aa^* represents a , aa , aaa , aaa , ...
 - $[ab]^*$ represents aaa , $abab$, bbb , ...

REGULAR EXPRESSIONS

- \wedge indicates the beginning of a line
- $\$$ indicates the end of a line.
- $|$ stands for the disjunction
- $()$ groups tokens
- $\{n\}$ = n occurrences of previous element
- $\{n,m\}$ = between n and m occurrences of previous element
- $\{n,\}$ = at least n occurrences of previous element

REGULAR EXPRESSIONS

- \d represents any digit
- \D means any character that is not a digit
- \w means any alpha-numeric character or a space
- \W means any character that is not alpha-numeric
- \s means any space (tab, blank, ...)
- \S means any character that is not a space

REGULAR EXPRESSIONS

- \backslash^* , $\backslash.$, $\backslash?$ represent, respectively, the special characters $*$, $.$ and $?$
- $\backslash n$, $\backslash t$ represent a newline and tab
- $[]$ and $^$ can also be used to declare characters that should not appear in the RE.
 - Example:
 - $[^a]$ means any character, except a

REGULAR EXPRESSIONS: APPLICATIONS

- Text Cleaning
 - Removing HTML Tags from Web Scrapped Data:
 - A regex pattern like /<[^>]+>/g can be used to find and remove all HTML tags (g means “everywhere”)
- Pattern Recognition/data extraction
 - Identifying Email Addresses:
 - The regex pattern \b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z | a-z]{2,}\b matches most email address formats
 - Extracting Dates:
 - A regex can help identify and extract the various formats of dates, useful for timeline analysis or event tracking

REGULAR EXPRESSIONS: APPLICATIONS

- Spam Detection:
 - Regular expressions can be instrumental in identifying common characteristics of spam messages, such as excessive use of capital letters, the presence of certain phrases (e.g., “BUY NOW”, “FREE”, “CLICK HERE”), or suspicious URLs

ACTIVE LEARNING MOMENT



EXERCISE

- Fill in the table, one letter in each cell
- Regular expressions indicate the letters that should appear in each column/line.
 - Example:
 - First line: you might have an H in the first cell and an E in the second, or an L in each cell or, at least, an O in one of the cells:

[^SPEAK]+	EP IP EF
HE LL O+	
[PLEASE]+	

EXERCISE

HE|LL|O+
[PLEASE]+

[^SPEAK]+

EPI|P|EF

H	E
L	P

KEY TAKEAWAYS

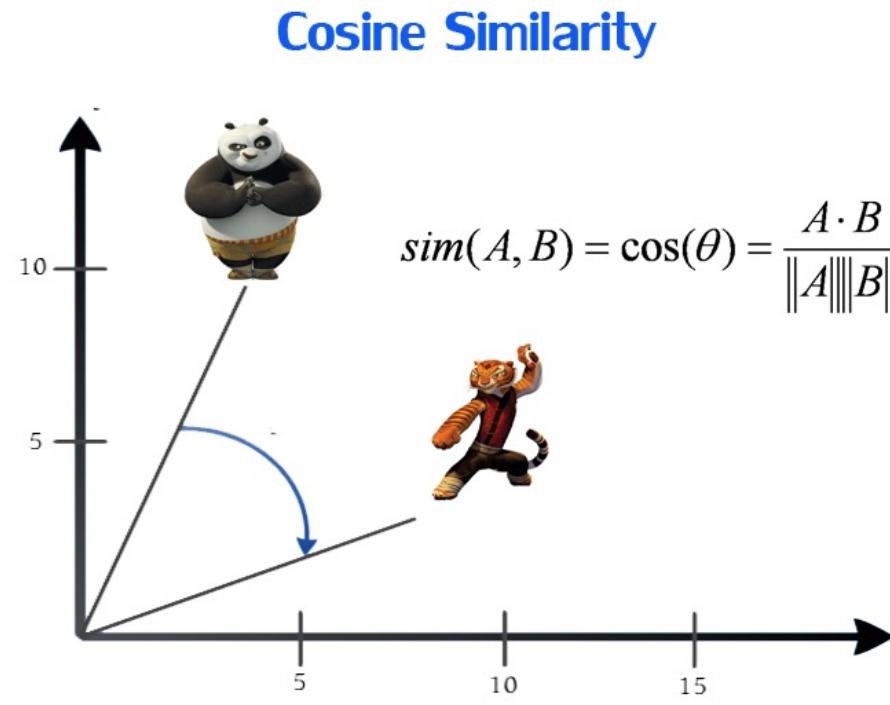
KEY TAKEWAYS

- The syntax of Regular Expressions and understand that they are still very useful in NLP, even in the Deep Learning era

SUGGESTED READINGS

READINGS

- Sebenta: chapter about Regular Expressions
- Jurafsky: 2.1



LANGUAGE AS VECTORS & FEATURE-BASED MACHINE LEARNING APPROACH

Luís Coheur

OVERVIEW

- Learning objectives
- Topics
 - Language Representation
 - Language as vectors
 - Documents as vectors
 - Words as vectors
 - Sentences as vectors
 - Feature Engineering Machine Learning Approach
 - Concept and examples of interesting features
 - Some widely used algorithms
 - End-to-end example
- Key takeaways
- Suggested readings

LEARNING OBJECTIVES

LEARNING OBJECTIVES

- After this class, students:
 - Should understand how language can be represented with vectors (sparse for now)
 - Describe different ways to create those vectors (including by using tf-idf)
 - Calculate the distance between given vectors by applying the cosine-similarity, including between sentences
 - Should be able to understand and define “feature engineering” machine learning and apply the Naïve Bayes algorithm
 - Reflect about how dangerous it can be to take decisions directly based on AI results

TOPICS

OVERVIEW

- Learning objectives
- Topics
 - Language Representation
 - Language as vectors
 - Documents as vectors
 - Words as vectors
 - Sentences as vectors
 - Feature Engineering Machine Learning Approach
 - Concept and examples of interesting features
 - Some widely used algorithms
 - End-to-end example
- Key takeaways
- Suggested readings

SNEAK PEEK

- Now we will see how to represent natural language as...



SNEAK PEEK

- We will see how to represent natural language as

VECTORS



LANGUAGE AS VECTORS (SPARSE)

- Distributional Semantics
 - Harris (1954)
 - “If A and B have almost identical environments [...] we say that they are synonyms”
 - J.R.Firth (1957)
 - “You shall know a word by the company it keeps”

LANGUAGE AS VECTORS (SPARSE)

- Example from Jurafsky:

tasty tnassiorc

greasy tnassiorc

tnassiorc with butter

tnassiorc for breakfast

→ FOOD

LANGUAGE AS VECTORS (SPARSE)

- The meaning of a word is given by the set of contexts in which it occurs
- Principles:
 - Meanings are locations in a **semantic space**
 - Words with similar distributional properties have similar meanings
- But, before moving into “Words as vectors”, let us check “Documents as vectors”

OVERVIEW

- Learning objectives
- Topics
 - Language Representation:
 - Language as vectors
 - Documents as vectors
 - Words as vectors
 - Sentences as vectors
 - Feature Engineering Machine Learning Approach
 - Concept and examples of interesting features
 - Some widely used algorithms
 - End-to-end example
- Key takeaways
- Suggested readings

DOCUMENTS AS VECTORS

- Documents as vectors main idea (from Information Retrieval (IR)):
 - A **vector space model** is a model for representing a text document as a vector

DOCUMENTS AS VECTORS

Documents

However, complexity...
We will see how small
Given a function based
Using entropy of traffic
We study the complexity
of influencing elections
through bribery: How
computationally complex
is it for an external actor
to determine whether by
a certain amount of
bribing voters a specified
candidate can be made
the election's winner? We
study this problem for
election systems as varied
as scoring ...



Vector-space representation

	D1	D2	D3	D4	D5
complexity	2				
algorithm	3			4	4
entropy	1			2	
traffic		2	3		
network		1	4		

number of times
"complexity" occurs in D1

Term-document matrix

Example from [SpeedLab](#)

DOCUMENTS AS VECTORS

MAIN IDEA

Documents



Vector-space
representation

However, complexity...
We will see how small...
Given a function based...
Using entropy of traffic...
We study the complexity...
of influencing elections...
through bribery: How...
computationally complex...
is it for an external actor...
to determine whether by...
a certain amount of...
bribing voters a specified...
candidate can be made...
the election's winner? We...
study this problem for...
election systems as varied...
as scoring ...

	D1	D2	D3	D4	D5
complexity	2		3	2	3
algorithm	3			4	4
entropy	1			2	
traffic		2	3		
network		1	4		

Term-document matrix

Vectors for D1 and D2

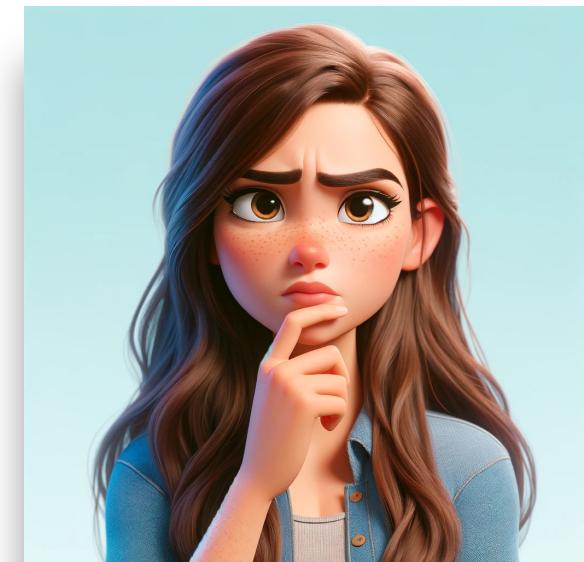
Example from [SpeedLab](#)

DOCUMENTS AS VECTORS

MAIN IDEA

- Vector space model
 - Being given a collection of documents, build a term-document matrix in which:
 - Each row represents a term in the vocabulary
 - Each column represents a document

But what do we put inside each cell of that matrix?



DOCUMENTS AS VECTORS

BUILDING THE MATRIX

- Binary
 - the elements of the vectors are either 1 or 0, where 1 indicates that the word has occurred in the document, and 0 that it has not
- Raw count
 - the elements are the raw frequency of occurrence of the word in the document

	Document 1	Document 2	Document 3
bank	0	0	4
bass	2	4	0
commercial	0	2	2
cream	2	0	0
guitar	1	0	0
fishermen	0	3	0
money	0	1	2

DOCUMENTS AS VECTORS

BUILDING THE MATRIX

- TF-IDF (the famous TF-IDF!!!!!) combines the term frequency (TF) with the inverse document frequency (IDF):
 - $\text{TF-IDF}(t, d, D) = \text{TF}(t,d) \times \text{IDF}(t, D)$
 - in which:
 - t term,
 - d document,
 - D collection of documents

DOCUMENTS AS VECTORS

BUILDING THE MATRIX

- TF (Term Frequency):
 - $Tf(t, d) = freq(t, d)$ = term t frequency in document d (it can be just the count or the relative frequency in the document; next we will use the second possibility)
- IDF (Inverse Document Frequency):
 - $IDF(t, D) = \log(|D| / |\{d \in D : t \in d\}|)$, in which:
 - $|D|$ is the number of documents
 - $|\{d \in D : t \in d\}|$ is the number of documents that contain the term t
 - \log is in base e
 - (notice that there are other possible formulas)

ACTIVE LEARNING MOMENT: THINK-PAIR-SHARE



EXERCISE

- Let:
 - $t = \text{gato}$ (cat)
 - d_{129} = document with 100 words, in which “gato” occurs 3 times
 - $|D|$ = collection of 10.000.000 documents; “gato” appears in 1000 documents
- Considering:
 - $\text{Tf}(t, d) = \text{freq}(t, d) = \text{term } t \text{ relative frequency in document } d$
 - $\text{IDF}(t, D) = \log (|D| / |\{d \in D : t \in d\}|)$
- Then:
 - $\text{TF-IDF}(\text{gato}, d_{129}, D) = \text{TF}(\text{gato}, d_{129}) \times \text{IDF}(\text{gato}, D) =$
 - $= 0,03 \times \log (|D| / 1000) = 0,03 \times 9,2$

DOCUMENTS AS VECTORS

BUILDING THE MATRIX

- Raw counts:

	Document 1	Document 2	Document 3
bank	0	0	4
bass	2	4	0
commercial	0	2	2
cream	2	0	0
guitar	1	0	0
fishermen	0	3	0
money	0	1	2

- Tf-idf

	Document 1	Document 2	Document 3
bank	0	0	0.76
bass	0.23	0.28	0
commercial	0	0.23	0.23
cream	0.62	0	0
guitar	0.48	0	0
fisherman	0	0.70	0
money	0	0.18	0.23

Note: just an example, doesn't mean that these counts are accurate

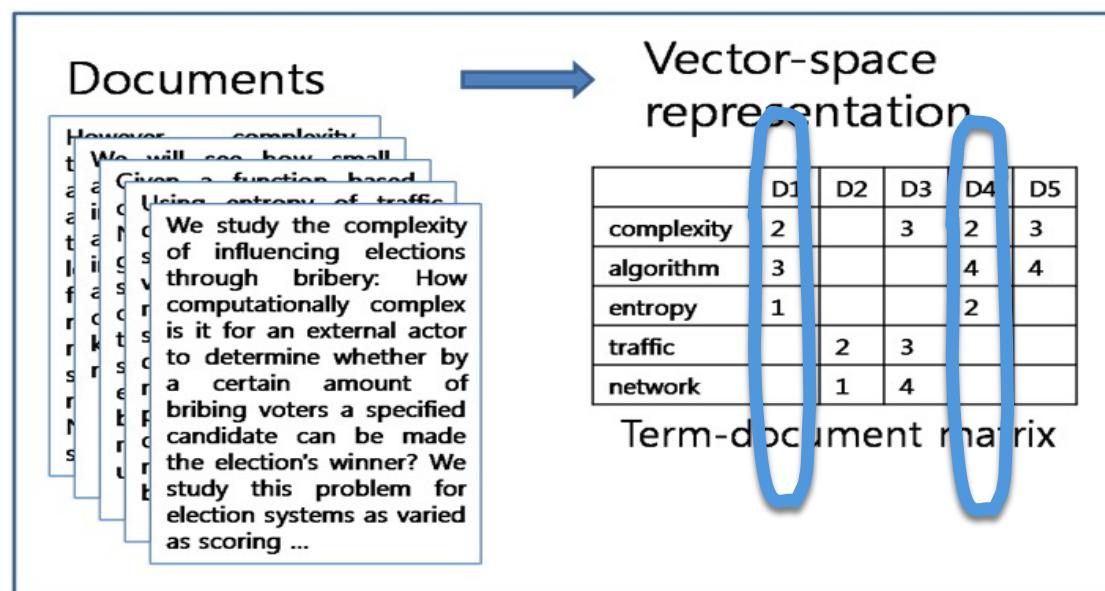
OVERVIEW

- Learning objectives
- Topics
 - Language Representation:
 - Language as vectors
 - Documents as vectors
 - **Words as vectors**
 - Sentences as vectors
 - Feature Engineering Machine Learning Approach
 - Concept and examples of interesting features
 - Some widely used algorithms
 - End-to-end example
- Key takeaways
- Suggested readings

WORDS AS VECTORS

- A word can have a dual role in word space:
 - Each word can be a dimension, an axis of the space (=> documents as vectors)
 - Each word can be a vector in the space in which documents are the axis (=> words as vectors)

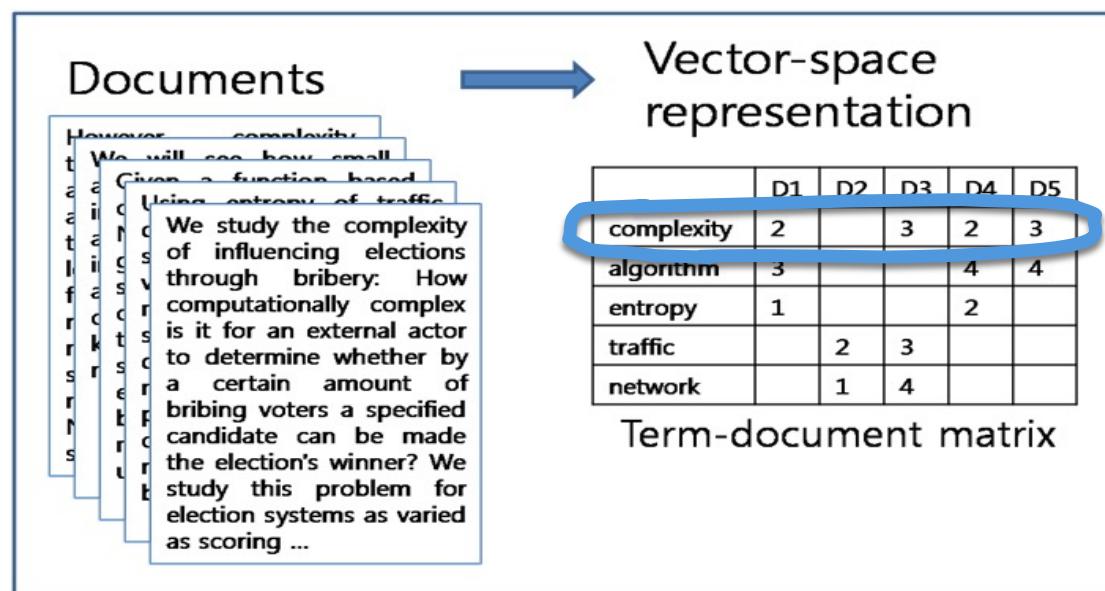
PREVIOUSLY:



WORDS AS VECTORS

- A word can have a dual role in word space:
 - Each word can be a dimension, an axis of the space (=> documents as vectors)
 - Each word can be a vector in the space in which documents are the axis (=> words as vectors)

PREVIOUSLY:



WORDS AS VECTORS

- But we can take this idea a little further:
 - Instead of a term-document matrix we can have term-term matrix (also called word-word matrix or term-context matrix) by considering the words' context



WORDS AS VECTORS BUILDING THE MATRIX

1. I enjoy flying.
2. I like NLP.
3. I like deep learning.

Window
(context) = 1

The resulting counts matrix will then be:

“Like” twice in the context of “I”

	<i>I</i>	<i>like</i>	<i>enjoy</i>	<i>deep</i>	<i>learning</i>	<i>NLP</i>	<i>flying</i>	.
<i>I</i>	0	2	1	0	0	0	0	0
<i>like</i>	2	0	0	1	0	1	0	0
<i>enjoy</i>	1	0	0	0	0	0	1	0
<i>deep</i>	0	1	0	0	1	0	0	0
<i>learning</i>	0	0	0	1	0	0	0	1
<i>NLP</i>	0	1	0	0	0	0	0	1
<i>flying</i>	0	0	1	0	0	0	0	1
.	0	0	0	0	1	1	1	0

WORDS AS VECTORS BUILDING THE MATRIX



: Center Word



: Context Word

c=1 The **cute** **cat** **jumps** over the lazy dog.

c=2 The **cute** **cat** **jumps** **over** the lazy dog.

<https://docs.chainer.org/en/stable/examples/word2vec.html>

WORDS AS VECTORS

BUILDING THE MATRIX

- $|V|$ = the size of the vocabulary
 - $|V|$ usually between 10.000 and 50.000
 - most frequent words are used; more than that is not helpful
 - size of the window between 1 and 8
 - => total context between 3 and 17

WORDS AS VECTORS

BUILDING THE MATRIX

- But... simple frequency is still not the best measure
 - words such as “the”, “it”, “they” are not discriminative
 - the best weighting or measure of association between two words should tell us more than a chance of co-occurrence
- Use:
 - pointwise mutual information (Tf-idf is usually not used for word/word similarity)
 - ...

