



aivancity

SCHOOL FOR

TECHNOLOGY, BUSINESS & SOCIETY

PARIS-CACHAN



03/02/2025

# Natural Language Processing (NLP)

*Introduction to NLP and Distributional Semantics*

# Quick word about me

- Postdoc Researcher at ISIR-CNRS (Sorbonne University)
- PhD from Paris-Saclay University (LISN-CNRS lab)
- Research topic: Multimodal and Multilingual NLP
- More about me: <https://paullerner.github.io>
- Contact: [lerner@isir.upmc.fr](mailto:lerner@isir.upmc.fr)

# Acknowledgements

This class directly builds upon:

- **Jurafsky, D., & Martin, J. H.** (2024). Speech and Language Processing : An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models (3rd éd.).
- **Eisenstein, J.** (2019). Natural Language Processing. 587.
- **Yejin Choi.** (Winter 2024). CSE 447/517: Natural Language Processing (University of Washington - Paul G. Allen School of Computer Science & Engineering)
- **Noah Smith.** (Winter 2023). CSE 447/517: Natural Language Processing (University of Washington - Paul G. Allen School of Computer Science & Engineering)
- **Benoit Sagot.** (2023-2024). *Apprendre les langues aux machines* (Collège de France)
- **Chris Manning.** (Spring 2024). Stanford CS224N: Natural Language Processing with Deep Learning
- Classes where I was/am Teacher Assistant:
  - **Christopher Kermorvant.** Machine Learning for Natural Language Processing (ENSAE)
  - **François Landes** and **Kim Gerdes.** Introduction to Machine Learning and NLP (Paris-Saclay)

Also inspired by:

- My PhD thesis: *Répondre aux questions visuelles à propos d'entités nommées* (2023)
- **Noah Smith** (2023): Introduction to Sequence Models (LxMLS)
- **Kyunghyun Cho:** Transformers and Large Pretrained Models (LxMLS 2023), Neural Machine Translation (ALPS 2021)
- My former PhD advisors **Olivier Ferret** and **Camille Guinaudeau** and postdoc advisor **François Yvon**
- My former colleagues at LISN

# Program for this semester

- This morning: What is NLP? What is a word? How do you get a sense of a word?
  - NLP = research field at the intersection of Computer Science and Linguistics / Technology at the heart of chatbots like ChatGPT
  - Meaning of a word is its use in the language: distributional semantics
- This afternoon: Practical Work 1 (2 sessions)
- Wednesday morning: Neural Network architectures used in Large Language Models:
  - Attention Mechanism
  - Transformers
- Wednesday afternoon: Practical Work 2 (2 sessions)

# Program for this semester

- Group Homework: Deadline Thursday 27th of February
  - Groups of 3
  - Report of max. 4 pages, Continuous assessment (50%)
- 4th of March morning: Large Language Models from Shannon to ChatGPT
  - pre-training and fine-tuning
  - alignment: reinforcement learning from human feedback (RLHF)
  - decoding/generation methods

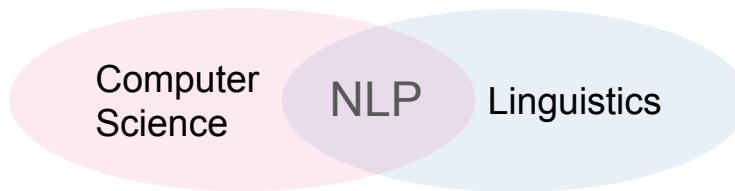
# Program for this semester

- 4th of March afternoon: Practical Work 3 (2 sessions)
- 6th of March morning:
  - Industrial applications and research benchmarks
  - Ethical, social, and environmental issues
- 6th of March afternoon: Practical Work 4 (2 sessions)
- Individual Final sitting Exam 50%

# Background

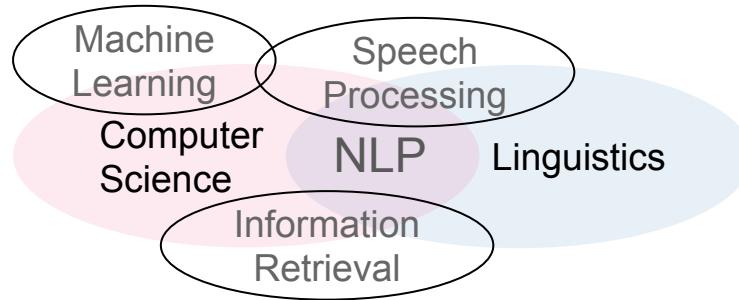
- Assumes basics in Machine Learning, Python, and Linear Algebra
  - Good exercises /cheatsheet "Day 0: Basic Tutorials" of [http://lxmbs.it.pt/2023/documents/LxMLS\\_guide\\_2023.pdf](http://lxmbs.it.pt/2023/documents/LxMLS_guide_2023.pdf)
- Recommended readings:
  - Jurafsky, D., & Martin, J. H. (2024). Speech and Language Processing : An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models.  
[https://web.stanford.edu/~jurafsky/slp3/ed3bookaug20\\_2024.pdf](https://web.stanford.edu/~jurafsky/slp3/ed3bookaug20_2024.pdf)
  - Eisenstein, J. (2019). Natural Language Processing.  
<https://nlp.cs.princeton.edu/cos484-sp21/readings/eisenstein-nlp-notes.pdf>

# Natural Language Processing (NLP)



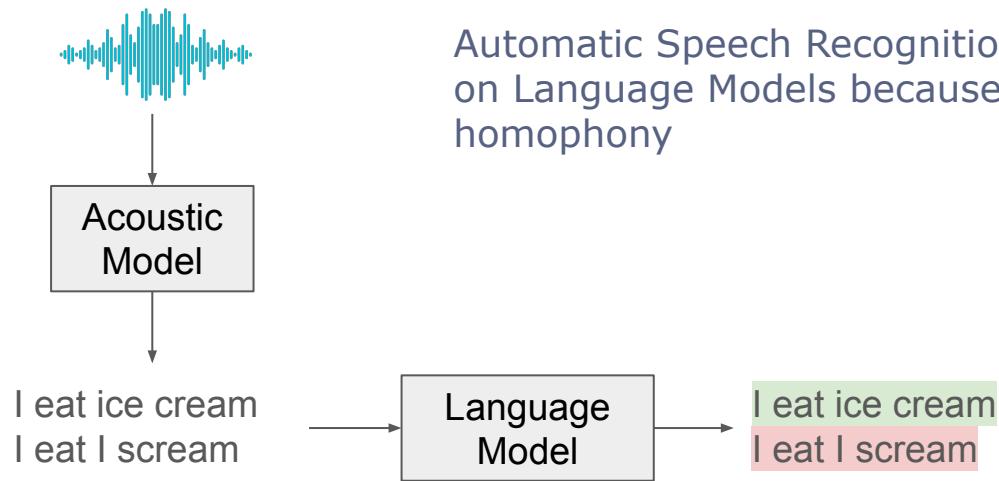
- Intersection of Computer Science and Linguistics:
  - Distributional Semantics: sense of a word from its context (today class)
  - Computational Linguistics, Computational Morphology, etc.: study of humans: how do we speak? how do we organize lexicon?

# Natural Language Processing (NLP)

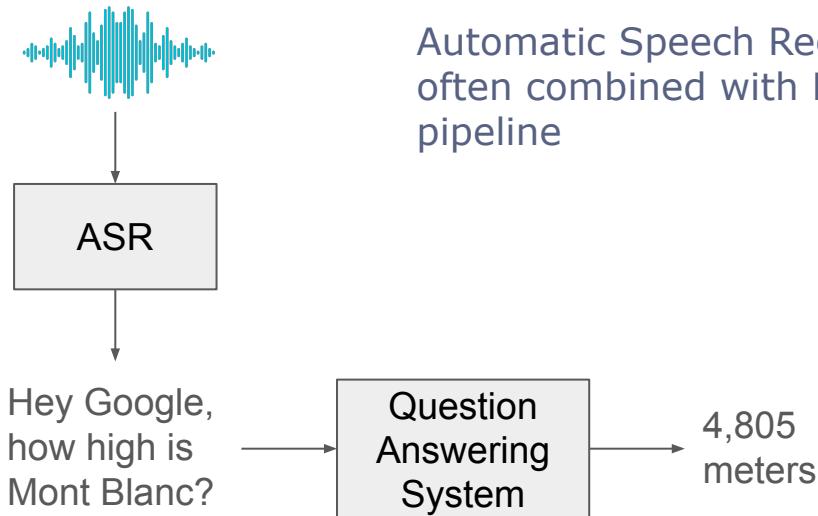


- Close to Speech Processing (Automatic Speech Recognition etc.)
- Close to Information Retrieval (Search engines like Google)
- Driven by Statistical/Machine Learning methods since the 90s
- Driven by Deep Learning since 2010s

# Speech and Language Processing



# Speech and Language Processing



Automatic Speech Recognition (ASR) is often combined with NLP methods in a pipeline

# Speech and Language Processing

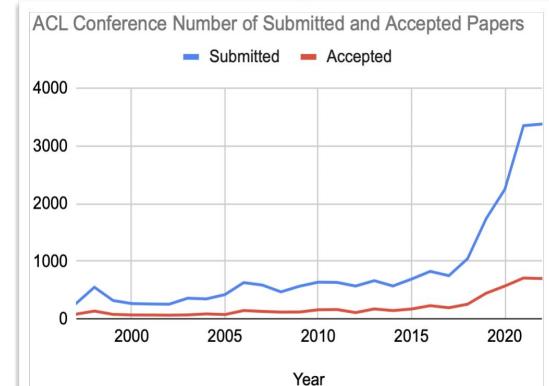
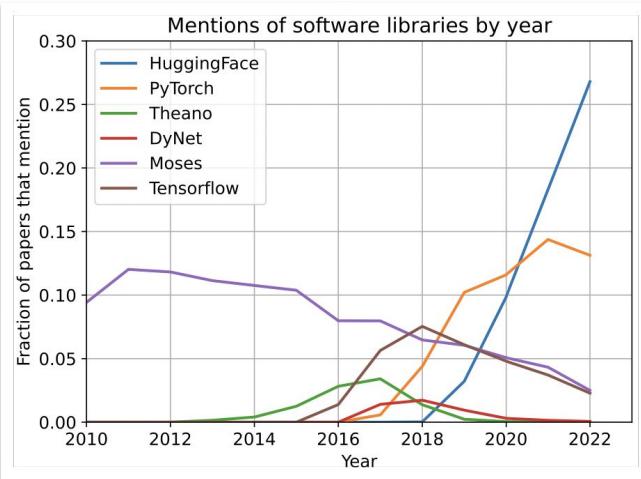
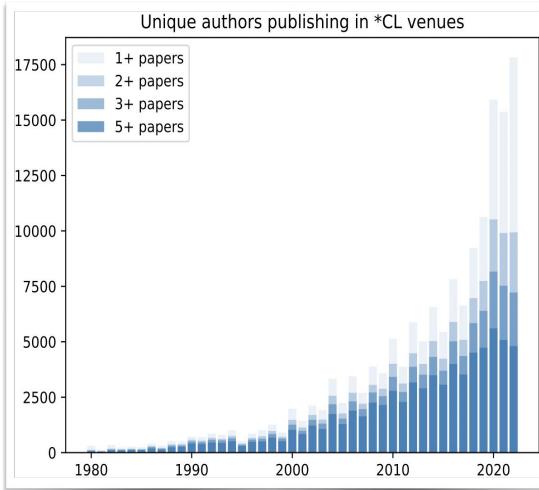