WORDS AS VECTORS BUILDING THE MATRIX

- Up to now: very high-dimensional space
 - Words represented as sparse vectors
 - Long vectors: length = $|V|$, 10.000-50.000
 - Sparse vectors: most entries = 0
 - => need for dimensionality reduction
- Next week we will see that words can be represented as dense vectors
 - Short vectors: 50-1.000 dimensions
 - Dense vectors: most entries $\neq 0$

SIMILARITY/DISTANCE METRICS AGAIN

- How to measure similarity between two words or two documents when they are represented as vectors?
 - Remember that distances can be transformed into similarity measures (low distances => similar words/documents)

	Document 1	Document 2	Document 3
bank	0	0	0.76
bass	0.23	0.28	0
commercial	0	0.23	0.23
cream	0.62	0	0
guitar	0.48	0	0
fisherman	0	0.70	0
money	0	0.18	0.23

SIMILARITY/DISTANCE METRICS AGAIN

- If documents are vectors, distances can also be easily calculated:



SIMILARITY/DISTANCE METRICS AGAIN

- If documents are vectors, distances can also be easily calculated:
 - Use the angle!!!!!!
 - small angles mean similar documents
 - maximal similarity when the angle is zero



SIMILARITY/DISTANCE METRICS AGAIN

- Cosine similarity (cosine is 1 when the angle between two vectors is 0 and smaller for every other angle)

$$\text{similarity}(A, B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n A_i^2} \times \sqrt{\sum_{i=1}^n B_i^2}}$$

SIMILARITY/DISTANCE METRICS AGAIN

$$\begin{array}{l} \xrightarrow{\quad\quad\quad} \\ \text{DOC 1} = (\begin{matrix} \text{abandon} & \dots & \text{plane} & \dots & \text{kill} & \dots & \text{survive} & \dots & \text{people} \\ (0.11 & \dots & 1.00 & \dots & 0.03 & \dots & 0.23 & \dots & 0.65) \end{matrix}) \\ \vdots \\ \xrightarrow{\quad\quad\quad} \\ \text{DOC 2} = (\begin{matrix} 0.00 & \dots & 0.01 & \dots & 1.00 & \dots & 0.11 & \dots & 0.09 \end{matrix}) \end{array}$$

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}}$$

$$\text{sim}(\text{DOC1} \& \text{DOC2}) = \frac{0.11 * 0.00 + \dots + 0.65 * 0.09}{\sqrt{0.11^2 + \dots + 0.65^2} * \sqrt{0.00^2 + \dots + 0.09^2}} \in [0,1]$$

OVERVIEW

- Learning objectives
- Topics
 - Language Representation:
 - Language as vectors
 - Documents as vectors
 - Words as vectors
 - [Sentences as vectors](#)
 - Feature Engineering Machine Learning Approach
 - Concept and examples of interesting features
 - Some widely used algorithms
 - End-to-end example
- Key takeaways
- Suggested readings

CASE STUDY

- With these metrics applied to sentences you can build a retrieval-based chatbot!!!!
- Knowledge base
 - $Q_1:A_1$
 - $Q_2:A_2$
 - $Q_3:A_3$
 - ...
- User: Q
- How: find the distance between Q and Q_i . Return the A_j associated with the Q_j that has the highest similarity (or smaller distance)

CASE STUDY

- Example:
 - “TRAINING” CORPUS (knowledge base of FAQs):
 - Q: “Writer of LOTR” A: Tolkien
 - “TEST”: Who wrote LOTR?
- Simple solution with $Jaccard(s, t) = |s \cap t| / |s \cup t|$
 - and some pre-processing that might help... or not:
 - Direct: Jaccard(Writer of LOTR, Who wrote LOTR?)
 - Lowercasing: Jaccard(writer of lotr, who wrote lotr?)
 - Stop-words + Punctuation removal: Jaccard(writer lotr, who wrote lotr)
 - Lemmatization: Jaccard(write lotr, who write lotr)
- Now you can use the cos-similarity instead of Jaccard
 - Which vectors? Should we return Tolkien?

CASE STUDY

- This was in 2017
 - and due to a chatbot based in similarity measures!!!!





OVERVIEW

- Learning objectives
- Topics
 - Language Representation:
 - Language as vectors
 - Documents as vectors
 - Words as vectors
 - Sentences as vectors
 - Feature Engineering Machine Learning Approach
 - Concept and examples of interesting features
 - Some widely used algorithms
 - End-to-end example
- Key takeaways
- Suggested readings

FEATURE ENGINEERING

- Feature Engineering is a machine learning approach in which **domain knowledge** is used to extract features from raw data. These features are used to improve the performance of machine learning algorithms
 - Some features make sense, others do not!

ACTIVE LEARNING MOMENT: THINK-PAIR-SHARE



EXERCISE

- Consider the task of classifying a proper noun, in Portuguese, according with its gender
 - That is, being given a proper name, decide if it is feminine or masculine
 - Example:
 - Lúisa, Manuel, João, Inês, ...
 - Question: which are the features that will help you to decide (if a given noun is feminine or masculine)?

EXERCISE

- Two classes:
 - fem (feminine) and masc (masculine)
- Build a corpus:
 - The X most frequent masculine (masc) and feminine (fem) proper names
- Example:
 - my_data = [('Maria', 'fem'), ('Matilde ', 'fem'), ('Carolina', 'fem'), ('Beatriz', 'fem'), ('Alícia', 'fem'), ('Nicole', 'fem'), ('João ', 'masc'), ('Rodrigo ', 'masc'), ('Martim', 'masc'), ('Francisco', 'masc'), ...]

EXERCISE

- Some features make sense:
 - The last character might give you an important clue
 - Example:
 - Maria, Luísa, Joana, Ana vs. Miguel, João, Pedro
- Some features don't:
 - In this scenario, the size of the proper noun is irrelevant
 - Example:
 - Ana (size 3) vs. Rui (size 3)
 - Inês (size 4) vs. João (size 4)
 - Antonieta (size 9) vs. Godofredo (size 9)
- Anyway, you will fail! 😞
 - Examples:
 - Luca, Garcia, Quaresma ...

EXERCISE

	First char	Last char	Num a's	...	Num z's	Has a's?	...	Has z's?	Gender
Maria	m	a	2		0	1		0	fem
Pedro	p	o	0		0	0		0	mes
Inês	i	s	0		0	0		0	fem
Miguel	m	l	0		0	0		0	mas
Jorge	j	e	0		0	0		0	mas
Fernando	f	o	1		0	1		0	mas

ACTIVE LEARNING MOMENT: THINK-PAIR-SHARE



EXERCISE

- Detect interactional style (paper from JURAFSKY)
 - Given speech and text from a conversation can we tell if a speaker is
 - Awkward?
 - Flirtatious?
 - Friendly?
 - Dataset
 - 1000 4-minute “speed-dates”
 - Each subject rated their partner for these styles
 - Example:



EXERCISE

- Which features should we consider?

DETECT INTERACTIONAL STYLE

- Prosodic and Lexical features:
 - Pitch min, max, mean, range, ...
 - Duration of turn
 - Number of words
 - Use of past tense
 - Use of “you”
 - Use of “we”
 - ...

DETECT INTERACTIONAL STYLE

- Discourse Features
 - # of Backchannels
 - Examples:
 - Uh-huh. Yeah. Right. Oh, Okay
 - # of Appreciations
 - Examples:
 - Wow. That's true. Oh, great!
 - # of Questions
 - Amount of Laughter
 - Total number of turns
 - # of disfluencies
 - Amount of overlapped speech
 - ...

BY THE WAY...

- Disfluencies:
 - "false starts"
 - words and sentences that are cut off
 - phrases that are restarted or repeated
 - repeated syllables
 - "fillers":
 - non-lexical utterances (ex: huh, uh, erm, um, and hmm)
 - semiarticulate utterances (ex: well, so, I mean, and like)
 - "repaired" utterances: correcting mispronunciations (for instance)

Adapted from Wikipedia

DETECT INTERACTIONAL STYLE

- Results presented by Jurafsky:
 - Good predictors, across both genders:
 - Awkward speaker: slow, lower pitched, stilted talk
 - Flirtatious speaker: greater laughter, more questions, and referring to the past
 - Friendly speaker: greater laughter
 - Gender differences
 - Flirtation:
 - Women raise pitch,
 - Men drop pitch
 - ...

OVERVIEW

- Learning objectives
- Topics
 - Language Representation:
 - Language as vectors
 - Documents as vectors
 - Words as vectors
 - Sentences as vectors
 - Feature Engineering Machine Learning Approach
 - Concept and examples of interesting features
 - [Some widely used algorithms](#)
 - End-to-end example
- Key takeaways
- Suggested readings

SOME WELL KNOWN ALGORITHMS

- Classification tasks:
 - Decision trees
 - Support Vector Machines
 - Naïve Bayes
 - ...
- Sequence Prediction tasks:
 - CRFs
 - ...

NAÏVE BAYES (NB) CLASSIFIER

- Family of algorithms that, to predict the category of a given sample:
 - apply Bayes theorem, and
 - assume that every feature is independent of the others
 - Being
 - $C = \{c_1, \dots, c_n\}$ // categories
 - $x = (E_1, \dots, E_m)$. // evidences/features
- Then
 - $\operatorname{argmax}_{c_i \text{ in } C} P(c_i | x) \approx$
 - $\operatorname{argmax}_{c_i \text{ in } C} P(E_1 | c_i) * \dots * P(E_m | c_i) * P(c_i)$
- So, find
 - $P(c_i | x) = P(E_1 | c_i) * \dots * P(E_m | c_i) * P(c_i)$ and check the highest value (remember that this is not a probability)

NB DRAWBACKS

- Independence assumption
 - But many features are not independent
- “zero-frequencies” are a problem
 - Possible solution?
 - Smoothing techniques (as add-one)

OVERVIEW

- Learning objectives
- Topics
 - Language Representation:
 - Language as vectors
 - Documents as vectors
 - Words as vectors
 - Sentences as vectors
 - Feature Engineering Machine Learning Approach
 - Concept and examples of interesting features
 - Some widely used algorithms
 - [End-to-end example](#)
- Key takeaways
- Suggested readings

FORENSIC LINGUISTICS

- Application of linguistic knowledge, methods, and insights to the forensic context of law, language, crime investigation, trial, and judicial procedure
- Nice stories:
 - my/myself vs. me/meself
 - on the left vs. to the left vs. on the left side
 - I am going vs. I don't want. Also: "She tried to break it off. So, I broke her neck."

**Forensic
Linguistics
Is also an
NLP TASK!**



ACTIVE LEARNING MOMENT



EXERCISE

- A cruel criminal (from now on CC) is stealing TV's remote controls from innocents' homes, and he/she always leaves a letter in the crime scene
- There was another crime, and a suspect was detained. Inspector Morcela wants to know if the suspect is the real criminal or an imitator

**Inspector
Morcela,
the ONE**



EXERCISE

- Due to your expertise in NLP, Inspector Morcela calls you:
 - he wants you to analyse the letters and tell him which are the chances of being the same criminal or an imitator
- You decide to approach this challenge as a classification problem:
 - You add to the given letters (**positive examples**) other letters written by other criminals (**negative examples**).
 - You decide to consider the following features:
 - 1 – the letter starts with the word “I” or not;
 - 2 – the letter has spelling errors or not;
 - 3 – the letter has few/average/high number of adjectives;
 - 4 – the letter is extremely polite/normal/rude.

Based on this, you build the following table:

Starts with "I"	Spelling errors	Adjectives	Style	Written by CC?
NO	YES	FEW	EXTR. POLITE	YES
NO	NO	FEW	EXTR. POLITE	YES
NO	YES	AVERAGE	EXTR. POLITE	NO
NO	YES	HIGH	NORMAL	NO
YES	YES	HIGH	RUDE	NO
NO	NO	HIGH	NORMAL	YES
YES	NO	AVERAGE	RUDE	NO
NO	YES	FEW	NORMAL	YES
YES	YES	FEW	RUDE	NO
YES	YES	HIGH	NORMAL	NO
YES	NO	FEW	NORMAL	NO
NO	NO	AVERAGE	NORMAL	NO
YES	YES	AVERAGE	EXTR. POLITE	NO
YES	NO	HIGH	NORMAL	YES

Q1: By using Naïve Bayes (no smoothing), find the probability of the new letter being written by CC.

Now you look at the **new letter**:

- 1 – It starts with “I”;
- 2 – There are no spelling mistakes;
- 3 – It has few adjectives;
- 4 – The style is extremely polite.

$P(\text{'Written_by_CC'} = \text{YES} \mid \text{'starts_with_I'} = \text{YES}, \text{'Spelling errors'} = \text{NO}, \text{Adjectives} = \text{FEW}, \text{Style} = \text{'EXTR. POLITE'})$

Starts with "I"	Spelling errors	Adjectives	Style	Written by CC?
NO	YES	FEW	EXTR. POLITE	YES
NO	NO	FEW	EXTR. POLITE	YES
NO	YES	AVERAGE	EXTR. POLITE	NO
NO	YES	HIGH	NORMAL	NO
YES	YES	HIGH	RUDE	NO
NO	NO	HIGH	NORMAL	YES
YES	NO	AVERAGE	RUDE	NO
NO	YES	FEW	NORMAL	YES
YES	YES	FEW	RUDE	NO
YES	YES	HIGH	NORMAL	NO
YES	NO	FEW	NORMAL	NO
NO	NO	AVERAGE	NORMAL	NO
YES	YES	AVERAGE	EXTR. POLITE	NO
YES	NO	HIGH	NORMAL	YES

$P(\text{'Written_by_CC'} = \text{YES} \mid \text{'starts_with_I'} = \text{YES}, \text{'Spelling errors'} = \text{NO},$
 $\text{Adjectives} = \text{FEW}, \text{Style} = \text{'EXTR. POLITE'}) =$

$$P(\text{'starts_with_I'} = \text{YES} \mid \text{'Written_by_CC'} = \text{YES}) * P(\text{'Spelling errors'} = \text{NO} \mid \text{'Written_by_CC'} = \text{YES}) * P(\text{Adjectives} = \text{FEW} \mid \text{'Written_by_CC'} = \text{YES}) * \\ P(\text{Style} = \text{'EXTR. POLITE'} \mid \text{'Written_by_CC'} = \text{YES}) * P(\text{'Written_by_CC'} = \text{YES})$$

Starts with "I"	Spelling errors	Adjectives	Style	Written by CC?
NO	YES	FEW	EXTR. POLITE	YES
NO	NO	FEW	EXTR. POLITE	YES
NO	YES	AVERAGE	EXTR. POLITE	NO
NO	YES	HIGH	NORMAL	NO
YES	YES	HIGH	RUDE	NO
NO	NO	HIGH	NORMAL	YES
YES	NO	AVERAGE	RUDE	NO
NO	YES	FEW	NORMAL	YES
YES	YES	FEW	RUDE	NO
YES	YES	HIGH	NORMAL	NO
YES	NO	FEW	NORMAL	NO
NO	NO	AVERAGE	NORMAL	NO
YES	YES	AVERAGE	EXTR. POLITE	NO
YES	NO	HIGH	NORMAL	YES

$P('Written_by_CC' = \text{YES} | 'starts_with_I' = \text{YES}, 'Spelling errors' = \text{NO},$
 $\text{Adjectives} = \text{FEW}, \text{Style} = \text{'EXTR. POLITE'}) =$

$$P('starts_with_I' = \text{YES} | 'Written_by_CC' = \text{YES})^*$$

$$P('Spelling errors' = \text{NO} | 'Written_by_CC' = \text{YES}) *$$

$$P(\text{Adjectives} = \text{FEW} | 'Written_by_CC' = \text{YES})^*$$

$$P(\text{Style} = \text{'EXTR. POLITE} | 'Written_by_CC' = \text{YES}) *$$

$$P('Written_by_CC' = \text{YES}) = 1/5 * 3/5 * 3/5 * 2/5 * 5/14 = 0,01$$

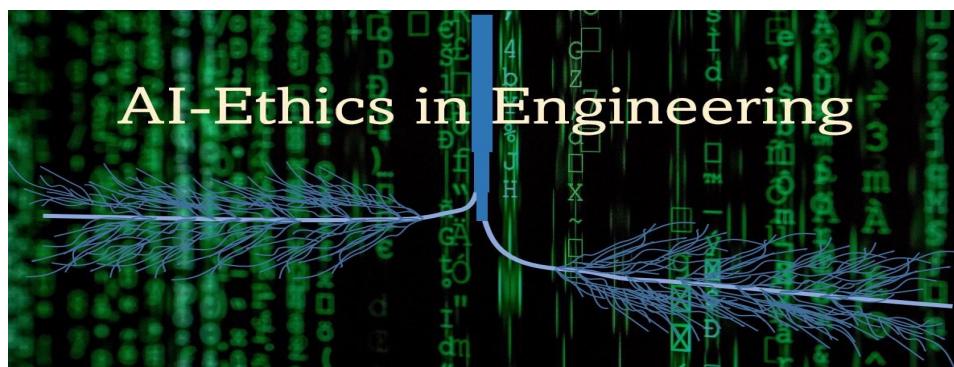
Do not forget that this is not a probability!

Starts with "I"	Spelling errors	Adjectives	Style	Written by CC?
NO	YES	FEW	EXTR. POLITE	YES
NO	NO	FEW	EXTR. POLITE	YES
NO	YES	AVERAGE	EXTR. POLITE	NO
NO	YES	HIGH	NORMAL	NO
YES	YES	HIGH	RUDE	NO
NO	NO	HIGH	NORMAL	YES
YES	NO	AVERAGE	RUDE	NO
NO	YES	FEW	NORMAL	YES
YES	YES	FEW	RUDE	NO
YES	YES	HIGH	NORMAL	NO
YES	NO	FEW	NORMAL	NO
NO	NO	AVERAGE	NORMAL	NO
YES	YES	AVERAGE	EXTR. POLITE	NO
YES	NO	HIGH	NORMAL	YES

Q2: What can you say to Inspector Morcela considering that your verdict can take an innocent to prison or lead to the release of a criminal? (this is serious)

THE VEREDICT

- After finding $P(\text{'Written by CC'} = \text{NO} \mid \text{bla-bla})$, you would have a clue about the letter being written or not by CC. **However**, even so, we have **very little data**, thus, we could never tell Morcela for sure that the suspect should be released or go to prison. **Besides**, we are using Naïve Bayes (naïve assumption). **Besides (again)**... an innocent could go to prison due to you. That is very disturbing.