Recently moving  
towards integrated,  
multimodal  
end-to-end models

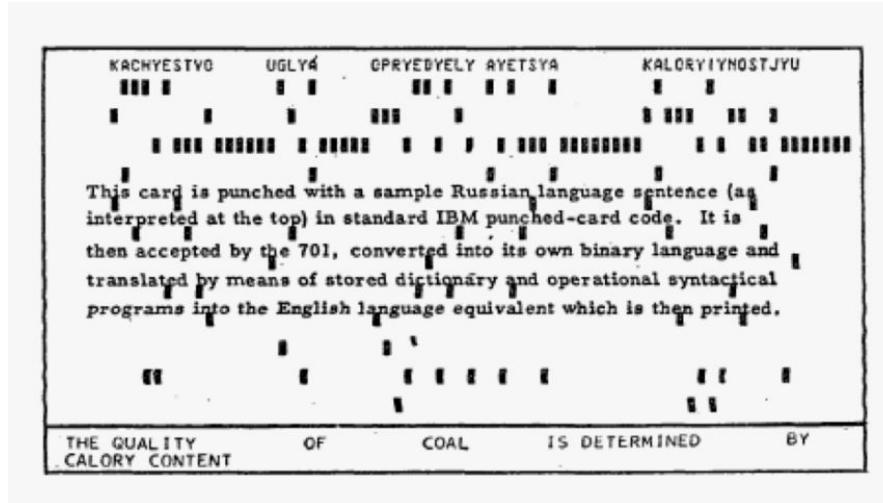
# What is scientific research?

- General goal: Pushing the limits of our **knowledge**
- **Incrementally!** Find a limit/caveat in existing method and solve it!
- For example: lack of parallelization in Recurrent Neural Networks → **Transformers** (Vaswani et al. 2017)
- "Vaswani et al. 2017": a single publication that was **submitted** to a conference, **reviewed** by scientists, then **reproduced**: **research != science**
- Most of methods in this class were published less than 10 years ago

# The shape of today's NLP research



# NLP applications: Machine Translation



Georgetown–IBM experiment 1954

- Machine Translation is the first NLP application
- Google Translate supports 243 languages

Cloud Translation		Model	Method	Usage	Price per unit
Pricing	Cloud Translation pricing	NMT	Text translations, which includes:	First 500,000 characters per month	Free (applied as \$10 credit every month) <sup>1</sup>
	Pricing examples		• Language detection	Over 500,000 characters per month	\$20 per million characters <sup>2</sup>
	Charged characters		• Text translation		
	Charged projects		• Batch text translation		
	Other Google Cloud costs		• XLSX document translation	Over 1 billion characters per month	
	What's next		• Romanz text		
			Document translation (DOCX, PPT, and PDF formats only)	Pages sent to the API per month	\$0.08 per page <sup>3</sup>

# NLP applications: Machine Translation

Ubiquitous on the web and social media

The screenshot shows the homepage of The New York Times. At the top right, there is a language selection bar with "anglais" and "français" buttons. Below the main navigation bar, there are several news articles. One article on the left is titled "Le père d'un adolescent suspecté d'une fusillade dans une école de Géorgie". Another article on the right is titled "Certsains aliments ultra-transformés sont-ils pires que d'autres ?". The interface includes a search bar, a news ticker at the bottom, and a "Google Translate" button.



Desmond Elliott  
@delliott

Fun new paper led by [@IngoZiegler](#) and [@akoksal\\_](#) that shows how we can use retrieval augmentation to create high-quality supervised fine tuning data. All you need to do is write a few examples that demonstrate the task.

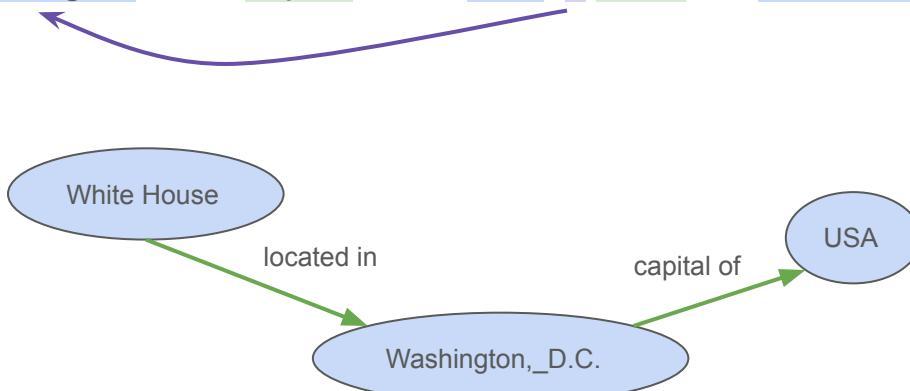
[A l'origine en anglais et traduit par Google](#)

Un nouvel article intéressant dirigé par [@IngoZiegler](#) et [@akoksal\\_](#) qui montre comment nous pouvons utiliser l'augmentation de la récupération pour créer des données de réglage fin supervisées de haute qualité. Tout ce que vous avez à faire est d'écrire quelques exemples qui illustrent la tâche.

# NLP applications: Information Extraction

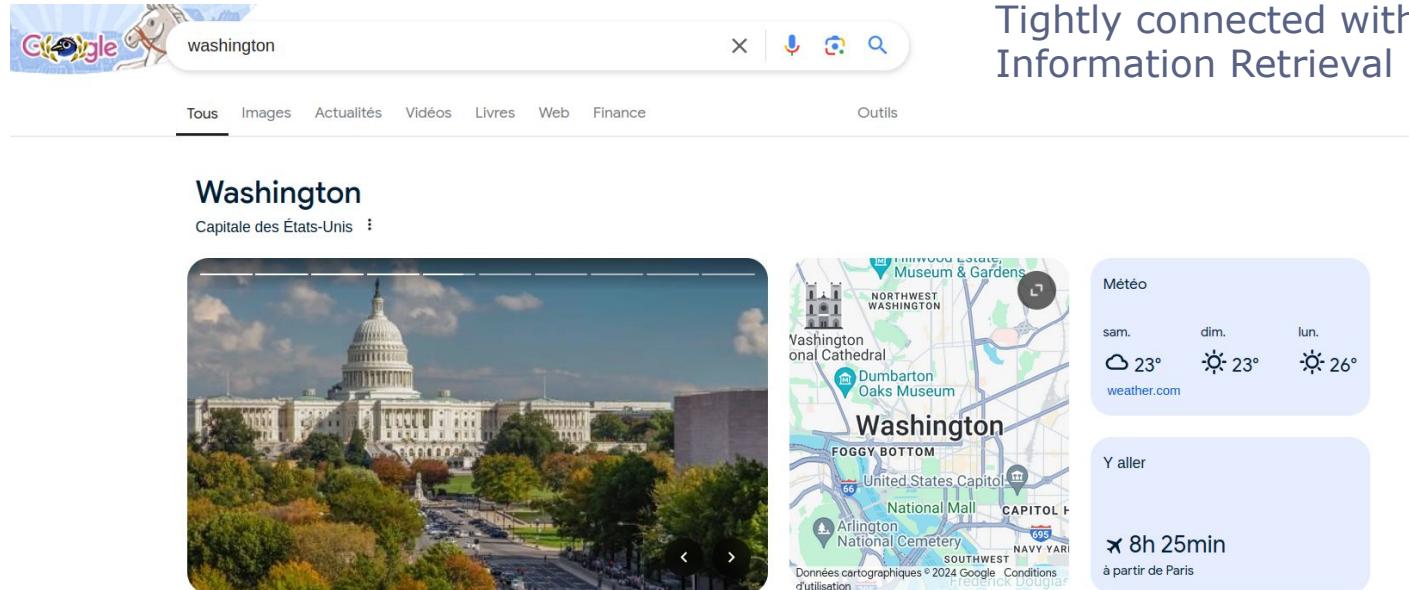
Washington, D.C. != George Washington

Washington is the capital of the USA. It hosts the White House.



- From unstructured text to knowledge graphs
- Named Entity Recognition
- Named Entity Disambiguation
- Coreference resolution
- Relation Extraction

# NLP applications: Information Extraction



A screenshot of a Google search results page for the query "washington". The search bar at the top shows the query "washington". Below the search bar, there are tabs for "Tous", "Images", "Actualités", "Vidéos", "Livres", "Web", and "Finance". On the right side of the search bar, there are icons for microphone, camera, and search. The main content area starts with a section titled "Washington" with the subtitle "Capitale des États-Unis". Below this is a large image of the U.S. Capitol building. To the right of the image is a map of Washington D.C. showing various landmarks like the National Mall, United States Capitol, and Foggy Bottom. Further to the right are two boxes: one for weather (Météo) and one for travel (Y aller).

Tightly connected with  
Information Retrieval

# NLP applications: Information Extraction

Not only for advancing human knowledge

## LOAN AGREEMENT

This **LOAN AGREEMENT**, dated as of November 17, 2014 (this “Agreement”), is made by and among Auxilium Pharmaceuticals, Inc., a corporation incorporated under the laws of the State of Delaware (“U.S. Borrower”), Auxilium UK LTD, a private company limited by shares registered in England and Wales (“UK Borrower”), and Endo Pharmaceuticals Inc., a company (“Lender”).

Page	
<b>SHAREHOLDER INFORMATION</b>	
THE WALT DISNEY CO.	
10QCS CORPORATE ACTION CALENDAR	
LATEST PUBLIC OFFERING	
Date of offering	1/83
Shares offered	1,00M
Share Price	\$ 66.88
Lead Manager	Morgan Stanley
Type	Common Stock
INSIDER TRADING	
Net \$ Value Buys and Sells As Of 01/15/05 (1995 - Present In Dollars)	# of Buyers 773 # of Sellers 288
Lowest activity	12/97 -372,00MLN
Highest activity	08/02 10,17MLN
Mean	-15,95MLN
Most recent 45 days	.00
INSTITUTIONAL OWNERSHIP	
# of Holders	1,970
Shares Held	1,39BLN
% Shares Out.	68,12
Shares Purchased	1,75MLN
Australia: 61 2 9777 8600 Brazil: 5511 3048 4500 Europe: 44 20 7330 7500 Germany: 49 69 990410 Hong Kong: 952 2577 6000 Japan: 81 3 3801 8900 Singapore: 65 6212 1000 U.S.: 1 212 218 2000 Copyright 2003 Bloomberg L.P. 1 24-Jan-05 15:16:57	

dgp Equity HDS						
Enter #<GO> to select aggregate portfolio and see detailed information						
00273178194-0010 HOLDINGS SEARCH CUSIP: 25458710						
DIS	US	THE WALT DISNEY CO.		Page 1 / 100		
Holder name	Portfolio Name	Source	Held	Outstd	Change	Latest Filing Date
BARCLAYS GLOBAL	BARCLAYS BANK PLC	13F	91,394M	4,476M	+7,912M	09/04
CITIGROUP INCORP	CITIGROUP INCORPORAT	13F	71,012M	3,475M	+89,816M	09/04
STATE STREET	STATE STREET CORPORA	13F	69,236M	3,385M	+1,244M	09/04
FIDELITY MANAGEM	FIDELITY MANAGEMENT	13F	67,611M	3,303M	+4,986M	05/04
SOUTHERNTRUST ASST	SOUTHEASTERN ASSET M	13F	52,349M	2,951M	+3,194M	09/04
VANGUARD GROUP	VANGUARD GROUP INC	13F	43,710M	2,139M	+979,055M	12/04
71ST FARM M AUTO	STATE FARM MUTUAL AU	13F	42,234M	2,067M	+10,300M	09/04
MELLON BANK N A	MELLON BANK CORP	13F	39,545M	1,935M	+2,998M	09/04
LORD ARBETT & CO	LORD ARBETT & CO	13F	37,460M	1,833M	+265,434M	12/04
MORGAN STANLEY	MORGAN STANLEY	13F	31,643M	1,545M	+1,348M	09/04
NORTHERN TRUST C	NORTHERN TRUST CORPO	13F	26,061M	1,275M	+39,493M	09/04
DEUTSCHE BANK AG	DEUTSCHE BANK AG	13F	21,990M	1,076M	+1,570M	09/04
DISNEY ROY EDMAR	n/a	Form 4	17,279M	0,846M	+7,726M	08/03
JANUS CAPITAL	JANUS CAPITAL CORPOR	13F	17,156M	0,840M	+2,077M	03/04
BKCAPITAL RSCH MGR	BKCAPITAL RESEARCH AND	13F	17,142M	0,835M	+1,507M	05/04
TUKMAN CAPITAL MNGT	TUKMAN CAPITAL MANAG	13F	16,962M	0,830M	+1,861M	09/04
T ROME PRICE ASSOCIA	T ROME PRICE ASSOCIA	13F	16,810M	0,823M	+1,367M	09/04
Sub-totals for current page:						
* Money market directory info available. Select portfolio, then hit #<GO>.						
Australia: 61 2 9777 8600 Brazil: 5511 3048 4500 Europe: 44 20 7330 7500 Germany: 49 69 990410 Hong Kong: 952 2577 6000 Japan: 81 3 3801 8900 Singapore: 65 6212 1000 U.S.: 1 212 218 2000 Copyright 2003 Bloomberg L.P. 1 24-Jan-05 15:15:32						

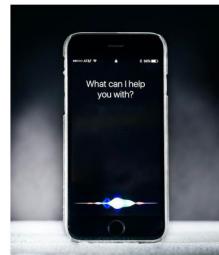
Bloomberg

Paul Lerner – February 2025

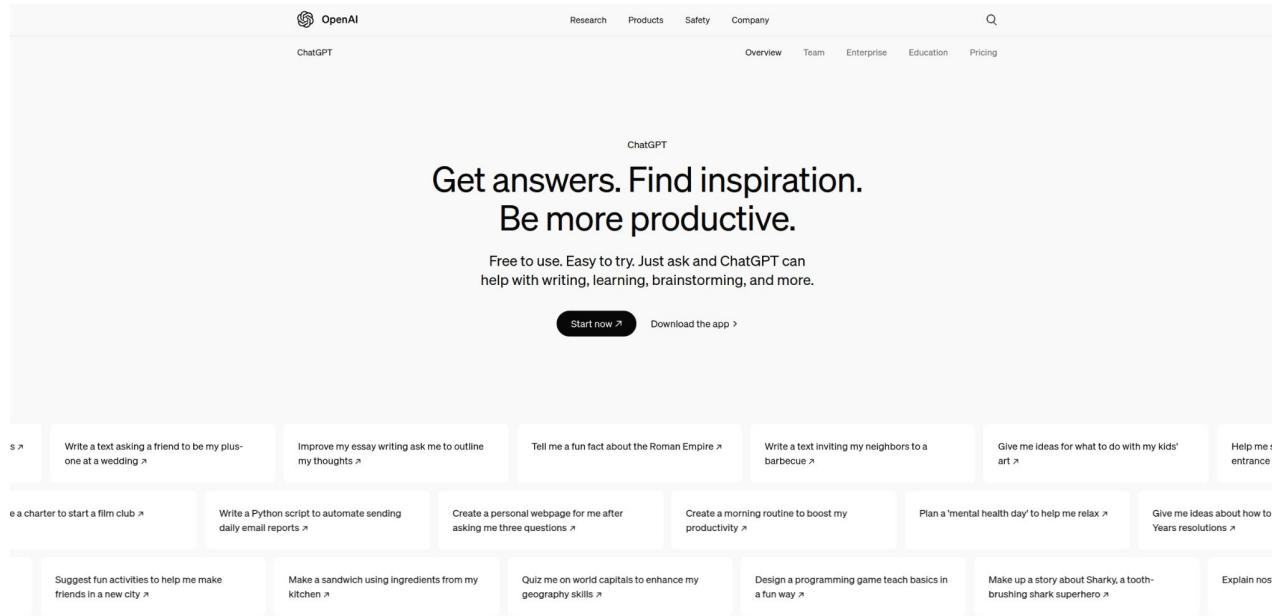
aivancity  
PARIS-CACHAN

# NLP applications: chatbots

- Task-oriented like Siri/Alexa or chitchat like ChatGPT (OpenAI)
- Moving from a complicated pipeline (ASR, Information Extraction, Information Retrieval) to end-to-end language modeling



# NLP applications: chatbots



# NLP applications: chatbots: demo

32 lines of python code

```
1  from transformers import AutoModelForCausalLM, AutoTokenizer, BitsAndBytesConfig
2
3  def pipeline(text, model, tokenizer):
4      messages = [
5          {"role": "user", "content": text}
6      ]
7      inputs = tokenizer.apply_chat_template(messages, return_dict=True, tokenize=True, return_tensors="pt", add_generation_prompt=True)
8      for k, v in inputs.items():
9          inputs[k] = v.cuda()
10     input_length = inputs["input_ids"].shape[1]
11     output = model.generate(**inputs, max_new_tokens=128)
12     output = tokenizer.batch_decode(output[:, input_length:], skip_special_tokens=True, clean_up_tokenization_spaces=True)
13     return output
14
15
16 def user_loop(model, tokenizer):
17     while True:
18         answer = input("=> ").strip()
19         output = pipeline(answer, model, tokenizer)[0]
20         print(f"{output}\n")
21
22
23 def main():
24     model_name = "google/gemma-2-2b-bit"
25     model = AutoModelForCausalLM.from_pretrained(
26         model_name,
27         device_map="auto",
28         token=True,
29         quantization_config=BitsAndBytesConfig(load_in_4bit=True)
30     )
31     tokenizer = AutoTokenizer.from_pretrained(model_name)
32     user_loop(model, tokenizer)
```

# NLP applications: chatbots: demo

```
>>> what is distributional semantics?  
## Distributional Semantics: Understanding Words in Context
```

Distributional semantics is a branch of natural language processing (NLP) that focuses on the \*\*relationships between words and their meaning based on their co-occurrence in a large corpus of text\*\*.

Think of it like this: instead of focusing on individual words and their exact definitions, distributional semantics analyzes how words are used in context.

\*\*Here's a breakdown:\*\*

- \* \*\*Words are not isolated:\*\* They are part of a larger network of words and phrases.
- \* \*\*Meaning is derived from context:\*\* The meaning of a word is determined by how it is

# NLP applications: chatbots: demo

>>> I'm teaching an NLP class at the aivancity AI school, any advices?

That's awesome! Teaching NLP is a fantastic way to share the power of AI with the next generation. Here's some advice to make your class engaging and impactful:

\*\*1. Know Your Audience:\*\*

\* \*\*Beginner or Advanced?\*\* Are you targeting absolute beginners or those with some prior knowledge? Tailor your content and pace accordingly.

\* \*\*AI Enthusiasts vs. General Interest:\*\* Are you aiming for a more technical audience or those with a broader interest in AI?

\* \*\*Learning Style:\*\* Consider how your students learn best (visual, auditory, kinesthetic). Incorporate diverse teaching methods.

# NLP applications: chatbots: demo

```
>>> you're speaking live to students of the aivancity AI school, introduce yourself  
Hello everyone! 🙋 I'm Gemma, an AI assistant here to help you learn and explore the fascinating world of AI.
```

I'm excited to be speaking with you today at Aivancity AI School. I've been trained on a massive dataset of text and code, which allows me to understand and generate human-like text.