KEY TAKEAWAYS

KEY TAKEAWAYS

- Words, sentences and documents can be represented as vectors (sparse for now)
- Being able to represent language as vectors allow us to apply mathematic stuff to language (for instance, similarity metrics and machine learning algorithms)
- In feature-based machine learning the specific characteristics of the data in hands is carefully analysed and tailored features are design to be given as algorithms input
- We should always be very carefully in taking blind decisions, based on a score produced by an algorithm

SUGGESTED READINGS

READINGS

- Jurafsky: 6.3, 6.4, 6.5



APPROACHES: DEEP LEARNING

Luísa Coheur

OVERVIEW

- Learning objectives
- Topics
 - Feedforward Neural Networks
 - Recurrent Neural Networks
 - Sequence to Sequence Models
 - Autoencoders
 - Generative Adversarial Networks
 - Attention
 - Transformers
- Key takeaways
- Suggested readings

LEARNING OBJECTIVES

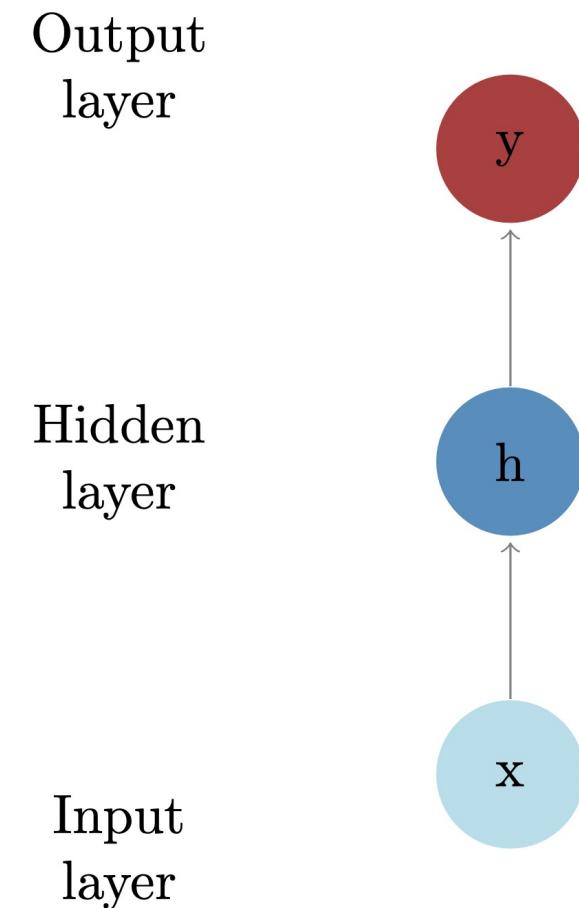
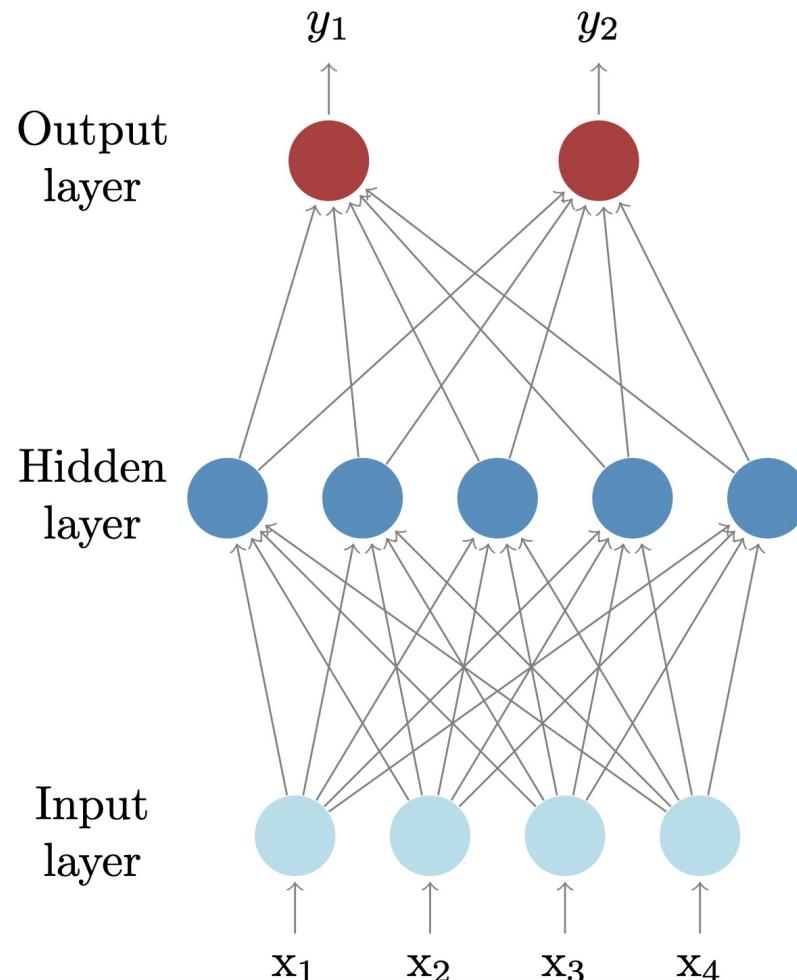
LEARNING OBJECTIVES

- After this class, students should be able to:
 - Learn to identify and describe different deep learning architectures and understand how these have progressed
 - Define several concepts related to Deep Learning

TOPICS

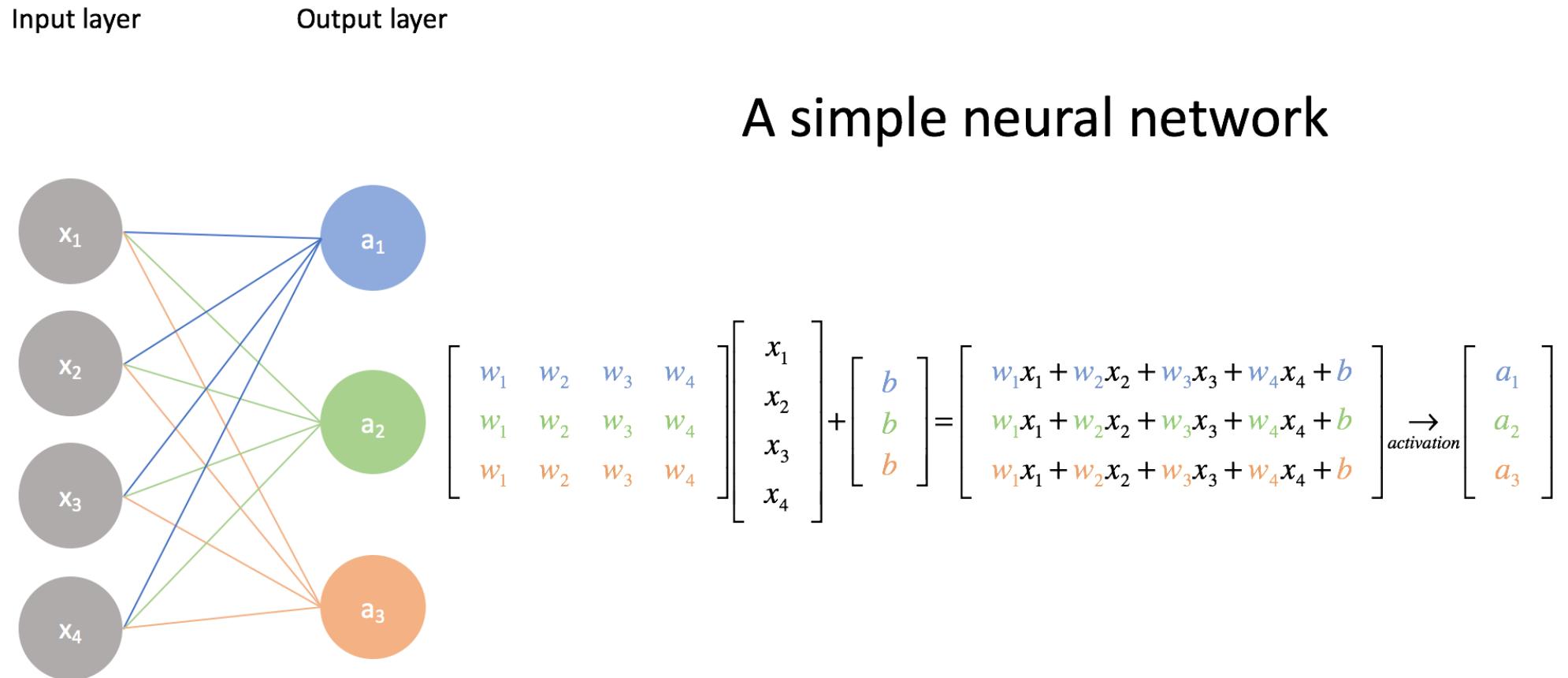
FEEDFORWARD NEURAL NETWORKS (FFNN)

- Remember:



FEED-FORWARD NEURAL NETWORKS (FFNN)

- Remember:

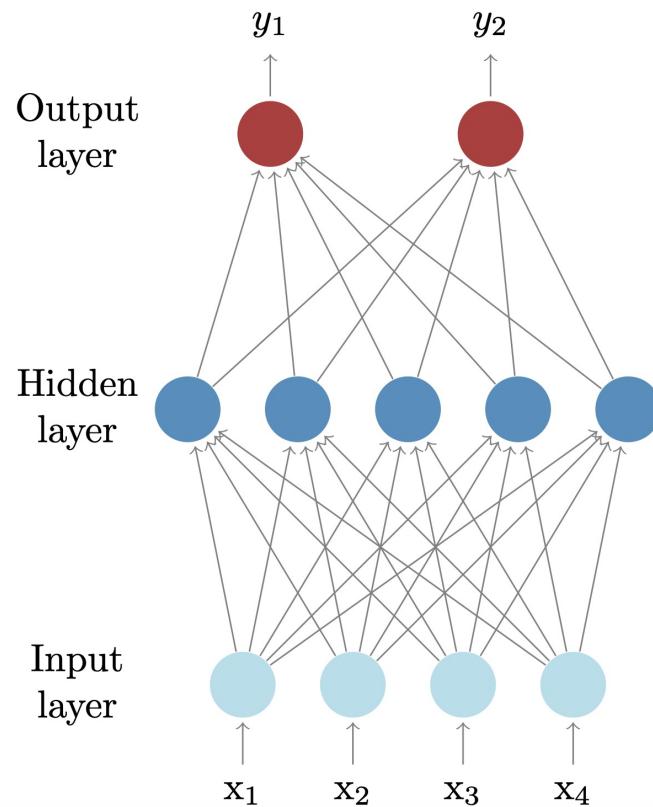


FEED-FORWARD NEURAL NETWORKS

- Remember:
 - Initialization of the weights
 - Take input and process it (forward propagation)
 - Result is compared with the desired output => the error is obtained
 - Try to minimize the error, by updating weights in the network (backward propagation)
 - A learning rate is used

FEEDFORWARD NEURAL NETWORKS

- Problems with FFNN
 - inputs are independent from each other; the same for outputs
 - Unable to handle variable length inputs



FEEDFORWARD NEURAL NETWORKS

But... that is mandatory in NLP

You are right, Ana!

If you want to predict the next word in a sentence (for instance) you better know which words came before it; also, sentences do not have a pre-defined number of words...



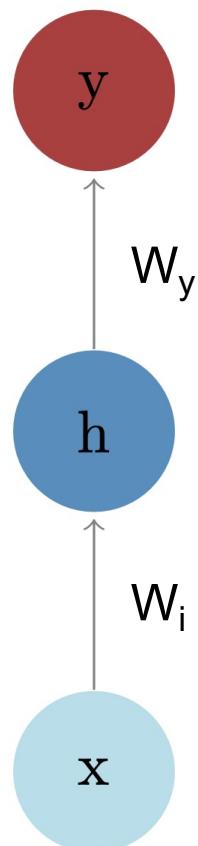
OVERVIEW

- Learning objectives
- Topics
 - Feedforward Neural Networks
 - Recurrent Neural Networks
 - Sequence to Sequence Models
 - Autoencoders
 - Generative Adversarial Networks
 - Attention
 - Transformers
- Key takeaways
- Suggested readings

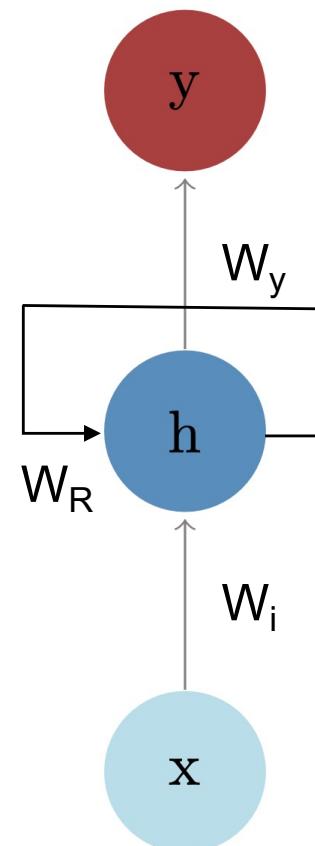
RECURRENT NEURAL NETWORKS

(Elman 1990)

FFNNs

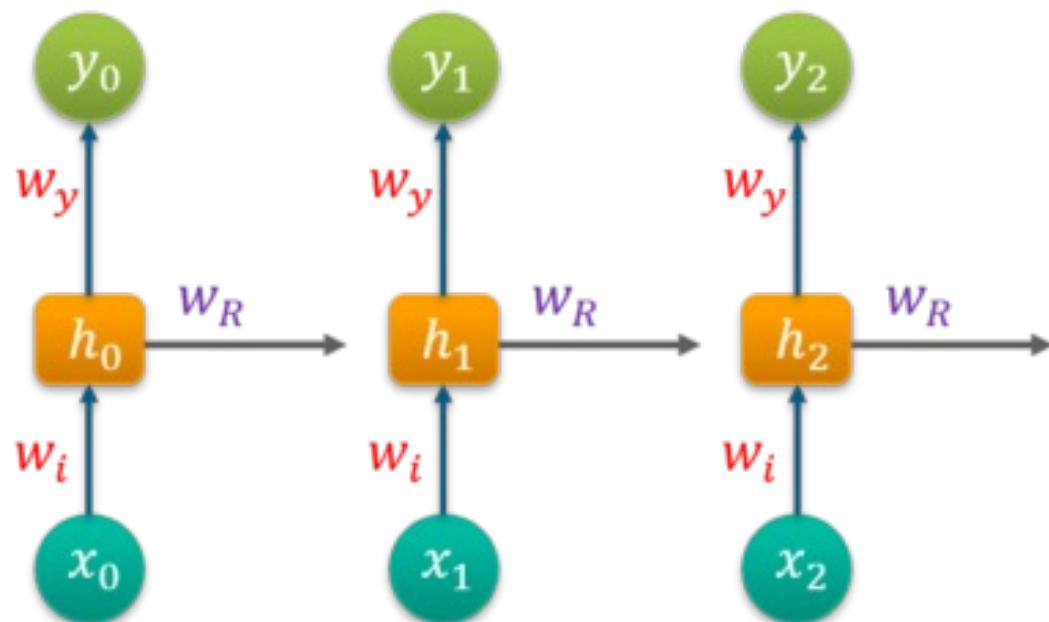


RNNs



RECURRENT NEURAL NETWORKS

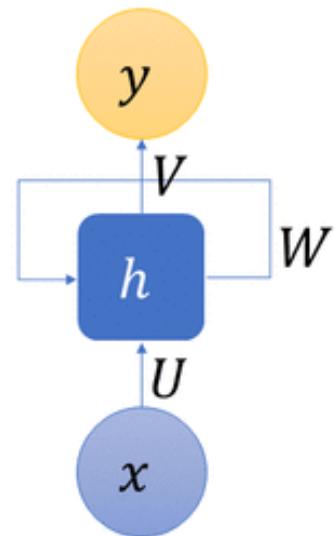
- At time step t
 - x_t is the input, h_t (the “memory”) the hidden state and y_t the output
- The RNN shares the same parameters (W_i , W_y , W_R) across all steps (it uses the same weights for each step)



$$h^{(t)} = g_h (w_i x^{(t)} + w_R h^{(t-1)} + b_h)$$
$$y^{(t)} = g_y (w_y h^{(t)} + b_y)$$

RECURRENT NEURAL NETWORKS

- Backpropagation needs to be adapted:
 - Backpropagation through time



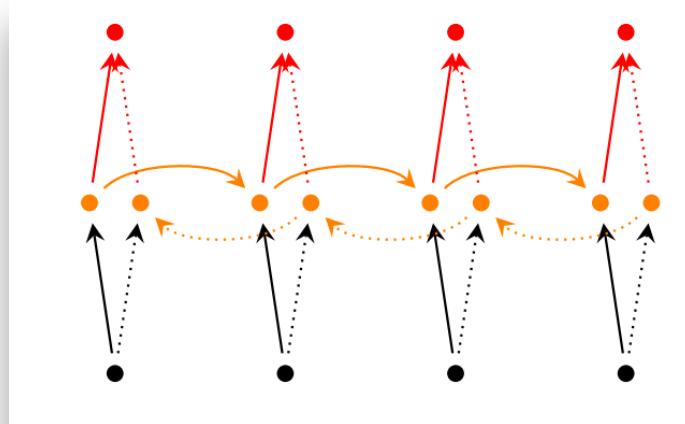
<https://discuss.pytorch.org/t/implementing-backpropagation-through-time/69765>

RECURRENT NEURAL NETWORKS (EXTENSIONS)

- In theory RNNs can make use of information in arbitrarily long sequences, but in practice they are limited to looking back only a few steps
- Extensions:
 - Long short-term memory (LSTM) (Sepp Hochreiter and Jürgen Schmidhuber, 1997)
 - Gated Recurrent Unit (GRU) ([Cho, et al.](#) in 2014)

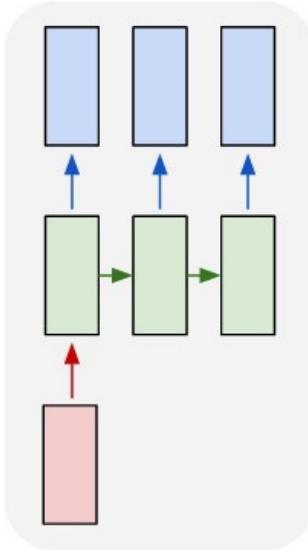
RECURRENT NEURAL NETWORKS (EXTENSIONS)

- Bidirectional RNNs: output at time t may not only depend on the previous elements in the sequence, but also future elements
 - For example, to predict a missing word in a sequence you may want to look at both the left and the right context.