What are you working on today? Do you have any questions about AI concepts, or are you working on a specific project? I'm here to help!

Let's dive into the world of AI together! 🤖

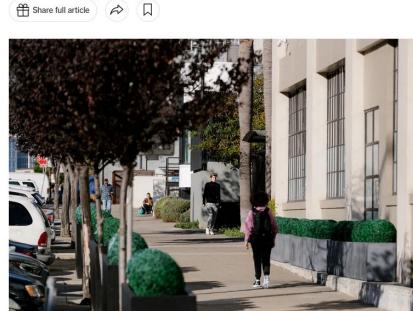
# Big Money in LLMs/Chatbots

The New York Times

OpenAI Tries to Grow Up   Changing a Chatbot's Mind   A.I.'s Threat to Itself   'Deepfake Elon Musk'

## OpenAI Completes Deal That Values the Company at \$80 Billion

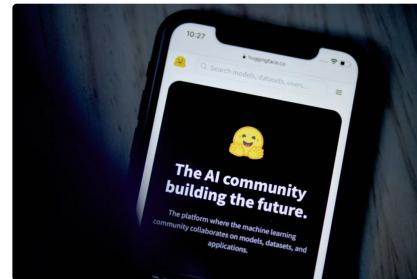
The A.I. start-up's valuation tripled in less than 10 months.



Outside OpenAI's offices in San Francisco. The company's latest deal is another example of the Silicon Valley deal-making machine pumping money into a handful of companies that specialize in generative A.I. Jason Henry for The New York Times

Bloomberg

## AI Startup Hugging Face Valued at \$4.5 Billion After Raising Funding From Google, Nvidia



AI Startup Hugging Face Valued at \$4.5 Billion After Raising Funding From Google, Nvidia - Bloomberg

The New York Times

Artificial Intelligence   OpenAI Tries to Grow Up   Changing a Chatbot's Mind   A.I.'s Threat to Itself   'Deepfake Elon Musk'   Q&A: Fake or Real Images?

## Mistral, a French A.I. Start-Up, Is Valued at \$6.2 Billion

Created by alumni from Meta and Google, Mistral is just a year old and has already raised more than \$1 billion in total from investors, leading to eye-popping valuations.

▶ Listen to this article · 2:29 min · Listen now

Share full article



# Big Money in LLMs/Chatbots

## Zuckerberg's Meta Is Spending Billions to Buy 350,000 Nvidia H100 GPUs

In total, Meta will have the compute power equivalent to 600,000 Nvidia H100 GPUs to help it develop next-generation AI, says CEO Mark Zuckerberg.



By Michael Kan January 18, 2024

f X g ...



(David Paul Morris/Bloomberg via Getty Images)

Paul Lerner – February 2025



# Big Money in LLMs/Chatbots

## Le Monde

ÉCONOMIE • PIXELS

Emmanuel Macron annonce 500 millions d'euros supplémentaires pour développer l'intelligence artificielle en France

« Nous devons faire émerger cinq à dix clusters pour avoir deux ou trois pôles d'excellence » au niveau mondial a déclaré le chef de l'Etat en visite au salon VivaTech.



Paul Lerner – February 2025

# Jean Zay cluster

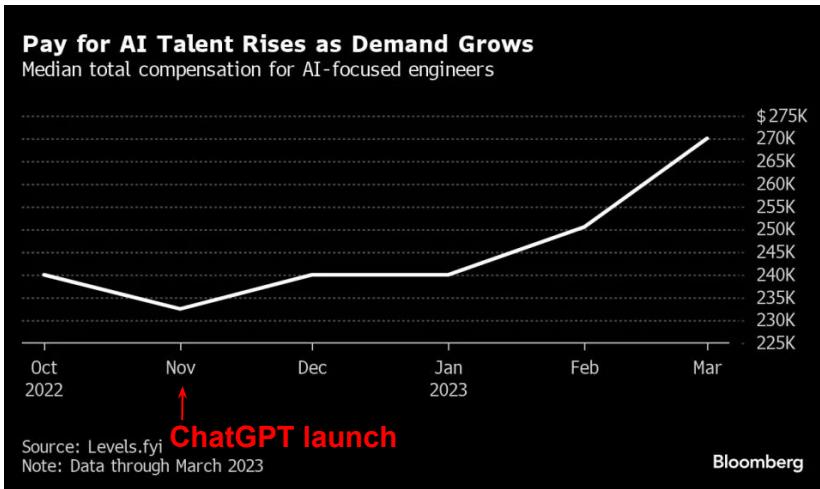
```
(matos) [us147fp@jean-zay: experiments]$ sinfo
PARTITION_AVAIL TIMELEFT NODES STATE NODESELECT
cpu_p1*    up 4:04:00:00      2 drain$ r3l0n[0-9]
cpu_p1*    up 4:04:00:00      1 maint r2l1n16
cpu_p1*    up 4:04:00:00      1 drain* r2l7n9
cpu_p1*    up 4:04:00:00      4 mix r1l3n12,r3l0n[4,25,34]
cpu_p1*    up 4:04:00:00      711 alloc r1l0n[0-35],r1l1n[0-35],r1l2n[0-35],r1l3n[0-11,13-35],r1l4n[0-35],r1l5n[0-35],r1l6n[0-35],r1l7n[0-35],r2l0n[0-35],r2l1n[0-5,17-35],r2l2n[0-35],r2l3n[0-35],r2l4n[0-35],r2l5n[0-35],r2l6n[0-8,19-35],r3l0n[0-35],r3l1n[0-35],r3l2n[0-30,32-35],r3l3n[0-35],r3l4n[0-7,10-35]
cpu_p1*    up 4:04:00:00      1 tde r3l3n1
gpu_p1*    up 4:04:00:00      1 drain* r2l1n16
gpu_p1*    up 4:04:00:00      1 drain* r2l7n9
gpu_p1*    up 4:04:00:00      1 drain* r3l0n9
gpu_p1*    up 4:04:00:00      99 mix r3l4n[1-2],r3l5n[0-4],r3l6n[2,4,6-7],r3l7n8,r6l0n7,r6l1n[2,4,6],r6l2n[0,2-5],r6l3n[1,6,8],r6l4n[0-1,3-5,7-8],r6l5n[1,3,5,8],r6l6n[0-7],r6l7n7,r7l0n[2-4],r7l1n5,r7l2n[1-4,5]-7l3n7,r7l4n[0-5,7-8],r7l5n3,r7l6n[2,4,7-8],r7l7n0,r8l0n1,r8l1n2-3],r8l3n6,r8l4n8,r8l5n[3,8],r8l7n[1,5,7],r9l0n4,r9l1n[0,2,7],r9l2n[0,5,7],r9l3n[2,4],r9l4n[1,5,8],r9l5n[0,8],r9l7n[0-4,5]-10l0n6,r10l1n5,r10l2n[0,2,5],r10l3n0,r10l5n[3-4],r10l6n[4,7-8],r10l7n[0,4-6]
gpu_p1*    up 4:04:00:00      292 alloc r3l4n[0-3,7],r3l5n[5-8],r3l6n[0-1,3,5,8],r3l7n[0-7],r6l0n[1-6,8],r6l1n[0-4,5]-7l2n[1-6,8],r6l3n[0-2,4-8],r6l4n[0-2,3-6,8],r6l5n[0-6,8]-7l4n[0-7,7],r7l0n[0-2,4-8],r7l1n[0-1,3-5,7-8]-7l7n[0-7,7],r8l0n[0-8],r8l1n[0-2,4-8]-8l1n[0-1,4-8],r8l3n[0-5,7-8],r8l4n[0-7],r8l5n[0-1,4,6-7],r8l6n[0-9]-8l7n[0-2,4-6,8]-r8l8n[0-3,5-8],r9l1n[1,3-6,8],r9l2n[1-4,6,8],r9l3n[0-1,3-5,8],r9l4n[0-2,4-6,8]-r9l5n[0-5-7],r9l6n[0-8],r9l7n[1-3,6-8],r10l0n[0-5,7-8],r10l1n[0-4,6-8],r10l2n[1,3-4,6-8],r10l3n[1-8],r10l4n[0-8],r10l5n[0-2,5-8],r10l6n[0-3,5-6],r10l7n[0-1,3-7,8]
gpu_p2*   up 4:04:00:00      1 drain$ jean-zay-la810
gpu_p2*   up 4:04:00:00      17 mix jean-zay-la[802,805-807,809,812,815-816,820,822,824-827,829-831]
gpu_p2*   up 4:04:00:00      12 alloc jean-zay-la[801,803-804,811,813-814,817-819,821,823,828]
gpu_p2*   up 4:04:00:00      1 tde jean-zay-la808
gpu_p2*   up 4:04:00:00      1 drain$ jean-zay-la808
gpu_p2*   up 4:04:00:00      5 mix jean-zay-la[802,805-807,809]
gpu_p2*   up 4:04:00:00      4 alloc jean-zay-la[801,803-804,811]
gpu_p2*   up 4:04:00:00      1 idle jean-zay-la808
gpu_p2*   up 4:04:00:00      12 mix jean-zay-la[812,815-816,820,822,824-827,829-831]
gpu_p2*   up 4:04:00:00      8 alloc jean-zay-la[813-814,817-819,821,823,828]
gpu_p5*   up 4:04:00:00      38 mix jean-zay-lam[01,03-13,15-18,20,22-26,28-29,33,35-36,38-42,44,46-48,51-52]
gpu_p5*   up 4:04:00:00      14 alloc jean-zay-lam[02,14,19,21,27,30-32,34,37,43,45,49-50]
vtsu*    up 4:00:00      1 drain* jean-zay-vtsu
vtcu*    up 4:00:00      1 rest$ jean-zay-vtsu
vtcu*    up 4:00:00      1 mix Jean-Zay-vtsu1
vtsu*    up 4:00:00      2 idle Jean-Zay-vtsu[2-3]
prepost* up 20:00:00      1 drain* jean-zay-pp1
prepost* up 20:00:00      3 mix Jean-Zay-pp[2-4]
archive* up 20:00:00      3 comp idrsrv[06-08]
archive* up 20:00:00      1 resv idrsrv[06-08]
compil* up 20:00:00      1 drain* jean-zay-pp1
compil* up 20:00:00      3 comp idrsrv[06-08]
compil* up 20:00:00      1 resv idrsrv[05]
compil* up 20:00:00      3 mix Jean-Zay-pp[2-4]
```

```
(matos) bash-5.1$ nvidia-smi
Fri Sep 6 12:04:10 2024
```

		Driver Version: 550.54.15		CUDA Version: 12.4	
GPU Name	Persistence-M	Bus-Id	Disp.A	Volatile	Uncorr. ECC
Fan	Temp	Perf	Pwr:Usage/Cap	Memory-Usage	GPU-Util Compute M.
0 Tesla V100-SXM2-32GB	On	00000000:1A:00.0	Off	0	
N/A 44C P0	45W / 300W	0MiB / 32768MiB	0%	Default	N/A



# Big Money in LLMs/Chatbots



Nvidia

107,21 \$ ↑ 2 298,43 % +102,74 5 a

Avant l'ouverture : 104,90 \$ (↓ 2,15 %) -2,31

Fermé : 6 sept., 05:35:32 UTC-4 · USD · NASDAQ · Clause de non-responsabilité



+568% in less than 2 years

# Break for questions and "appel"

# What is a word?

- Open question in phonology vs. morphology
- Inflection: is brother != brothers?
- Compounding: is motorbike == motor + bike?
- Multi-word expressions: mother in law == 1 or 3 words?
- Polysemy: is chair (furniture) != chair (person)?
- Orthography: is modeling != modelling?