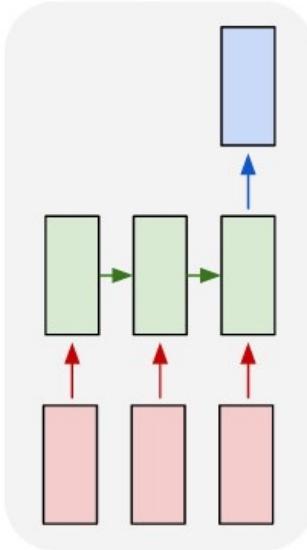


RECURRENT NEURAL NETWORKS (APPLICATIONS IN NLP)

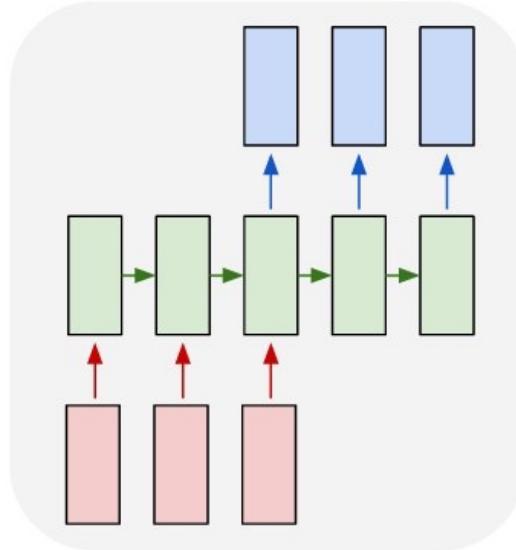
one to many



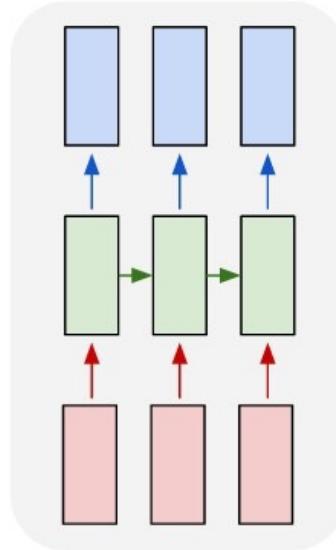
many to one



many to many



many to many



<http://karpathy.github.io/2015/05/21/rnn-effectiveness/>

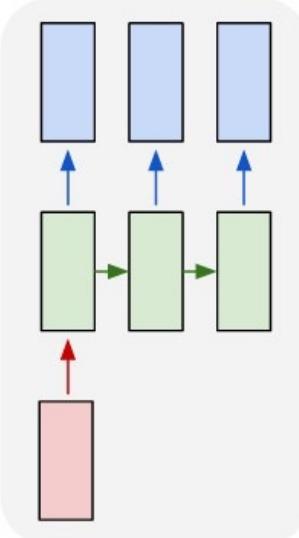
ACTIVE LEARNING MOMENT: THINK-PAIR-SHARE



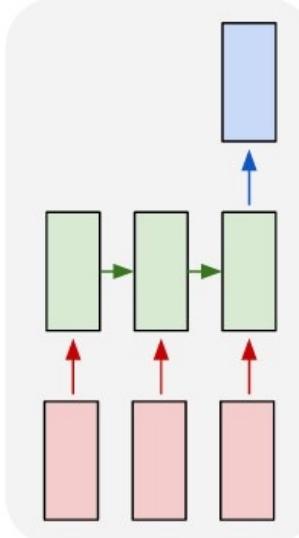
EXERCISE

Give examples of NLP-related applications that follow in these architectures:

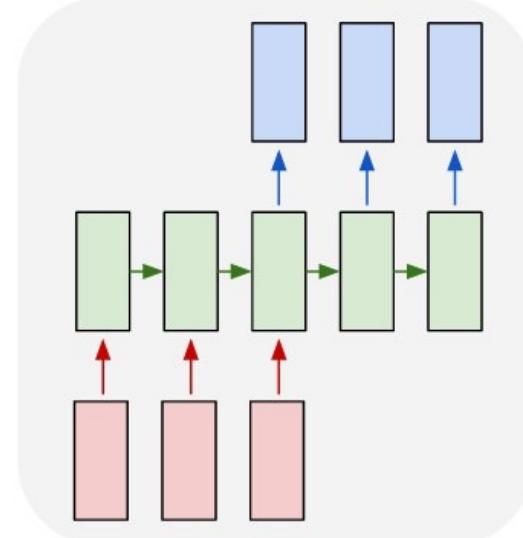
one to many



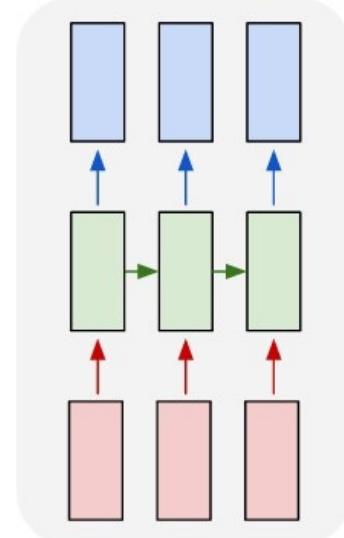
many to one



many to many

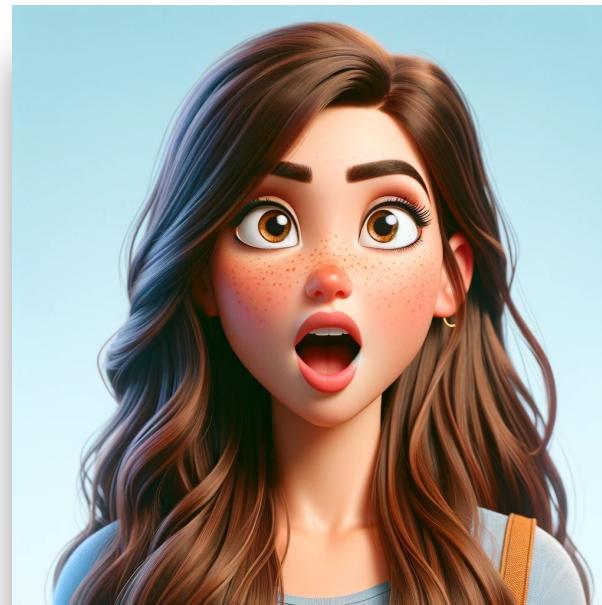


many to many



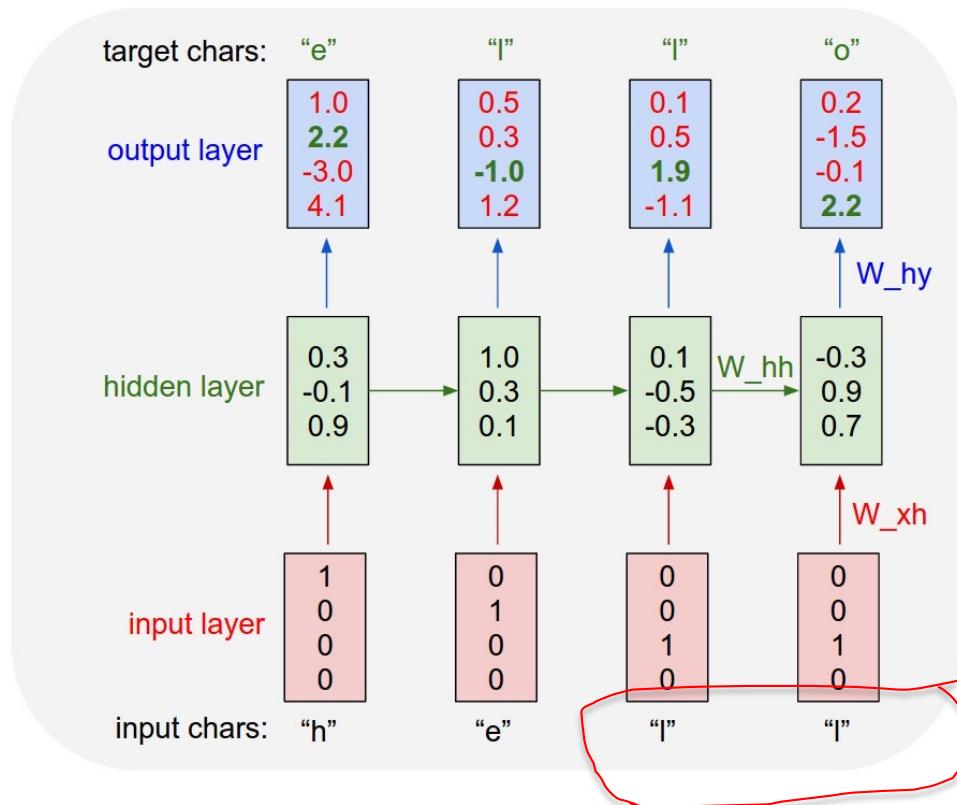
CHARACTER-BASED LANGUAGE MODELS

- Character level language models learn to generate the next character
 - model the probability distribution of the next character in the sequence, being given a sequence of previous characters



CHARACTER-BASED LANGUAGE MODELS

- The following examples are taken from “The unreasonable effectiveness of Recurrent Neural Networks”. This was in 2015!!!



Test time: feed a character and feed next with the resulting character

First target: l; Second target: o
Context/memory is important!

CHARACTER-BASED LANGUAGE MODELS

- Shakespeare example: all the works of Shakespeare (4.4MB file). 3-layer RNN, 512 hidden nodes on each layer. After a few hours of training, obtained:

*“Second Lord: They would be ruled after this chamber,
and my fair nues begun out of the fact, to be conveyed,
Whose noble souls I'll have the heart of the wars.
Clown: Come, sir, I will make did behold your worship.
VIOLA: I'll drink it.”*

CHARACTER-BASED LANGUAGE MODELS

- Wikipedia example: Hutter Prize 100MB dataset of raw Wikipedia. Trained a LSTM. After a few hours of training, obtained:

“Naturalism and decision for the majority of Arab countries' capitalide was grounded by the Irish language by [[John Clair]], [[An Imperial Japanese Revolt]], associated with Guangzham's sovereignty. .”

CHARACTER-BASED LANGUAGE MODELS

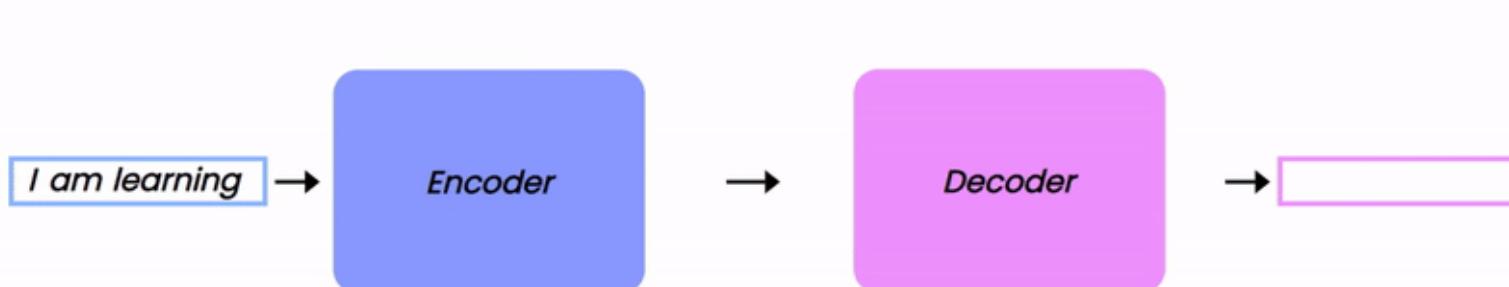
- Latex example: Raw Latex source file of algebraic stacks/geometry book (a 16MB file). Trained a multilayer LSTM:
 - “\begin{proof} We may assume that \mathcal{I} is an abelian sheaf on \mathcal{C} . \item Given a morphism $\Delta : \mathcal{F} \rightarrow \mathcal{I}$ is injective and let \mathfrak{q} be an abelian sheaf on X . Let \mathcal{F} be a fibered complex. Let \mathcal{F} be a category. \begin{enumerate} \item \hyperref[setain-construction-phantom]{Lemma} \label{lemma-characterize-quasi-finite}”

OVERVIEW

- Learning objectives
- Topics
 - Feedforward Neural Networks
 - Recurrent Neural Networks
 - [Sequence to Sequence Models](#)
 - Autoencoders
 - Generative Adversarial Networks
 - Attention
 - Transformers
- Key takeaways
- Suggested readings

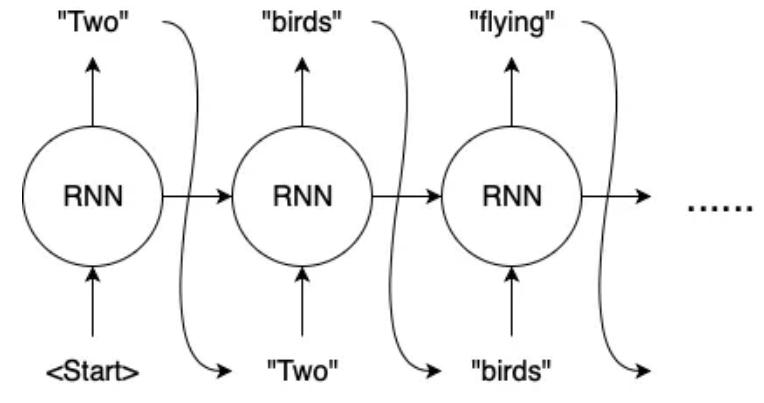
SEQUENCE TO SEQUENCE MODELS (S2S, seq2seq, Google, 2014)

- Seq2Seq models have an encoder-decoder architecture:
 - the encoder processes the input and encodes the information into a context vector (of a fixed length)
 - Idea: context vector is expected to be a good summary of the entire input
 - The decoder is then initialized with this context vector and starts generating the transformed output

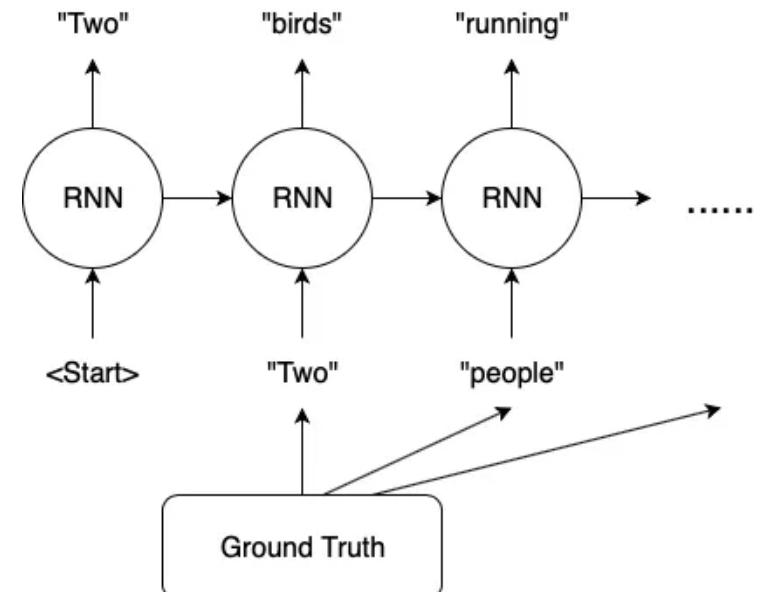


BY THE WAY... TEACHER FORCING (INTERESTING RELATED CONCEPTS)

- Teacher forcing: training technique used for sequence prediction tasks
 - the actual output from the training dataset at the current time step is used as the input at the next time step, rather than using the output generated by the model



Without Teacher Forcing



With Teacher Forcing

TEACHER FORCING (INTERESTING RELATED CONCEPTS)

- Benefits
 - The model learns faster because it is always guided by the correct sequences during training
 - Reduces error accumulation
- Drawbacks:
 - Over-reliance on teacher forcing might prevent the model from learning to recover from its own mistakes



TEACHER FORCING (INTERESTING RELATED CONCEPTS)

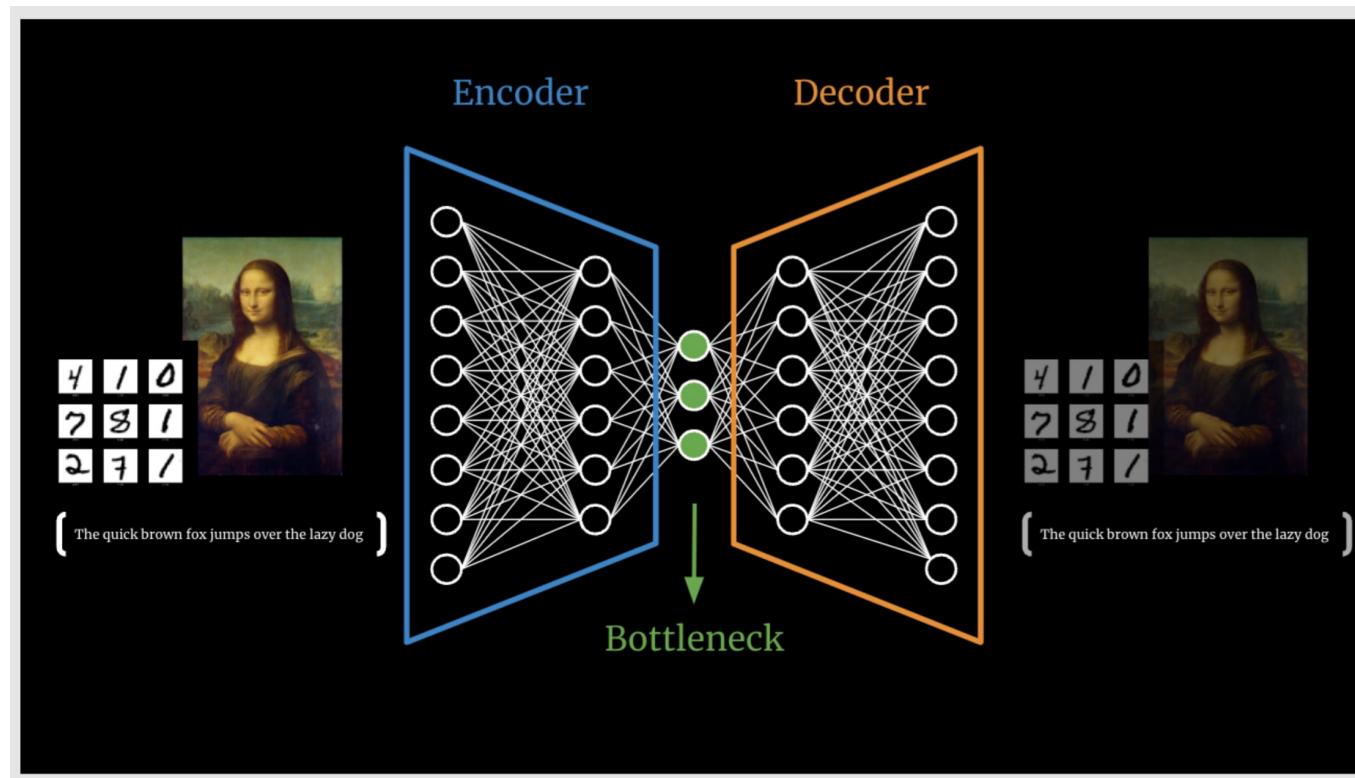
- Balancing Teacher Forcing:
 - [Randomized Teacher Forcing](#): use teacher forcing randomly with a certain probability that can change over time
 - [Scheduled Sampling](#): gradually decrease the use of teacher forcing as training progresses, allowing the model to learn from its own predictions

OVERVIEW

- Learning objectives
- Topics
 - Feedforward Neural Networks
 - Recurrent Neural Networks
 - Sequence to Sequence Models
 - [Autoencoders](#)
 - Generative Adversarial Networks
 - Attention
 - Transformers
- Key takeaways
- Suggested readings

AUTOENCODERS

- Autoencoders learn a representation (encoding) for a set of data
 - keep the variations in the data required to reconstruct the input; ignore redundancies

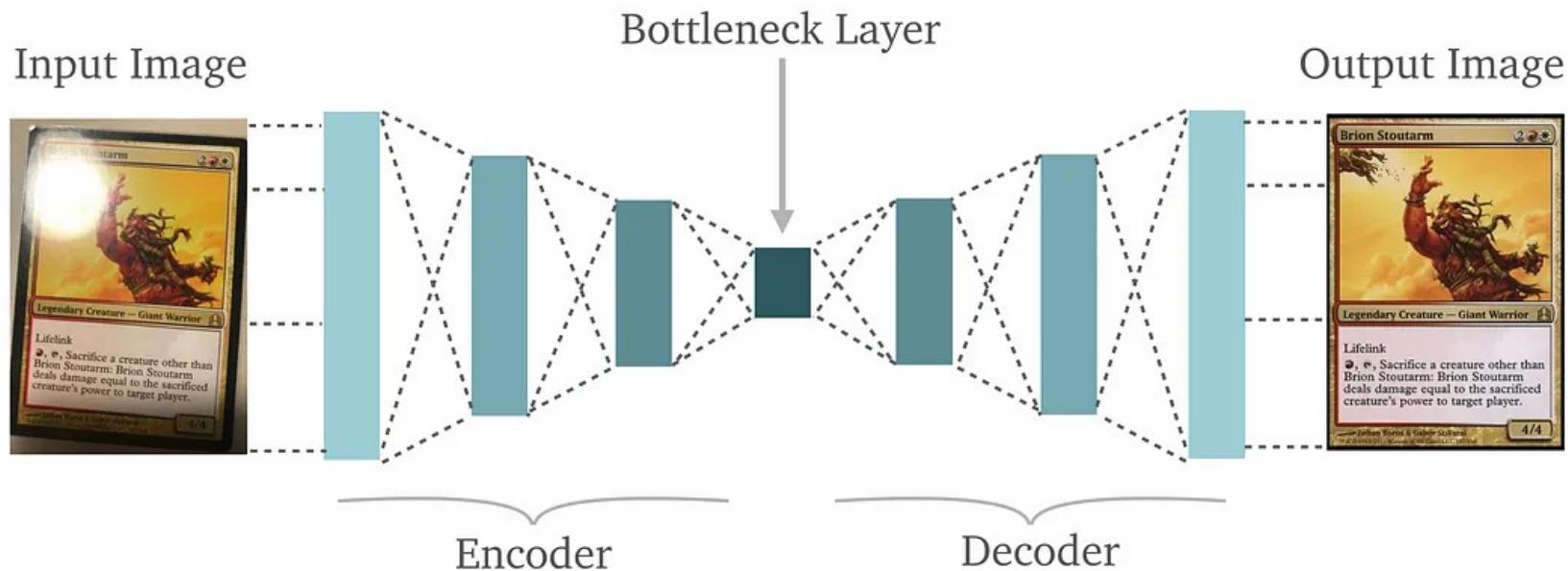


AUTOENCODERS

- Useful for representation learning and dimensionality reduction
 - the vector serving as a hidden representation compresses the data into a smaller number of dimensions
 - after training, only the encoder is used (the decoder is trashed)

AUTOENCODERS (INTERESTING APPLICATION)

- Denoising: during training, is presented with pairs of input data (noisy) and target data (clean).