# NLP deals with orthographic words...

- "My brother is sitting on a chair" →  
['My', 'brother', 'is', 'sitting', 'on', 'a', 'chair']  
**(tokenization:** sequence of *tokens*)
- Inflection (brother vs brothers): usually not modeled
- Compounding (motorbike vs motor + bike): usually not modeled
- Multi-word expressions (mother in law): usually not modeled
- Polysemy (chair [furniture] vs chair [person]): usually modeled after sharing an initial representation
- **Orthography:** 'modeling' != 'mode **ll**ing'

# ...except when it's the research topic!

Inflection (brother vs brothers): "brother" is a *lemma* (singular, masc. form): useful for **indexing** (keyword-like) in Information Retrieval

## Indicatif

### Présent

je mange  
tu manges  
il mange  
nous mangeons  
vous mangez  
ils mangent

### Passé simple

je mangeai  
tu mangeas  
il mangea  
nous mangeâmes  
vous mangeâtes  
ils mangèrent

### Passé composé

j'ai mangé  
tu as mangé  
il a mangé  
nous avons mangé  
vous avez mangé  
ils ont mangé

### Passé antérieur

j'eus mangé  
tu eus mangé  
il eut mangé  
nous eûmes mangé  
vous eûtes mangé  
ils eurent mangé

### Imparfait

je mangeais  
tu mangeais  
il mangeait  
nous mangions  
vous mangiez  
ils mangeaient

### Futur simple

je mangeraï  
tu mangeras  
il mangera  
nous mangerons  
vous mangerez  
ils mangeron

### Plus-que-parfait

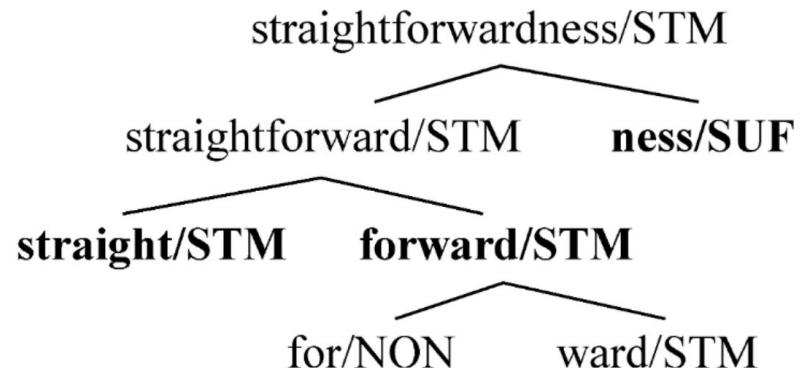
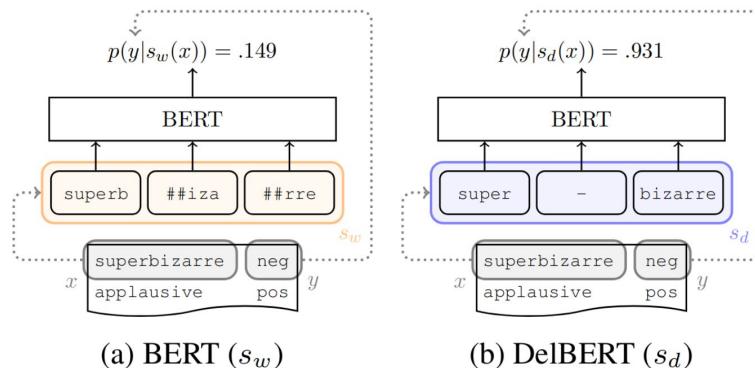
j'avais mangé  
tu avais mangé  
il avait mangé  
nous avions mangé  
vous aviez mangé  
ils avaient mangé

### Futur antérieur

j'aurai mangé  
tu auras mangé  
il aura mangé  
nous aurons mangé  
vous aurez mangé  
ils auront mangé

# ...except when it's the research topic!

Compounding (motorbike vs motor + bike): very niche but studied in computational linguistics



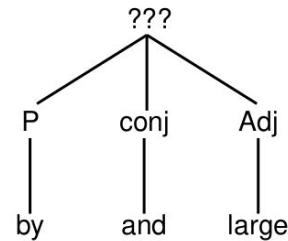
# ...except when it's the research topic!

Multi-word expressions ("mother in law")

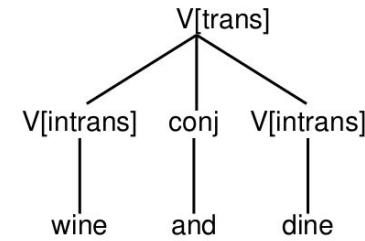
Pointwise mutual information (PMI):

$$\log \frac{P(x,y)}{P(x)P(y)}$$

*by and large*



*wine and dine*



## ...except when it's the research topic!

Orthography ('modeling' ← 'mode **ll**ing') for **User-Generated Content**

*i left ACL    cus    im sickk ! Yuu better be their    tmrw    . GN 4now*

↓              ↓              ↓              ↓              ↓

*i left ACL because I'm sick ! You better be their tomorrow . GN 4now*

# What do words mean?

- Why is it "brother" in English and "frère" in French?
- Because "brōþēr" in Proto-Germanic and "frātrem" in Latin!  
(arbitrariness of the sign, de Saussure, 1916)  
But why does it *mean* brother?
- The meaning of a word is its **use** in the language (Ludwig Wittgenstein, 1921):  
"I was playing with my **brother** and *sister*"  
"My *mom* is feeding my **brother**"
- "brother" co-occurs with "mom" and "sister"  
like "frère" co-occurs with "maman" and "sœur"
- Polysemy: "I sit on a *chair*" vs "He is the *chair* of this session"

# How words are used?

- words are defined by their environments (the words around them)
- If A and B have almost identical environments we say that they are **synonyms** (Harris, 1954).
- define the meaning of a word by its distribution in language use: its neighboring words

# What does "ongchoi" mean?

- Suppose you see these sentences:
  - *Ongchoi* is delicious **sautéed with garlic**.
  - *Ongchoi* is superb over **rice**
  - *Ongchoi leaves* with **salty** sauces
- And you've also seen these:
  - ...*spinach* **sautéed with garlic** over **rice**
  - *Chard* stems and **leaves** are **delicious**
  - *Collard greens* and other **salty** leafy greens
- *Ongchoi* is a leafy green like *spinach*, *chard*, or *collard greens*



# Defining context (word-word matrix)

Two words are similar in meaning if their context vectors are similar

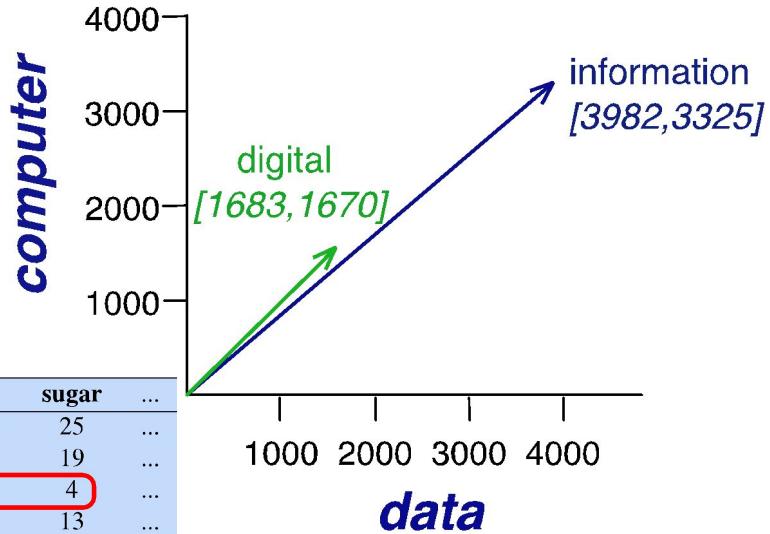
is traditionally followed by **cherry** pie, a traditional dessert  
 often mixed, such as **strawberry** rhubarb pie. Apple pie  
 computer peripherals and personal **digital** assistants. These devices usually  
 a computer. This includes **information** available on the internet

	aardvark	...	computer	data	result	pie	sugar	...
cherry	0	...	2	8	9	442	25	...
strawberry	0	...	0	0	1	60	19	...
digital	0	...	1670	1683	85	5	4	...
information	0	...	3325	3982	378	5	13	...

# Defining context (word-word matrix)

Two words are similar in meaning if their context vectors are similar

	aardvark	...	<b>computer</b>	data	result	pie	sugar	...
cherry	0	...	2	8	9	442	25	...
strawberry	0	...	0	0	1	60	19	...
digital	0	...	1670	1683	85	5	4	...
information	0	...	3325	3982	378	5	13	...