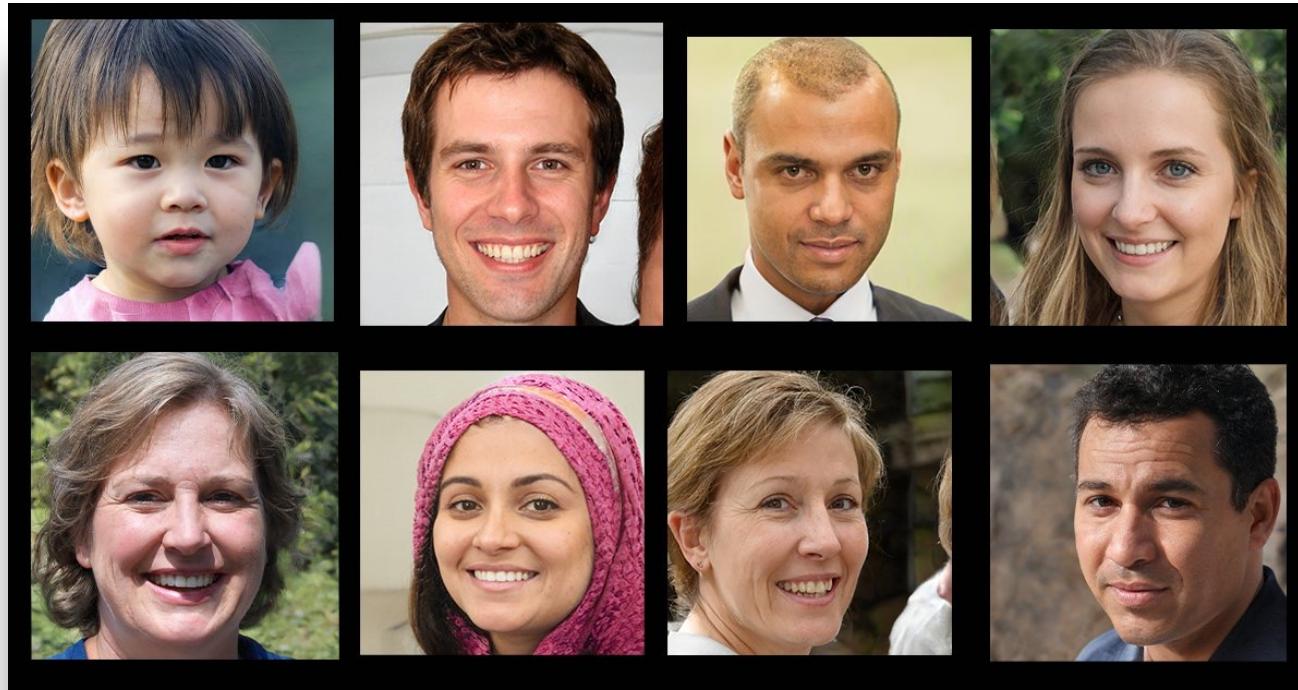


OVERVIEW

- Learning objectives
- Topics
 - Feedforward Neural Networks
 - Recurrent Neural Networks
 - Sequence to Sequence Models
 - Autoencoders
 - Generative Adversarial Networks
 - Attention
 - Transformers
- Key takeaways
- Suggested readings

GENERATIVE ADVERSARIAL NETWORKS (or GANs, Ian Goodfellow, 2014)

- A GAN is a deep neural net architecture that comprised two nets, pitting one against the other (thus “adversarial”)

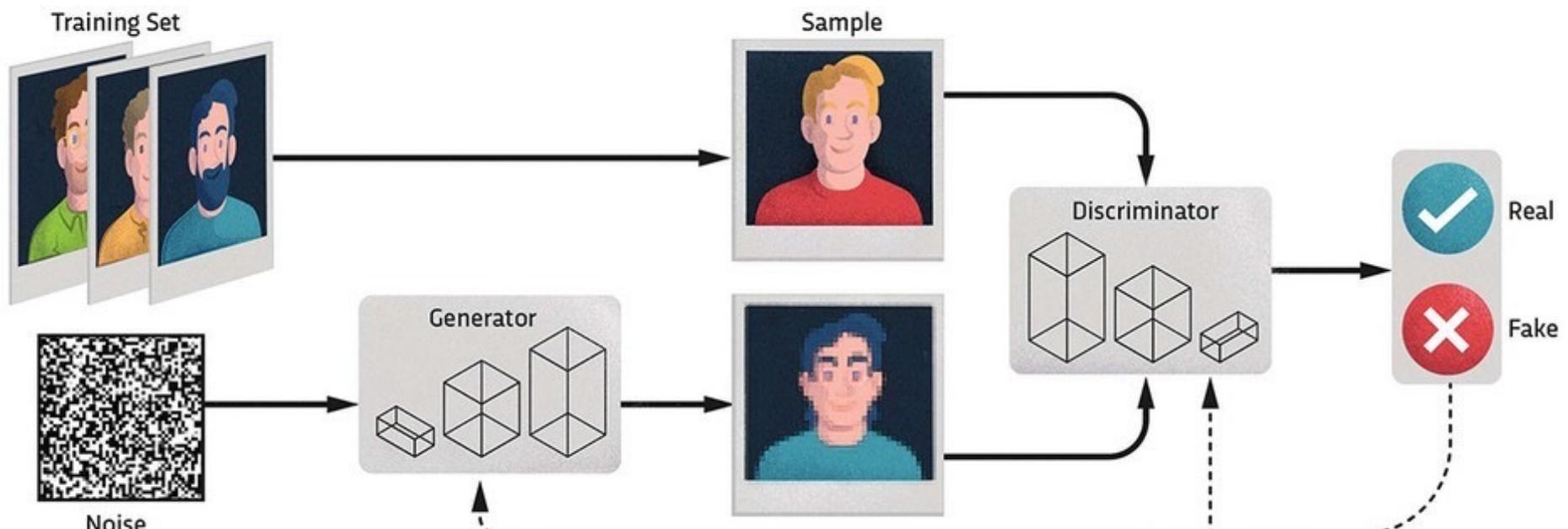


These people do not exist

GENERATIVE ADVERSARIAL NETWORKS

- One neural network ([the generator](#)) generates new data instances
 - The goal of the generator is to produce realistic-looking data samples that are indistinguishable from real data
- The other ([the discriminator](#)) takes input samples, either real or generated by the generator, and predicts whether each sample is real or fake
- [During training](#), the generator and discriminator are trained iteratively
 - The discriminator is trained to maximize its classification accuracy, while the generator is trained to minimize the discriminator's accuracy by generating realistic samples

GENERATIVE ADVERSARIAL NETWORKS (APPLICATIONS)



<https://www.linkedin.com/pulse/exploring-fascinating-realm-generative-adversarial-networks-kaurav/>

ACTIVE LEARNING MOMENT: THINK-PAIR-SHARE



EXERCISE

Consider what you have learned about GENERATIVE ADVERSARIAL NETWORKS (GANs). What is the role of the “Discriminator” within a GAN? Illustrate with an example in NLP.

Possible Answer: the discriminator evaluates the instances generated by the generator for authenticity, that is, it decides whether each instance of data that it reviews belongs to the actual training dataset or not. For instance, in Question Generation, the discriminator will decide if a given question was taken from the training dataset or generated by the generator.



OVERVIEW

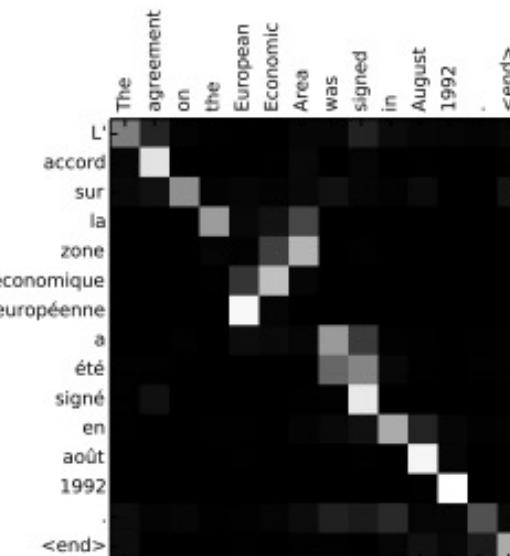
- Learning objectives
- Topics
 - Feedforward Neural Networks
 - Recurrent Neural Networks
 - Sequence to Sequence Models
 - Autoencoders
 - Generative Adversarial Networks
 - Attention
 - Transformers
- Key takeaways
- Suggested readings

ATTENTION MECHANISMS

- Loosely based on the visual attention mechanism found in humans
 - Humans can focus on a certain region with “high resolution”, while perceiving the surroundings in “low resolution”, and then adjusting the focal point over time

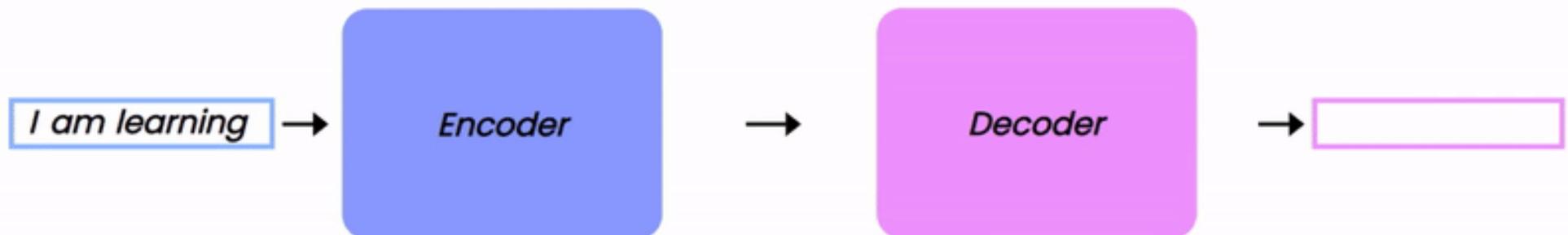
ATTENTION MECHANISMS

- Proposed as a method to both align and translate
 - Alignment: identifies which parts of the input sequence are relevant to each word in the output
 - Translation: use the relevant information to select the appropriate output



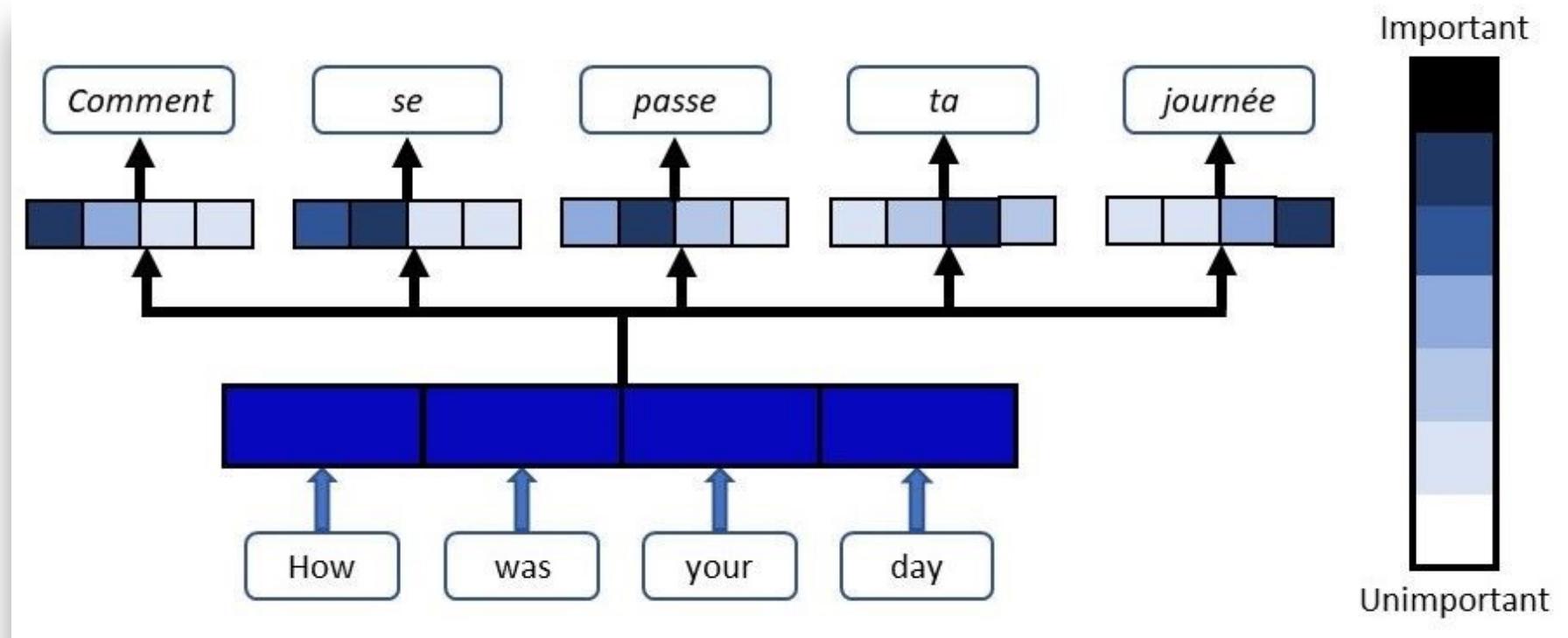
ATTENTION MECHANISMS

- Previously (seq2seq):



ATTENTION MECHANISMS

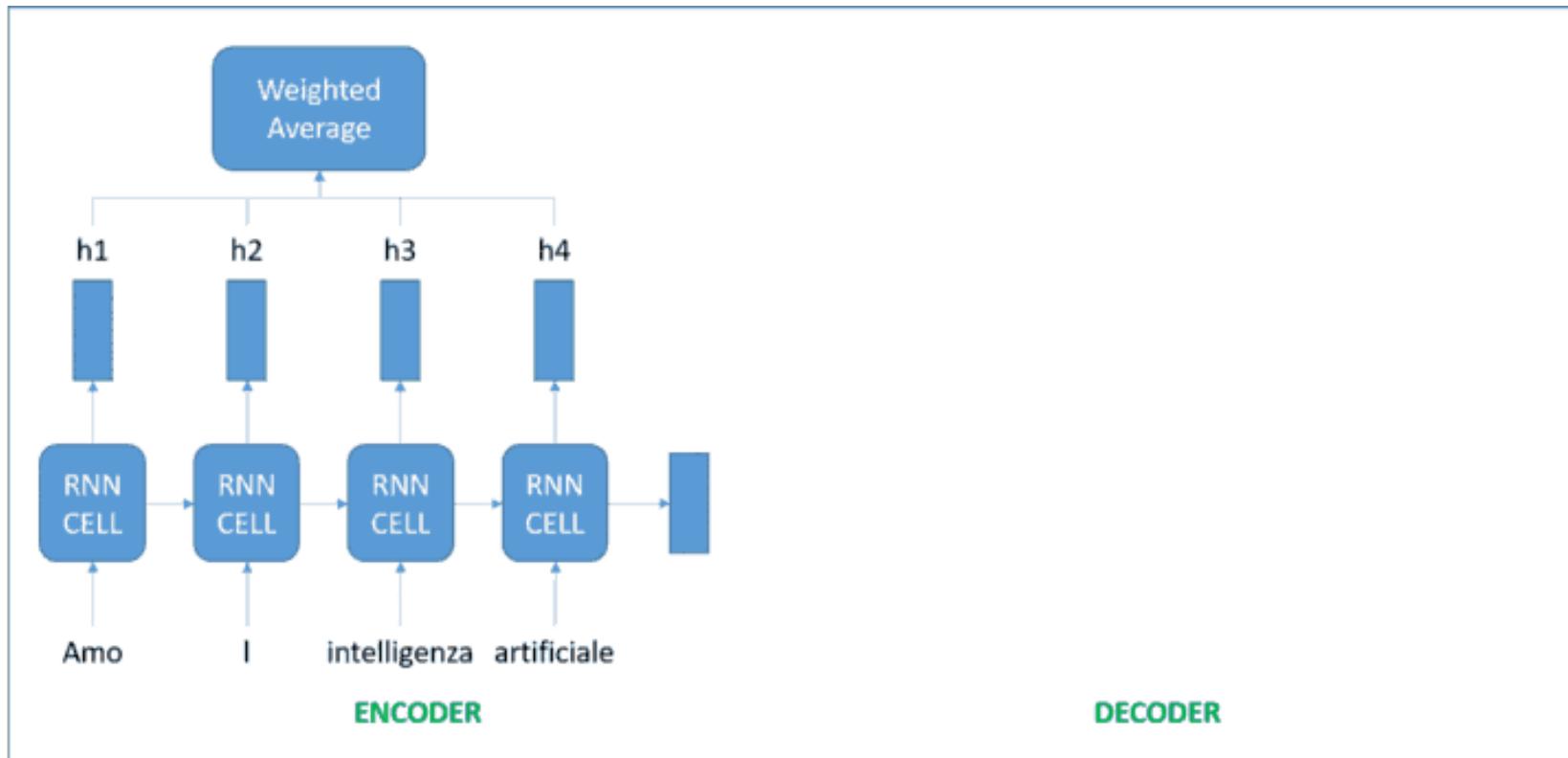
- Now: instead of encoding the input sequence into a single fixed context vector, the attention model develops a context vector for each output time step



ATTENTION MECHANISMS

- That is, we have an attention layer between encoder/decoder
 - the attention mechanism tells the decoder, which is the part of the input to which it should pay more attention (for each output time step).

<https://medium.com/swlh/attention-please-1e16e7011a08>



ATTENTION MECHANISMS

- There are several types of attention mechanisms
 - Soft Attention vs. Hard Attention: the first allows models to weigh the importance of different parts of the input data (what we have seen so far); hard attention selects specific parts of the input to focus on and ignores the rest
 - ...

ATTENTION MECHANISMS

- There are several types of attention mechanisms
 - Self-attention (or Intra-attention): models how some parts of a sentence relate to other parts of the same sentence
 - We will see how self-attention is used in Transformers

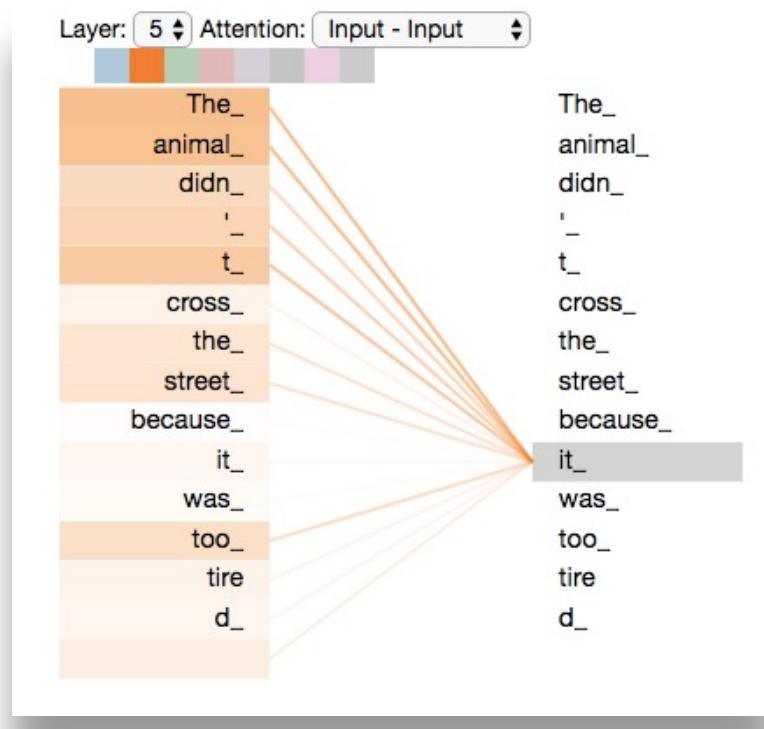


Image from <http://jalammar.github.io/illustrated-transformer/>

ATTENTION MECHANISMS

- There are several types of attention mechanisms
 - Multi-Head Attention: An extension of self-attention that runs several attention mechanisms in parallel. Allows the model to capture different types of relationships in the data, such as syntactic and semantic dependencies
 - We will see how multi-head-attention is used in Transformers

ATTENTION MECHANISMS

- Attention coefficients are also used to explain the decisions of the model



A woman is throwing a frisbee in a park.