# Computing word similarity: Dot product

The dot product between two vectors is a scalar:

$$\text{dot product}(\mathbf{v}, \mathbf{w}) = \mathbf{v} \cdot \mathbf{w} = \sum_{i=1}^N v_i w_i = v_1 w_1 + v_2 w_2 + \dots + v_N w_N$$

The dot product tends to be high when the two vectors have large values in the same dimensions

Dot product can thus be a useful similarity metric between vectors

# Problem with raw dot-product

Dot product favors long vectors

Dot product is higher if a vector is longer (has higher values in many dimension)

Vector length (euclidean norm):

$$|\mathbf{v}| = \sqrt{\sum_{i=1}^N v_i^2}$$

Frequent words (of, the, you) have long vectors (since they occur many times with other words).

So dot product overly favors frequent words

# Alternative: cosine for word similarity

$$\text{cosine}(\vec{v}, \vec{w}) = \frac{\vec{v} \cdot \vec{w}}{|\vec{v}| |\vec{w}|} = \frac{\sum_{i=1}^N v_i w_i}{\sqrt{\sum_{i=1}^N v_i^2} \sqrt{\sum_{i=1}^N w_i^2}}$$

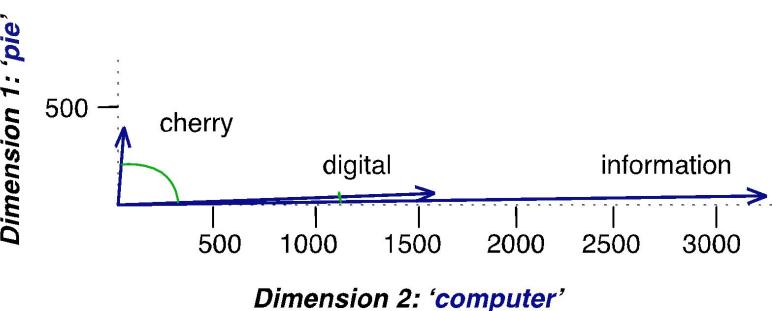
Based on the definition of the dot product between two vectors  $\mathbf{a}$  and  $\mathbf{b}$

$$\mathbf{a} \cdot \mathbf{b} = |\mathbf{a}| |\mathbf{b}| \cos \theta$$

$$\frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{a}| |\mathbf{b}|} = \cos \theta$$

# Cosine examples

	pie	data	computer
cherry	442	8	2
digital	5	1683	1670
information	5	3982	3325



$$\cos(\text{cherry}, \text{information}) = \frac{442 * 5 + 8 * 3982 + 2 * 3325}{\sqrt{442^2 + 8^2 + 2^2} \sqrt{5^2 + 3982^2 + 3325^2}} = .017$$

$$\cos(\text{digital}, \text{information}) = \frac{5 * 5 + 1683 * 3982 + 1670 * 3325}{\sqrt{5^2 + 1683^2 + 1670^2} \sqrt{5^2 + 3982^2 + 3325^2}} = .996$$

# Can we compute word similarity like this?

	$V$ vocabulary size							
	aardvark	...	computer	data	result	pie	sugar	...
cherry	0	...	2	8	9	442	25	...
strawberry	0	...	0	0	1	60	19	...

- **Sparse** vectors (most words never co-occur together)
- Very **high dimension!**  $V$ : vocabulary size (usually 20,000 - 200,000)

# How do we reduce dimensionality?

from  $V$  (vocabulary size) to  $d \ll V$

	aardvark	...	computer	data	result	pie	sugar	...
cherry	0	...	2	8	9	442	25	...
strawberry	0	...	0	0	1	60	19	...

- Generic solutions:
  - Principal Component Analysis (PCA)
  - Singular Value Decomposition (SVD) → Latent Semantic Indexing/Analysis (Deerwester et al., 1990)
- Deep learning solution: Skipgram (word2vec, Mikolov 2013)
- Output = **embedding** = **dense** vector of dimension  $d$ 
  - reflects semantic similarity
  - can be used as features by any machine learning model

# Latent Semantic Indexing/Analysis

$$\underset{V \times C}{\mathbf{A}} \approx \hat{\mathbf{A}} = \underset{V \times d}{\mathbf{M}} \times \text{diag}(\mathbf{s}) \times \underset{d \times C}{\mathbf{C}^T}$$

Singular Value Decomposition (SVD)

- Usually done with word-document occurrences instead of word-word
- Actually Pointwise Mutual Information instead of raw counting
- Closely related to Skipgram (Levy and Goldberg, 2014)

SVD:

truncated at  $d$ :

# Skipgram (word2vec, Mikolov)

- Instead of **counting** how often each word  $w$  occurs near "apricot"  
Train a classifier on a binary prediction task: Is  $w$  likely to show up near "apricot"?
- We don't actually care about this task  
But we'll take the learned classifier weights as the word embeddings
- Big idea: **self-supervision**:
  - A word  $c$  that occurs near *apricot* in the corpus acts as the gold "correct answer" for supervised learning
  - No need for human labels

# Skipgram (word2vec, Mikolov)

- Treat the target word  $w$  and a neighboring context word  $c$  as positive examples.
- Randomly sample other words in the lexicon to get negative examples
- Use logistic regression to train a classifier to distinguish those two cases
- Use the learned weights as the embeddings

# Skipgram (word2vec, Mikolov)

Assume a +/- 2 word window, given training sentence:

...lemon, a tablespoon of apricot jam, a pinch...

Goal: train a classifier that is given a candidate (word, context) pair

(apricot, jam)

(apricot, aardvark)

And assigns each pair a probability:

$$P(+|w, c)$$

$$P(-|w, c) = 1 - P(+|w, c)$$

# Turning dot products into probabilities

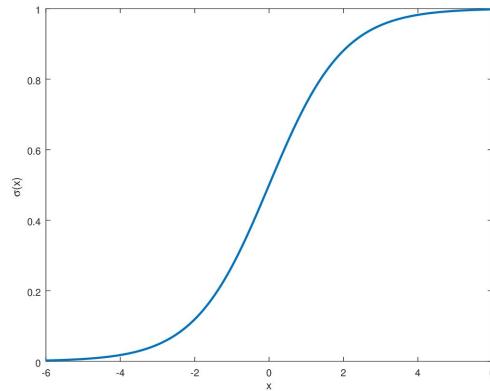
$$\text{Sim}(w, c) \approx w \cdot c$$

To turn this into a probability

We'll use the sigmoid from logistic regression:

$$P(+|w, c) = \sigma(c \cdot w) = \frac{1}{1 + \exp(-c \cdot w)}$$

$$\begin{aligned} P(-|w, c) &= 1 - P(+|w, c) \\ &= \sigma(-c \cdot w) = \frac{1}{1 + \exp(c \cdot w)} \end{aligned}$$



# From 1 context word to full context

$$P(+|w, c) = \sigma(c \cdot w) = \frac{1}{1 + \exp(-c \cdot w)}$$

**Assume** all context words are **independent** → joint probability = product

$$P(+|w, c_{1:L}) = \prod_{i=1}^L \sigma(c_i \cdot w)$$

$$\log P(+|w, c_{1:L}) = \sum_{i=1}^L \log \sigma(c_i \cdot w)$$

log Prob: **systematic** trick for **numerical stability**

# Skip-Gram Training data

...lemon, a tablespoon of apricot jam, a pinch...

## positive examples +

t	c
apricot	tablespoon
apricot	of
apricot	jam
apricot	a

## negative examples -

t	c	t	c
apricot	aardvark	apricot	seven
apricot	my	apricot	forever
apricot	where	apricot	dear
apricot	coaxial	apricot	if

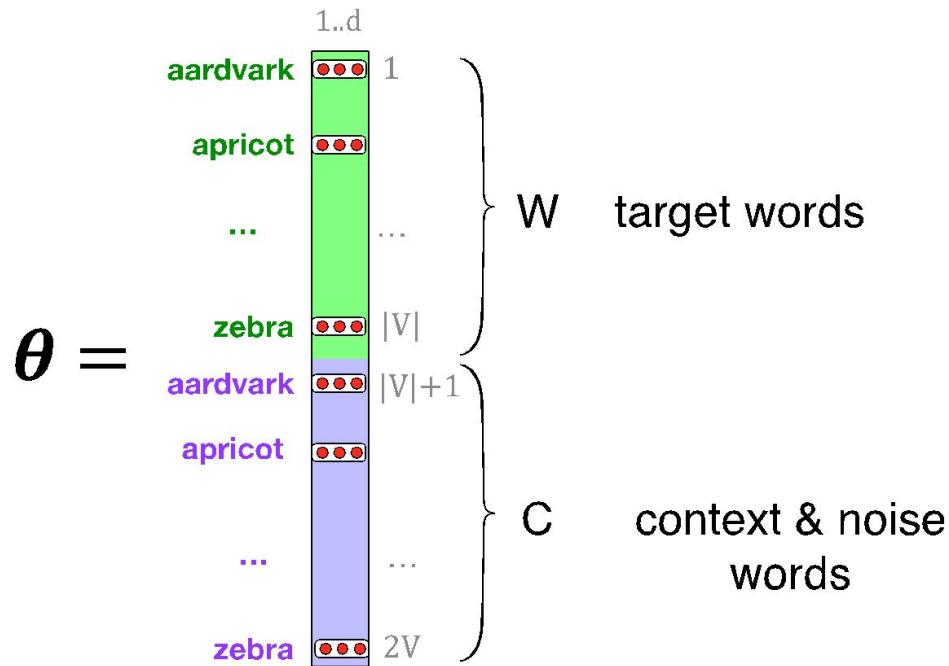
- Maximize the similarity of the target word, context word pairs ( $w, c+$ ) drawn from the positive data
- Minimize the similarity of the ( $w, c-$ ) pairs drawn from the negative data.

# Loss function for one $w$

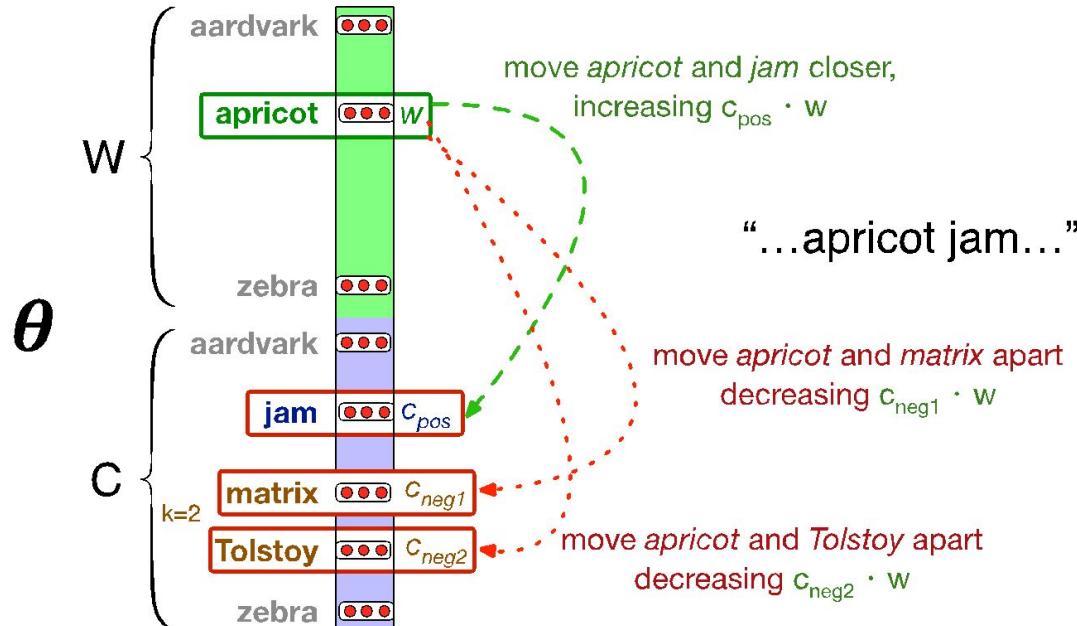
- Maximize the similarity of the target word, context word pairs ( $w, c_+$ ) drawn from the positive data
- Minimize the similarity of the ( $w, c_-$ ) pairs drawn from the negative data.

$$\begin{aligned}
 L_{CE} &= -\log \left[ P(+|w, c_{pos}) \prod_{i=1}^k P(-|w, c_{neg_i}) \right] \\
 &= - \left[ \log P(+|w, c_{pos}) + \sum_{i=1}^k \log P(-|w, c_{neg_i}) \right] \\
 &= - \left[ \log P(+|w, c_{pos}) + \sum_{i=1}^k \log (1 - P(+|w, c_{neg_i})) \right] \\
 &= - \left[ \log \sigma(c_{pos} \cdot w) + \sum_{i=1}^k \log \sigma(-c_{neg_i} \cdot w) \right]
 \end{aligned}$$

# Learning with Stochastic gradient descent

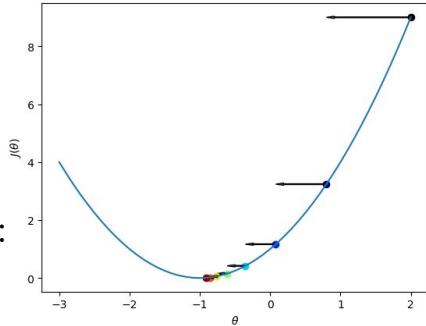


# Learning with Stochastic gradient descent



# Stochastic gradient descent (SGD) reminder

- Learning rate  $\alpha \in \mathbb{R}, \alpha > 0$
- Randomly initialize  $\theta^{(0)}$
- Iteratively get better estimate with:



Next estimate      Learning rate (step size)

$$\theta^{(i+1)} = \theta^{(i)} - \alpha * \boxed{\frac{\partial L}{\partial \theta}(\theta^{(i)})}$$

Previous Estimate

**Gradient is:**

- the vector of partial derivatives of the parameters with respect to the loss function
- A linear approximation of the loss function at  $\theta^{(i)}$

$$\frac{\partial L}{\partial \theta}(\theta^{(i)}) = \begin{bmatrix} \frac{\partial L}{\partial \theta_1^{(i)}} \\ \frac{\partial L}{\partial \theta_2^{(i)}} \\ \vdots \\ \frac{\partial L}{\partial \theta_n^{(i)}} \end{bmatrix}$$

# The derivatives of the loss function

$$L_{CE} = - \left[ \log \sigma(c_{pos} \cdot w) + \sum_{i=1}^k \log \sigma(-c_{neg_i} \cdot w) \right]$$

$$\frac{\partial L_{CE}}{\partial c_{pos}} = [\sigma(c_{pos} \cdot w) - 1]w$$

$$\frac{\partial L_{CE}}{\partial c_{neg}} = [\sigma(c_{neg} \cdot w)]w$$

$$\frac{\partial L_{CE}}{\partial w} = [\sigma(c_{pos} \cdot w) - 1]c_{pos} + \sum_{i=1}^k [\sigma(c_{neg_i} \cdot w)]c_{neg_i}$$

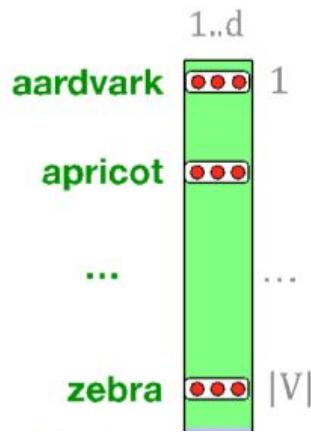
# Stochastic gradient descent update

$$c_{pos}^{t+1} = c_{pos}^t - \eta [\sigma(c_{pos}^t \cdot w^t) - 1] w^t$$

$$c_{neg}^{t+1} = c_{neg}^t - \eta [\sigma(c_{neg}^t \cdot w^t)] w^t$$

$$w^{t+1} = w^t - \eta \left[ [\sigma(c_{pos} \cdot w^t) - 1] c_{pos} + \sum_{i=1}^k [\sigma(c_{neg_i} \cdot w^t)] c_{neg_i} \right]$$

# Embedding = lookup table or linear layer?



One-hot encoding

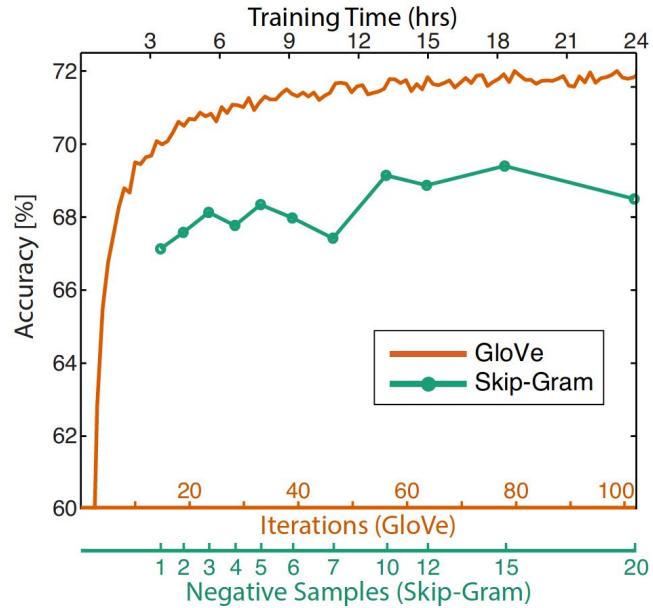
$$\text{Standard basis of } \mathbb{R}^n : e_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, e_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \dots, e_n = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

$|\mathcal{V}| = n:$

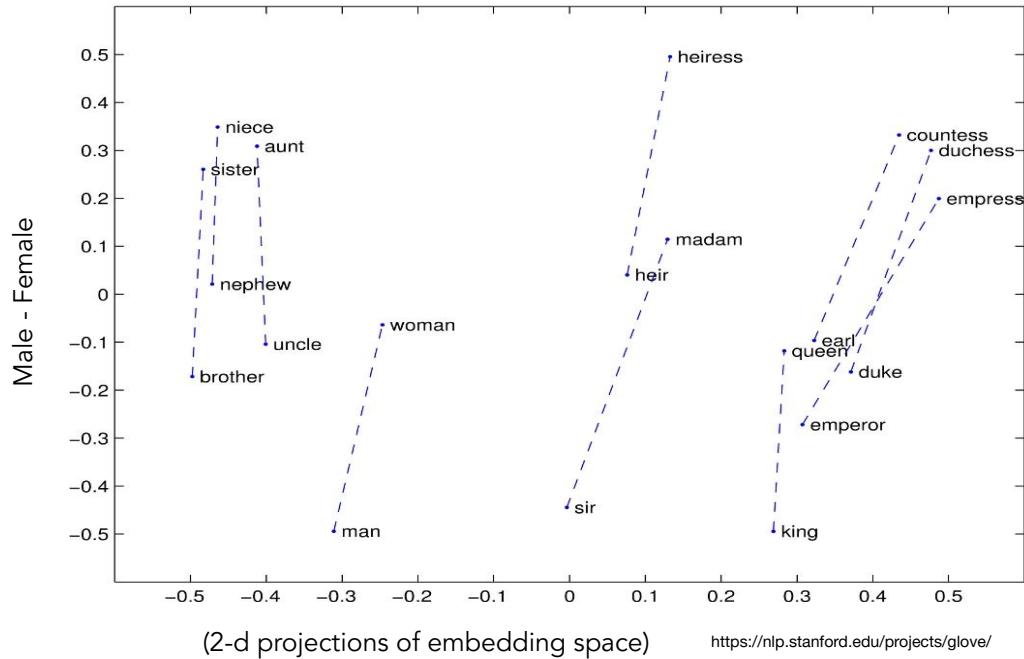
$\text{features}(v_i) = We_i \in \mathbb{R}^k = \text{ith column of } W$

# As always, hyperparameters

- Vocabulary size  $V$
- Context window  $C$
- Number of negative examples  $k$
- Embedding dimension  $d$
- The usual:
  - learning rate etc.
- → **Empirical** evaluation!