ATTENTION MECHANISMS

- Initially used in addition to other architectures, like RNNs
 - However, it performs very well on its own, as combined with feed-forward layers, attention units can simply be stacked, to form encoders.
- Let us see how

OVERVIEW

- Learning objectives
- Topics
 - Feedforward Neural Networks
 - Recurrent Neural Networks
 - Sequence to Sequence Models
 - Autoencoders
 - Generative Adversarial Networks
 - Attention
 - [Transformers](#)
- Key takeaways
- Suggested readings

TRANSFORMERS

Transformers
are about
this paper:

Attention Is All You Need

Ashish Vaswani*

Google Brain

avaswani@google.com

Noam Shazeer*

Google Brain

noam@google.com

Niki Parmar*

Google Research

nikip@google.com

Jakob Uszkoreit*

Google Research

usz@google.com

Llion Jones*

Google Research

llion@google.com

Aidan N. Gomez* †

University of Toronto

aidan@cs.toronto.edu

Lukasz Kaiser*

Google Brain

lukaszkaiser@google.com

Illia Polosukhin* ‡
illia.polosukhin@gmail.com

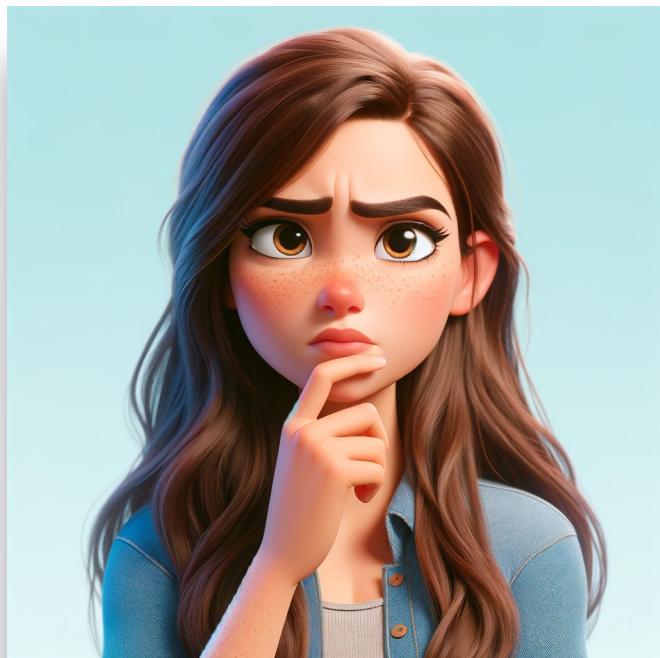
Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.0 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature.



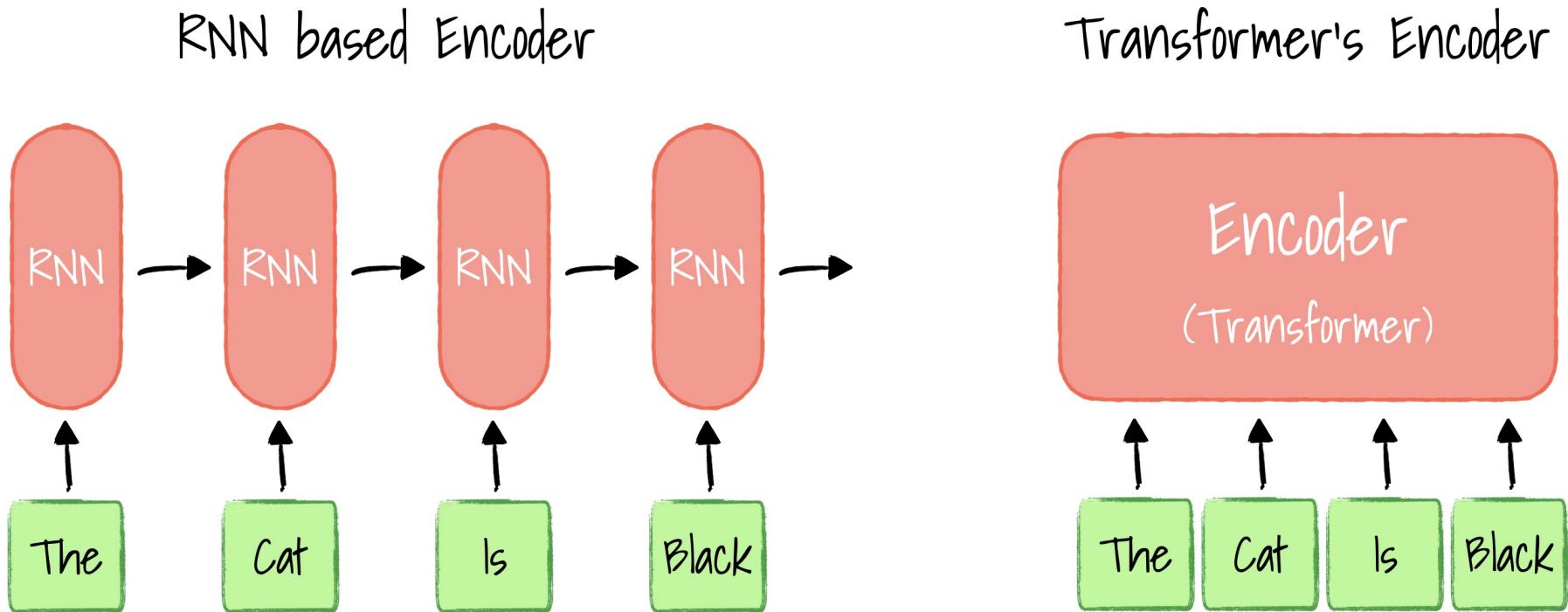
TRANSFORMERS

- What the...!!??



TRANSFORMERS

- Transformers: not recurrent; parallel



<https://towardsdatascience.com/illustrated-guide-to-transformer-cf6969ffa067>

TRANSFORMERS

- Transformers use self-attention mechanisms to gather information about the relevant context of a given word, and then encode that context in the vector that represents the word

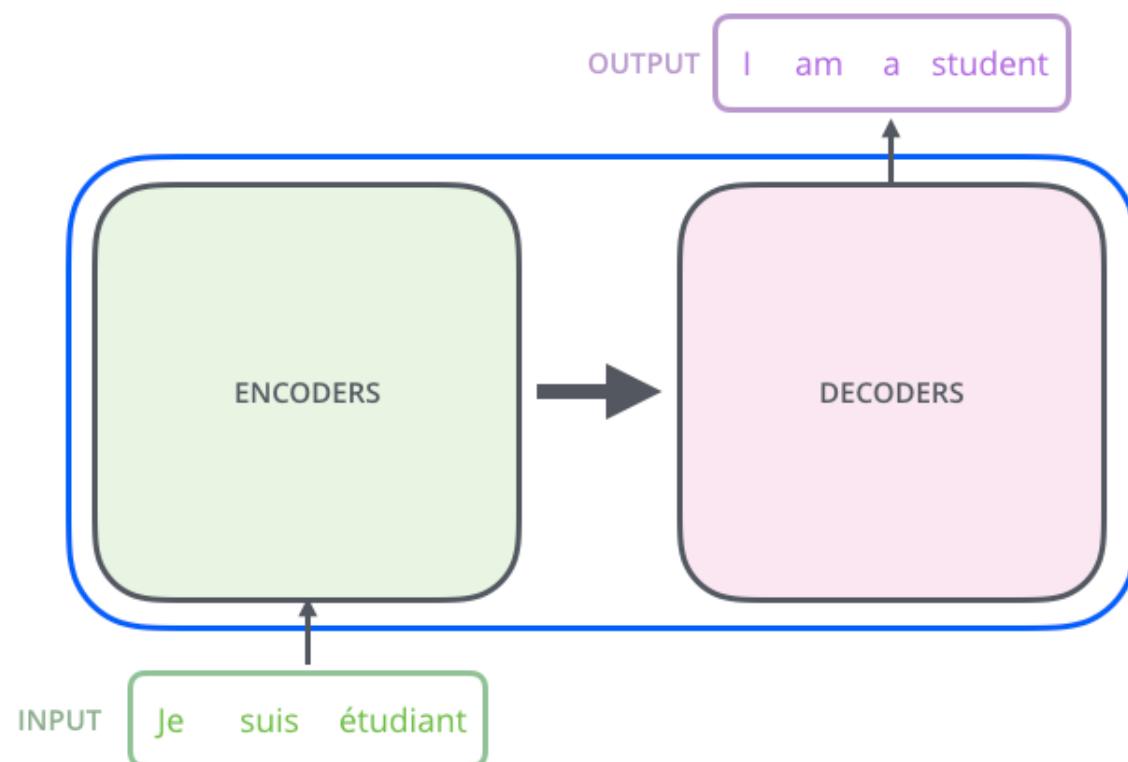
TRANSFORMERS



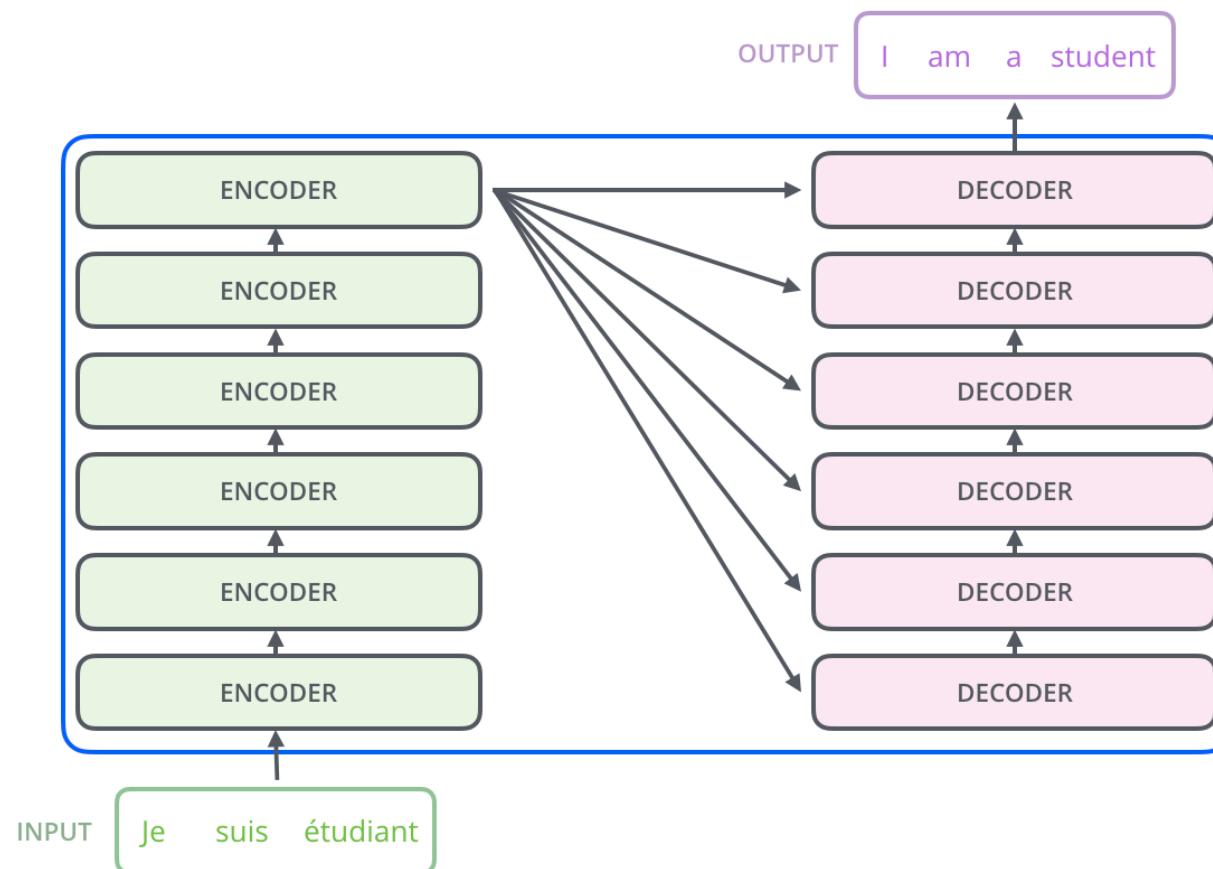
- Most of the following slides/figures are from:
 - The Illustrated Transformer (Jay Alammar):
<http://jalammar.github.io/illustrated-transformer/>



TRANSFORMERS

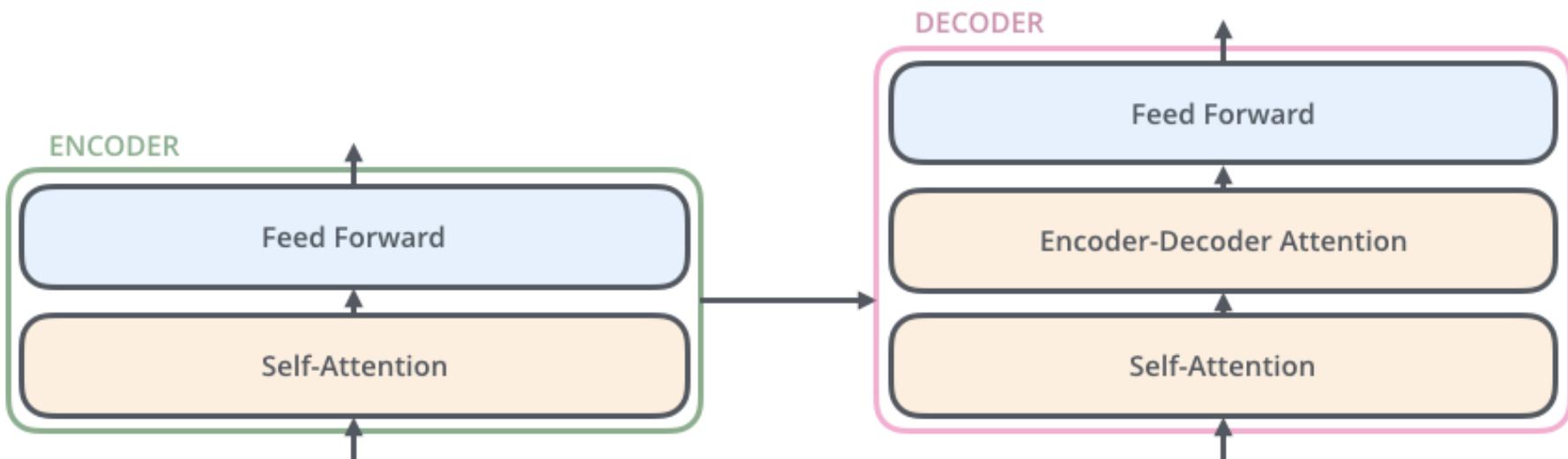


TRANSFORMERS



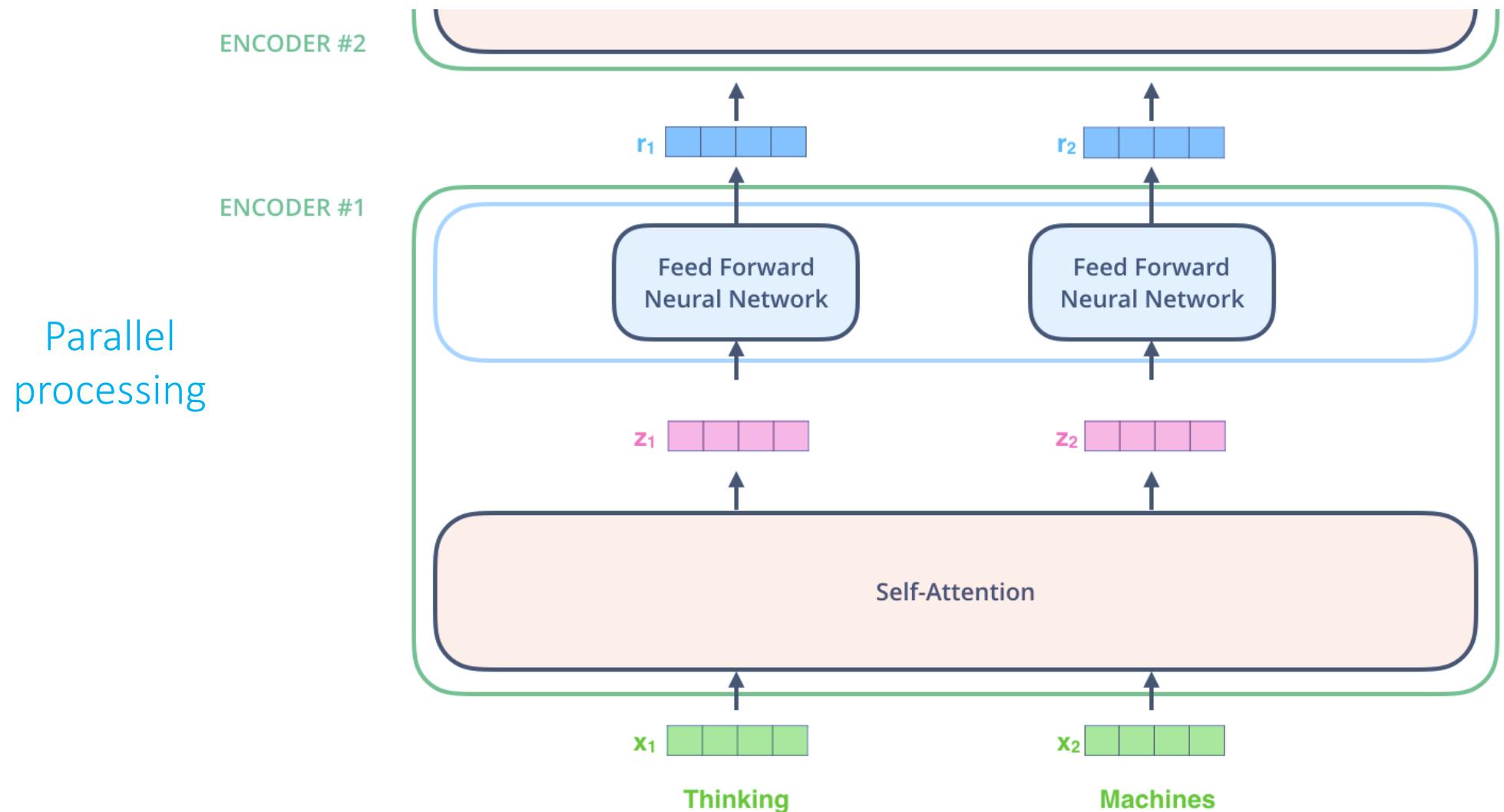
TRANSFORMERS

- General structure of each encoder/decoder:



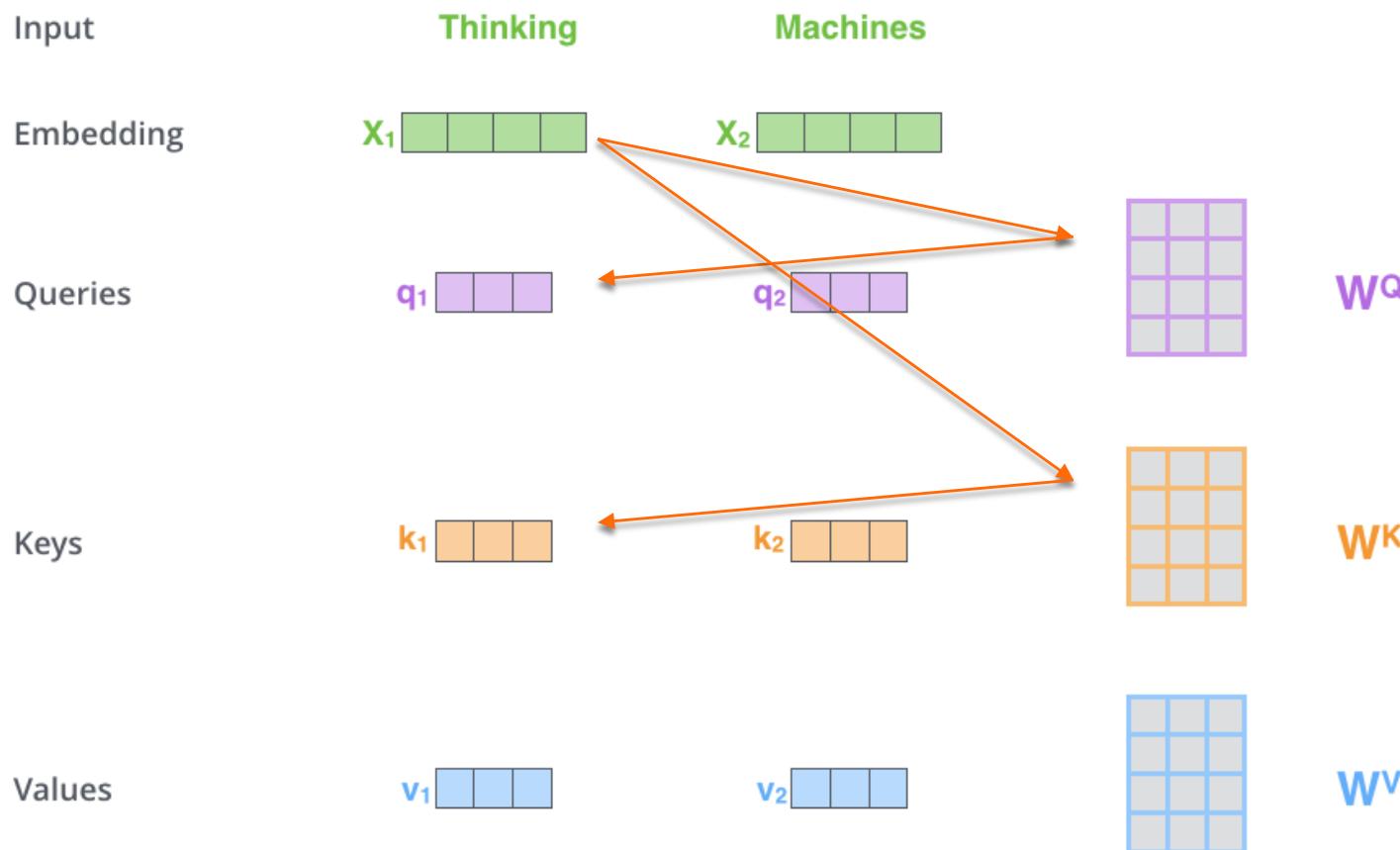
TRANSFORMERS

- Encoder



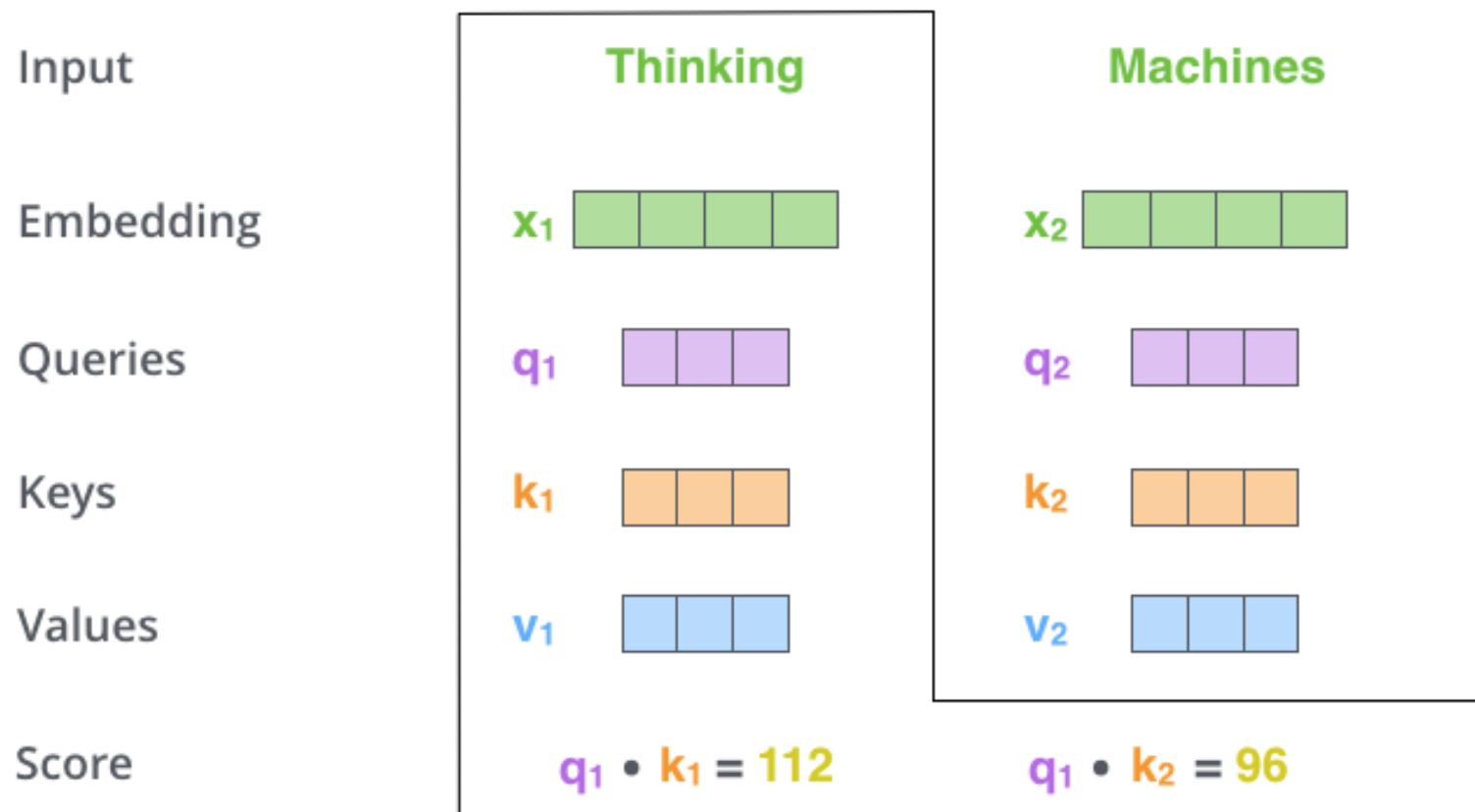
TRANSFORMERS CALCULATING SELF-ATTENTION

- (1) Create vectors **Query**, **Key** and **Value** for each input vector by multiplying each input vector by three matrices (trained during the training process).



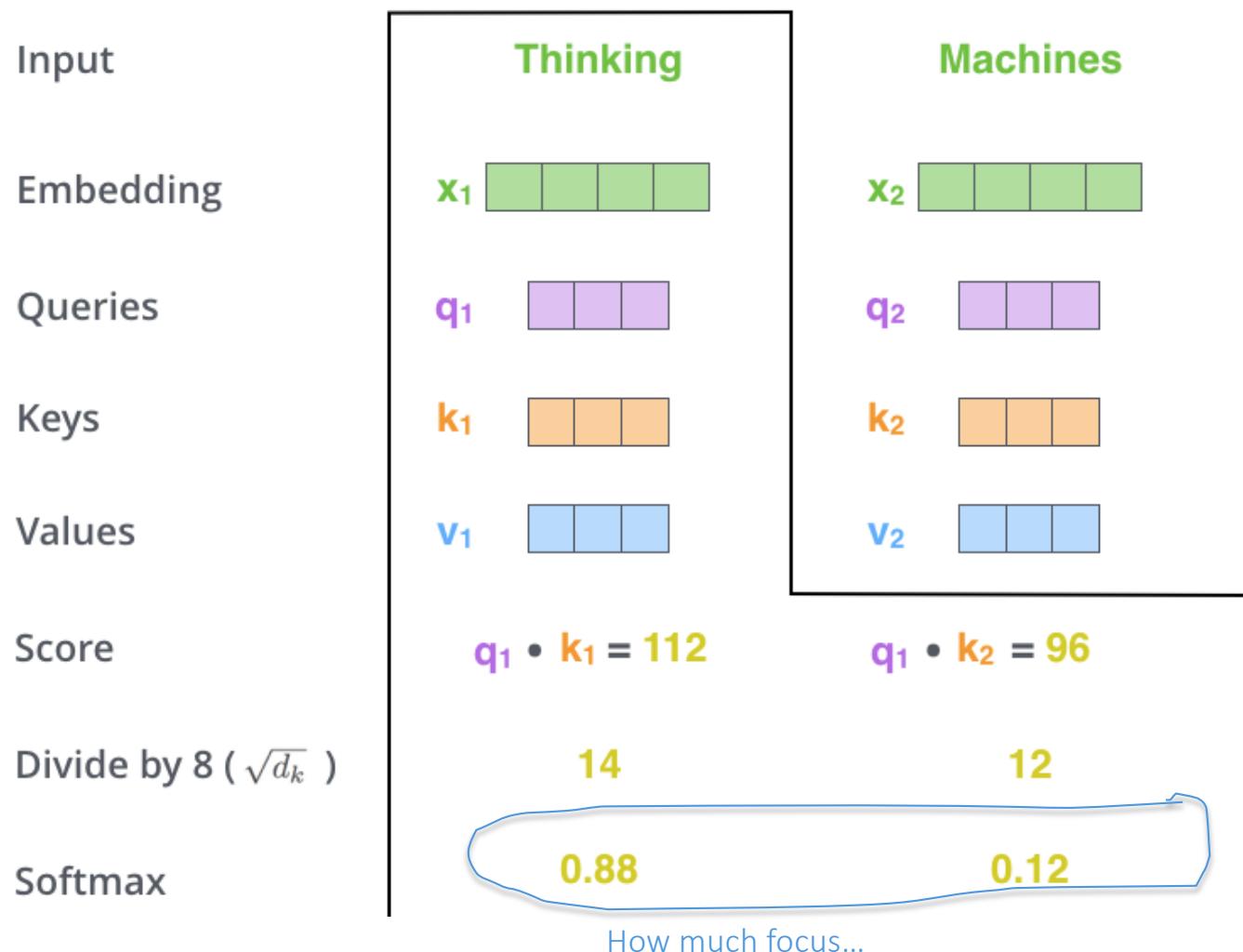
TRANSFORMERS CALCULATING SELF-ATTENTION

- (2) Score each word of the input sentence against the others. The score determines how much focus to place on other parts of the input sentence as we encode a certain word



TRANSFORMERS CALCULATING SELF-ATTENTION

- (3) + (4): divide the scores by 8 (default), then pass the result through a Softmax operation



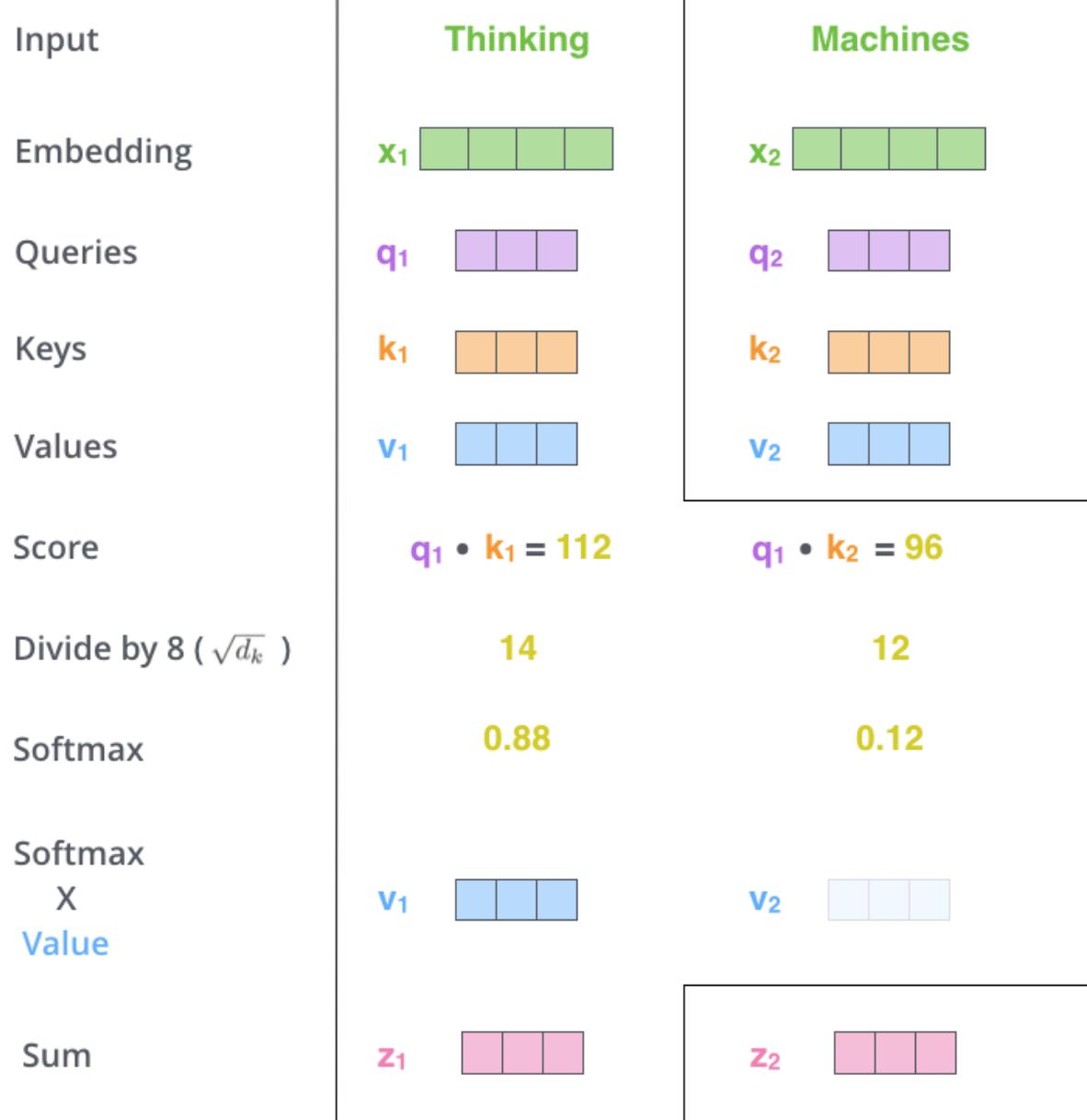
TRANSFORMERS CALCULATING SELF-ATTENTION

- (5): multiply each value vector by the softmax score.

Idea: drown-out irrelevant words.

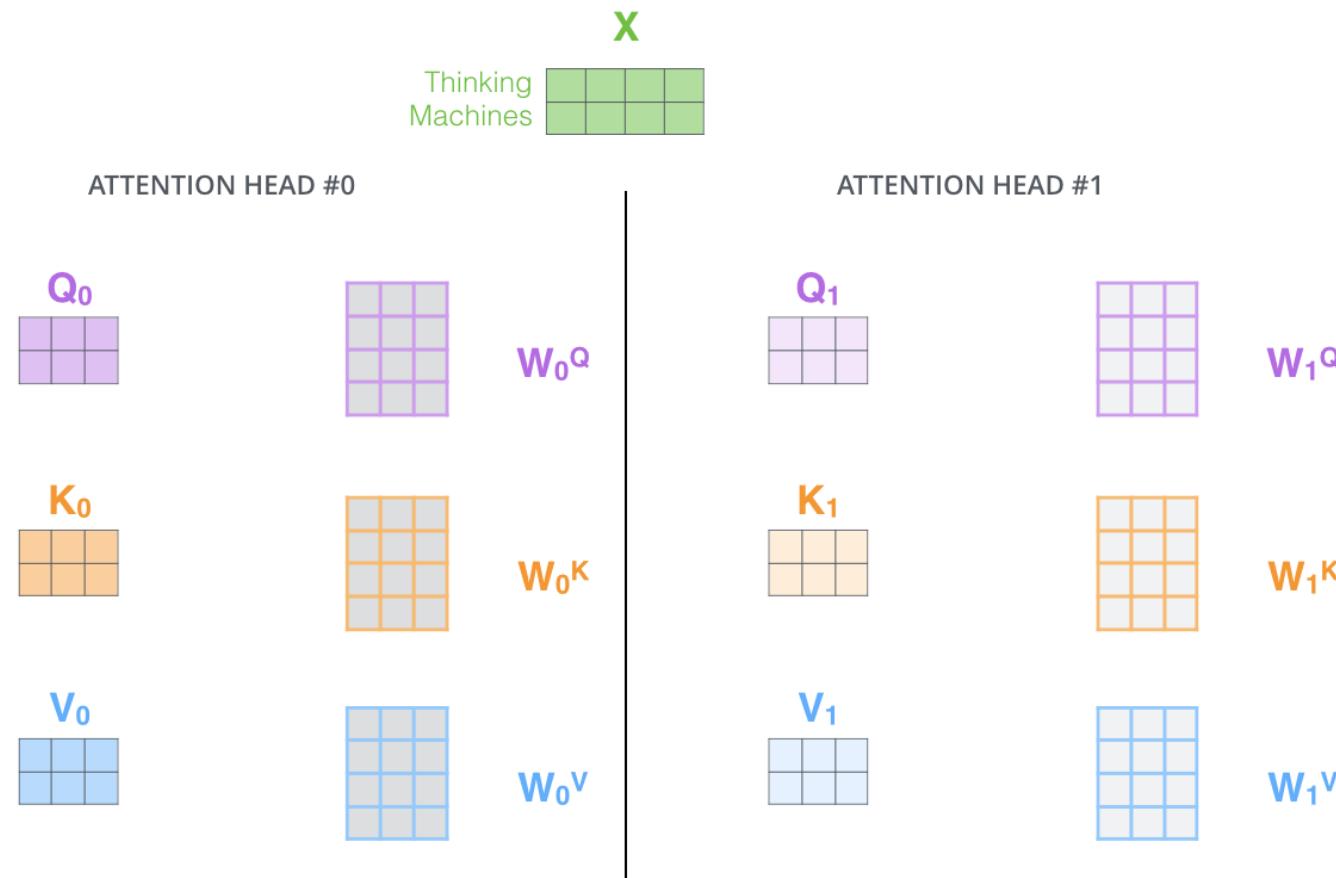
- (6): sum up the weighted value vectors.

- This produces the output of the self-attention layer at this position



TRANSFORMERS MULTI-HEAD ATTENTION

- Multi-head attention expands the model's ability to focus on different positions
 - multiple sets of Query/Key/Value weight matrices (the original Transformer uses 8).



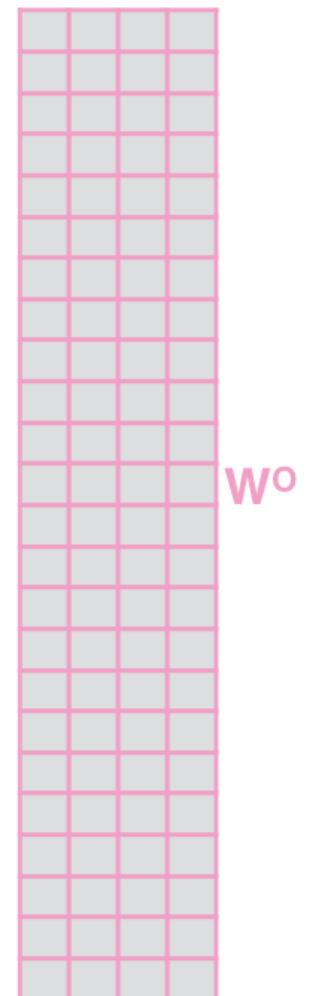
TRANSFORMERS MULTI-HEAD ATTENTION

1) Concatenate all the attention heads



2) Multiply with a weight matrix W^o that was trained jointly with the model

\times



3) The result would be the Z matrix that captures information from all the attention heads. We can send this forward to the FFNN

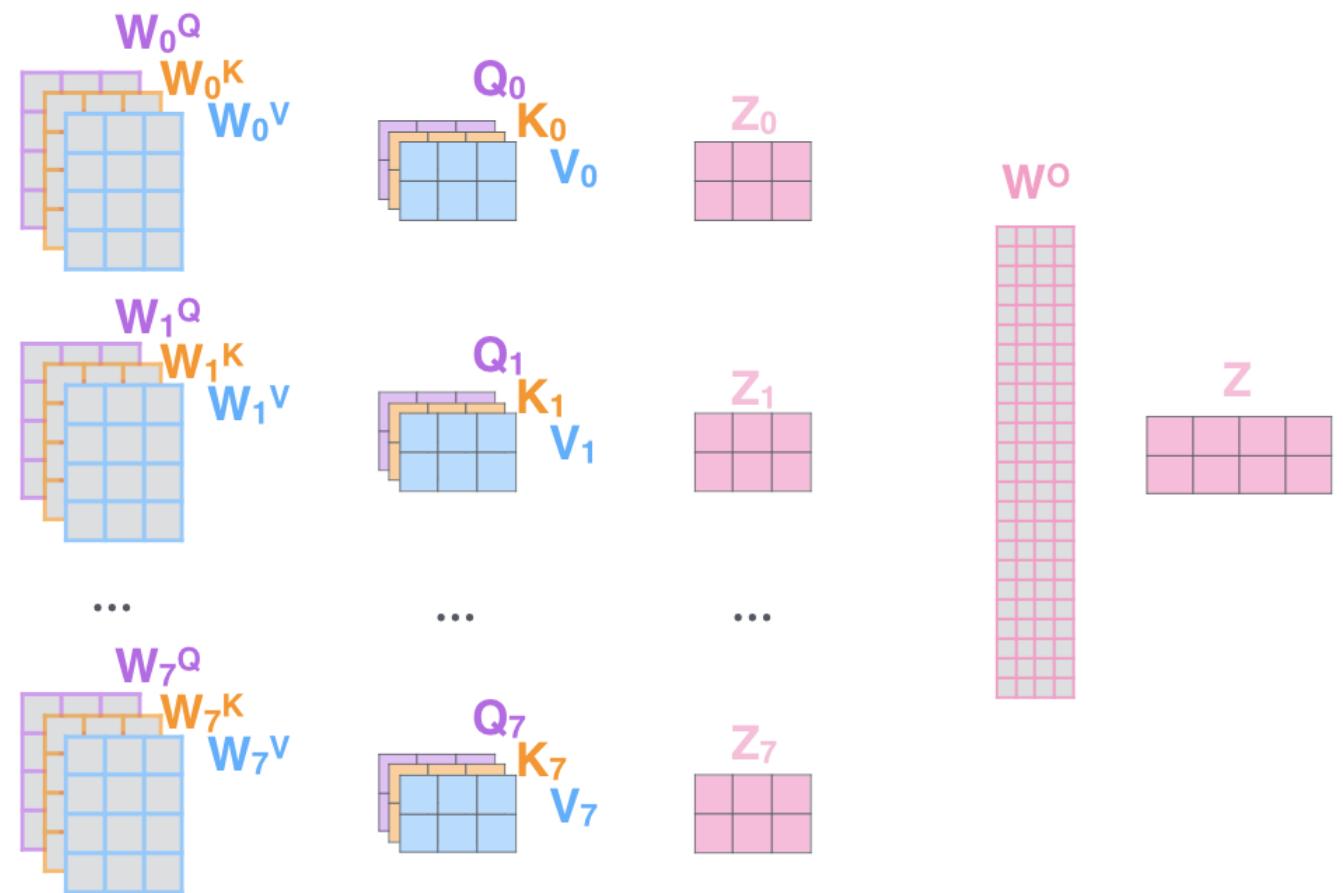
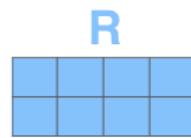
$$= \begin{matrix} Z \\ \begin{matrix} \text{---} & \text{---} & \text{---} & \text{---} \end{matrix} \end{matrix}$$

TRANSFORMERS ALL TOGETHER NOW – ENCODING

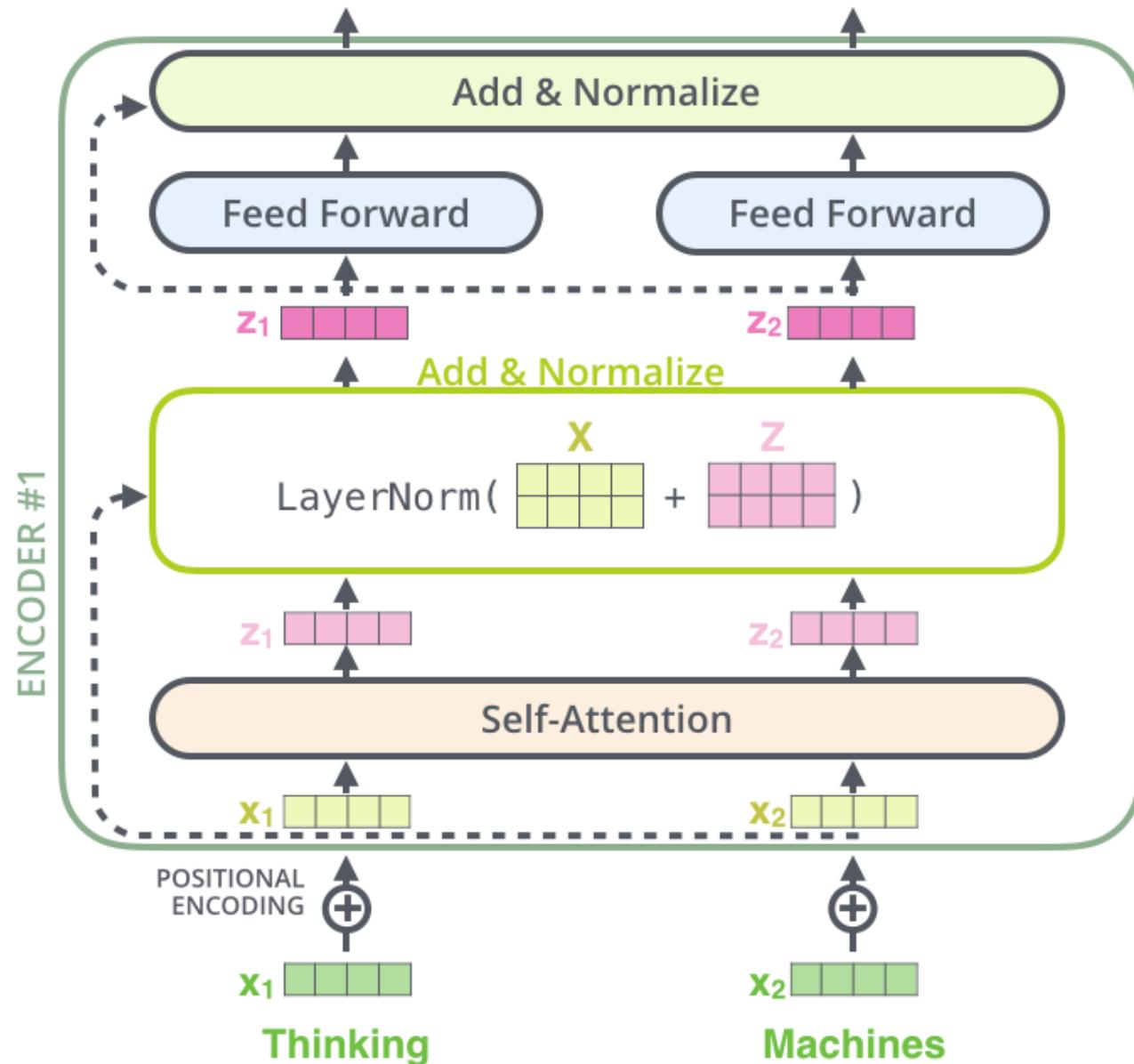
- 1) This is our input sentence* X
- 2) We embed each word* R
- 3) Split into 8 heads. We multiply X or R with weight matrices W_0^Q, W_0^K, W_0^V
- 4) Calculate attention using the resulting $Q/K/V$ matrices
- 5) Concatenate the resulting Z matrices, then multiply with weight matrix W^O to produce the output of the layer



* In all encoders other than #0, we don't need embedding. We start directly with the output of the encoder right below this one

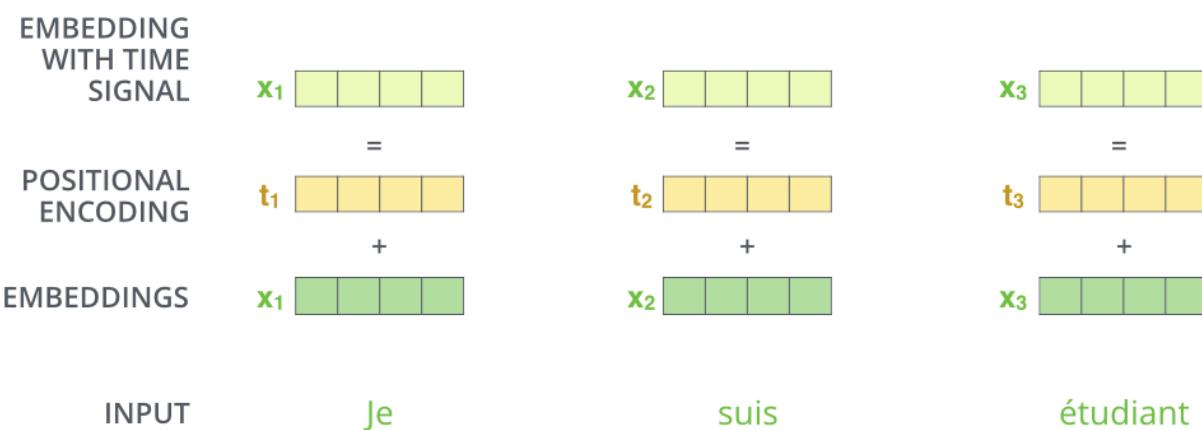
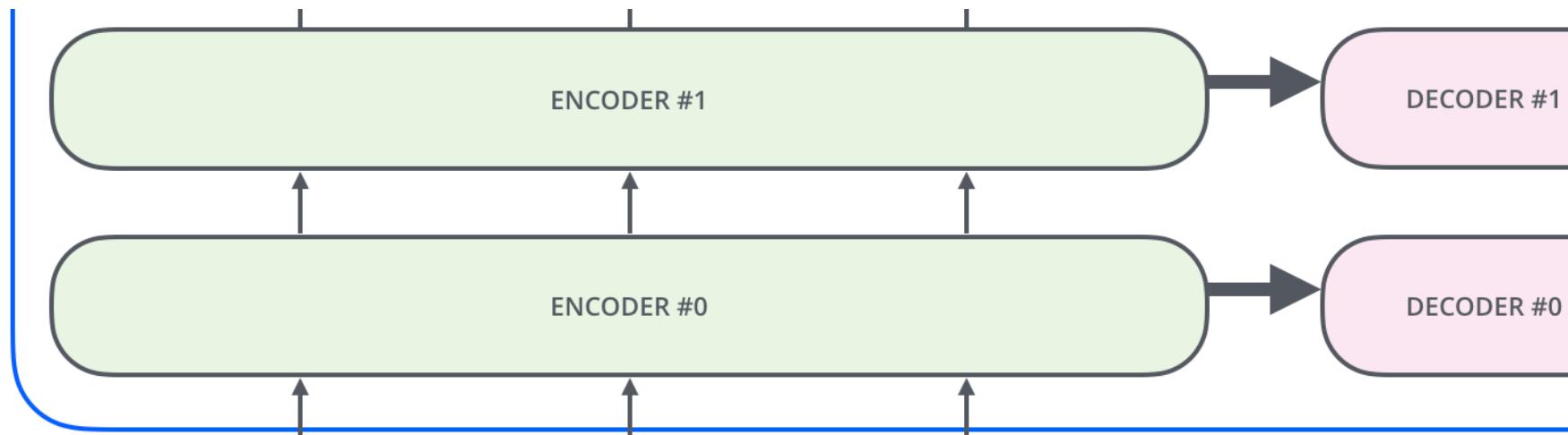


TRANSFORMERS (AND WHEN WE THOUGHT IT WAS OVER)



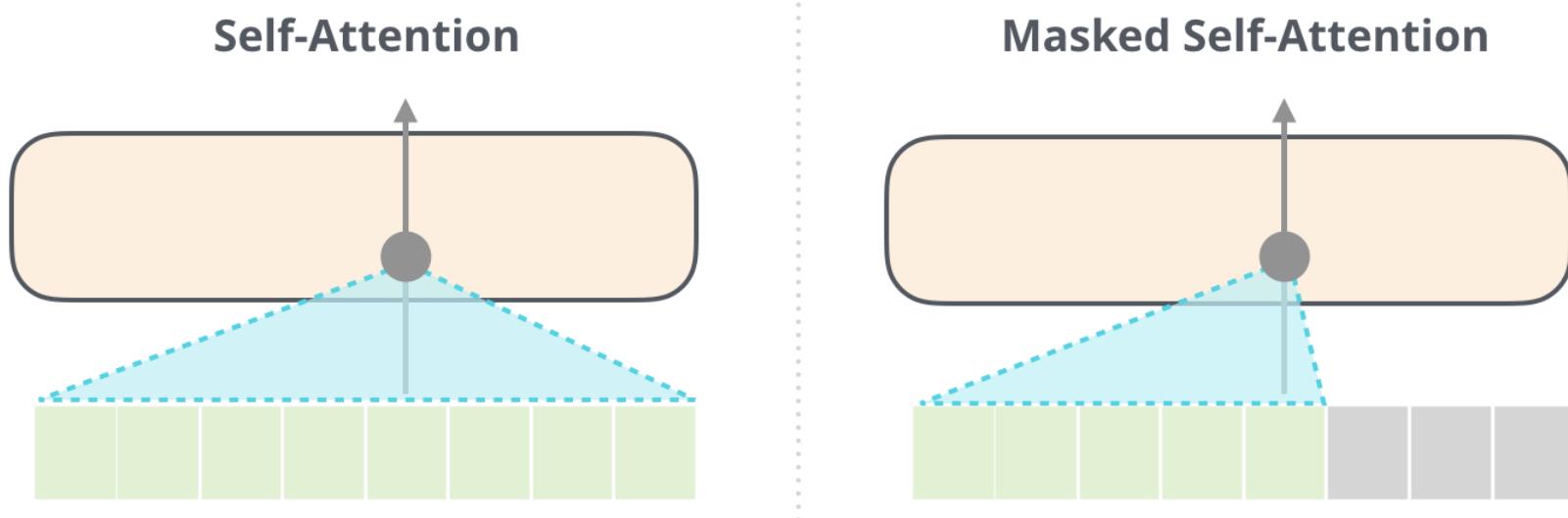
TRANSFORMERS

REPRESENTING THE ORDER OF THE SEQUENCE USING POSITIONAL ENCODING



TRANSFORMERS SEVERAL STEPS AFTERWARDS

- ... we still have the decoder...

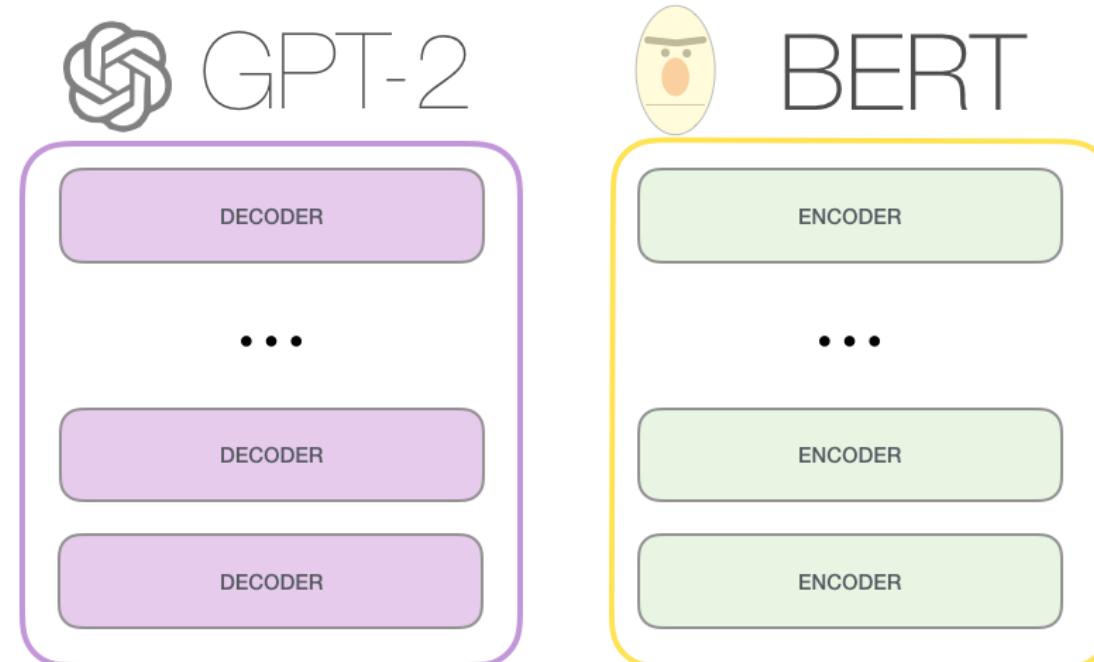


From: The Illustrated GPT-2 (Visualizing Transformer Language Models)

TRANSFORMERS

- But now we can understand...

TRANSFORMERS (APPLICATIONS)



Auto-regressive models

Auto-encoding models

ACTIVE LEARNING MOMENT



Let us try GPT-2 (?)

- <https://play.aidungeon.io/main/home>



- Suggestion:
 - Login, explore a World. I hope the internet will be ok.

KEY TAKEAWAYS

KEY TAKEAWAYS

- Concepts:
 - Feedforward Networks
 - Recurrent Neural Networks
 - RNNs, Bidirectional RNNs, LSTMs, GRUs
 - Sequence to Sequence Models (Seq2Seq)
 - Autoencoders
 - Denoising with autoencoders
 - Generative Adversarial Networks (GANs)
 - Attention Mechanisms:
 - Multi-Head Attention, Soft Attention, Hard Attention, Spatial Attention, Temporal Attention, Cross-Attention
 - Transformers
- Understand how we moved from FFNN to Transformers (and other architectures)

SUGGESTED READINGS

READINGS

- Jurafsky, Chapter 9, 10.1, 10.2, 10.3
- “The Illustrated Transformer” (Jay Alammar)