# What now?

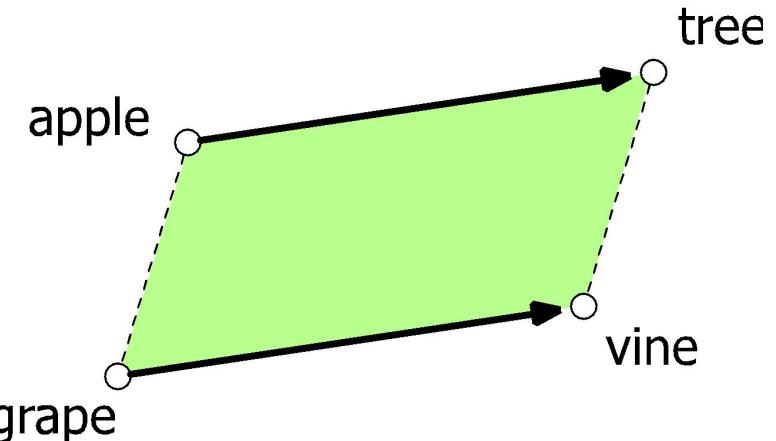


# Intrinsic evaluation

- Do (cosine) similarities of pairs of words' vectors correlate with judgments of similarity by humans?
- TOEFL-like synonym tests, e.g., rug → {sofa ✗ ottoman ✗ carpet ✓ hallway ✗}
- analogies:
  - syntactic
  - semantic

# Analogical relations

- The classic parallelogram model of analogical reasoning (Rumelhart and Abrahamson 1973)
- To solve: "apple is to tree as grape is to \_\_\_\_\_"
- Add tree – apple to grape to get vine
- Syntactic analogies, e.g., "walking is to walked as eating is to what?" Solved via:



$$\max_{v \in \mathcal{V}} \cos(\mathbf{v}_v, -\mathbf{v}_{\text{walking}} + \mathbf{v}_{\text{walked}} + \mathbf{v}_{\text{eating}})$$

# Quantitatively

WS353 (WORDSIM) [13]		MEN (WORDSIM) [4]	
Representation	Corr.	Representation	Corr.
SVD (k=5)	0.691	SVD (k=1)	0.735
SPPMI (k=15)	0.687	SVD (k=5)	0.734
SPPMI (k=5)	0.670	SPPMI (k=5)	0.721
<b>SGNS (k=15)</b>	0.666	SPPMI (k=15)	0.719
SVD (k=15)	0.661	<b>SGNS (k=15)</b>	0.716
SVD (k=1)	0.652	SGNS (k=5)	0.708
SGNS (k=5)	0.644	SVD (k=15)	0.694
SGNS (k=1)	0.633	SGNS (k=1)	0.690
SPPMI (k=1)	0.605	SPPMI (k=1)	0.688

Spearman's  $\rho$    k is the number of "negative" samples

MEN : 3000 items

a	b	label
sun	sunlight	50.0
automobile	car	50.0
river	water	49.0
stairs	staircase	49.0
morning	sunrise	49.0
...	...	...
feathers	truck	1.0
festival	whiskers	1.0
muscle	tulip	1.0
bikini	pizza	1.0
bakery	zebra	0.0

# Quantitatively

MIXED ANALOGIES [20]		SYNT. ANALOGIES [22]	
Representation	Acc.	Representation	Acc.
SPPMI (k=1)	0.655	SGNS (k=15)	0.627
SPPMI (k=5)	0.644	SGNS (k=5)	0.619
<b>SGNS (k=15)</b>	0.619	SGNS (k=1)	0.59
SGNS (k=5)	0.616	SPPMI (k=5)	0.466
SPPMI (k=15)	0.571	SVD (k=1)	0.448
SVD (k=1)	0.567	SPPMI (k=1)	0.445
SGNS (k=1)	0.540	SPPMI (k=15)	0.353
SVD (k=5)	0.472	SVD (k=5)	0.337
SVD (k=15)	0.341	SVD (k=15)	0.208

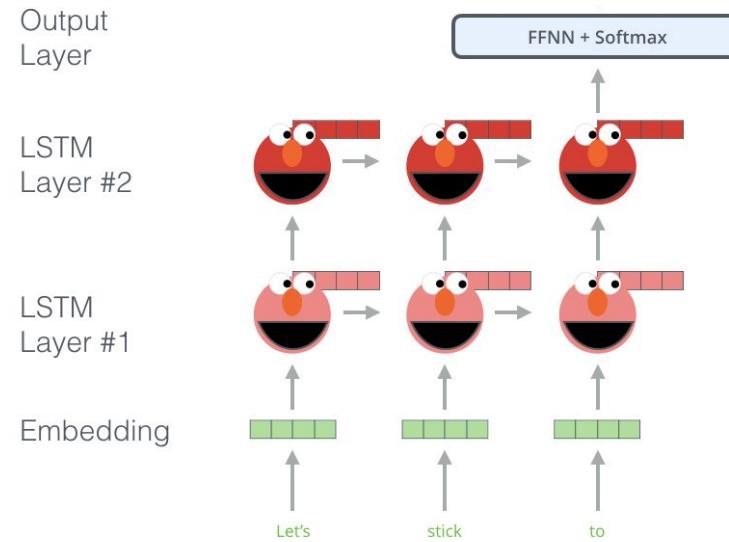
k is the number of “negative” samples

Word Pair 1		Word Pair 2	
Athens	Greece	Oslo	Norway
Astana	Kazakhstan	Harare	Zimbabwe
Angola	kwanza	Iran	rial
Chicago	Illinois	Stockton	California
brother	sister	grandson	granddaughter

Example
good:better rough:___
good:best rough:___
better:best rougher:___
year:years law:___
city:city's bank:___
see:saw return:___
see:sees return:___
saw:sees returned:___

# Extrinsic evaluation

- Embeddings are the first brick of any more complex models (described in next class)
- Embeddings can be initialized with Skip-gram: **pretraining/transfer** learning
- either keep them **frozen** or **fine-tune** them



# Named Entity Recognition with pretrained embeddings

Washington is the capital of the USA. It hosts the White House.

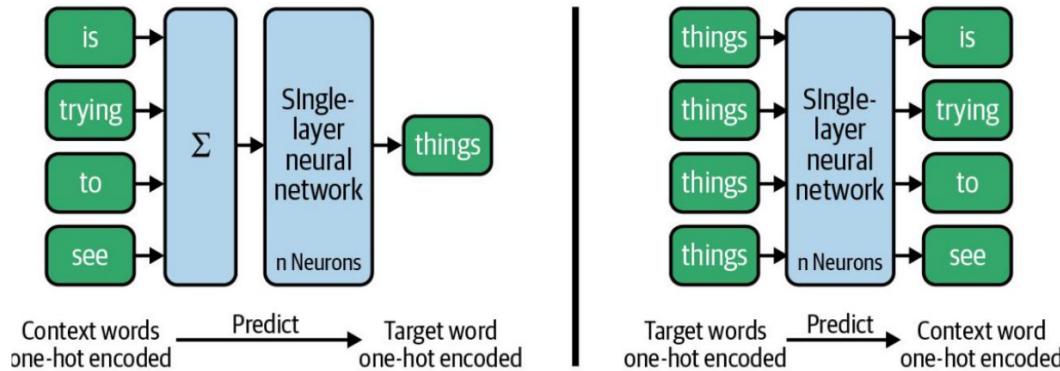
Model	Dev	Test	ACE	MUC7
Discrete	91.0	85.4	77.4	73.4
SVD	90.8	85.7	77.3	73.7
SVD-S	91.0	85.5	77.6	74.3
SVD-L	90.5	84.8	73.6	71.5
HPCA	92.6	<b>88.7</b>	81.7	80.7
HSMN	90.5	85.7	78.7	74.7
CW	92.2	87.4	81.7	80.2
CBOW	93.1	88.2	82.2	81.1
GloVe	<b>93.2</b>	88.3	<b>82.9</b>	<b>82.2</b>

F1 score

# Alternatives to Skipgram: continuous bag of words (CBOW)

instead of predicting context from word, predict word from context (much like a language model)

Life **is trying things to see** if they work. (Ray Bradbury)



# Alternatives to Skipgram: continuous bag of words (CBOW)

"bag of words" because does not model word order, puts all words in the same "bag"

$$\bar{\mathbf{v}}_m = \frac{1}{2h} \sum_{n=1}^h \mathbf{v}_{w_{m+n}} + \mathbf{v}_{w_{m-n}}$$

average of embeddings for words in the immediate neighborhood ( $\mathbf{m-h}, \dots, \mathbf{m+h}$ )

- $x_1$ : yes , we have no bananas
- $x_2$ : say yes for bananas
- $x_3$ : no bananas , we say

	1	2	3
,	1	0	1
bananas	1	1	1
for	0	1	0
have	1	0	0
no	1	0	1
say	0	1	1
we	1	0	1
yes	1	1	0

# Alternatives to Skipgram: continuous bag of words (CBOW)

$$\begin{aligned}\log p(\mathbf{w}) &\approx \sum_{m=1}^M \log p(w_m \mid w_{m-h}, w_{m-h+1}, \dots, w_{m+h-1}, w_{m+h}) \\&= \sum_{m=1}^M \log \frac{\exp (\mathbf{u}_{w_m} \cdot \bar{\mathbf{v}}_m)}{\sum_{j=1}^V \exp (\mathbf{u}_j \cdot \bar{\mathbf{v}}_m)} \\&= \sum_{m=1}^M \mathbf{u}_{w_m} \cdot \bar{\mathbf{v}}_m - \log \sum_{j=1}^V \exp (\mathbf{u}_j \cdot \bar{\mathbf{v}}_m).\end{aligned}$$

# Empirical comparison

Model	Size	WS353	MC	RG	SCWS	RW
SVD	6B	35.3	35.1	42.5	38.3	25.6
SVD-S	6B	56.5	71.5	71.0	53.6	34.7
SVD-L	6B	65.7	72.7	75.1	56.5	37.0
CBOW <sup>†</sup>	6B	57.2	65.6	68.2	57.0	32.5
SG <sup>†</sup>	6B	62.8	65.2	69.7	58.1	37.2
GloVe	6B	65.8	72.7	77.8	53.9	38.1
SVD-L	42B	74.0	76.4	74.1	58.3	39.9
GloVe	42B	75.9	83.6	82.9	59.6	47.8
CBOW*	100B	68.4	79.6	75.4	59.4	45.5

Spearman's  $\rho$

Model	Dim.	Size	Sem.	Syn.	Tot.
ivLBL	100	1.5B	55.9	50.1	53.2
HPCA	100	1.6B	4.2	16.4	10.8
GloVe	100	1.6B	67.5	54.3	60.3
SG	300	1B	61	61	61
CBOW	300	1.6B	16.1	52.6	36.1
vLBL	300	1.5B	54.2	64.8	60.0
ivLBL	300	1.5B	65.2	63.0	64.0
GloVe	300	1.6B	80.8	61.5	70.3
SVD	300	6B	6.3	8.1	7.3
SVD-S	300	6B	36.7	46.6	42.1
SVD-L	300	6B	56.6	63.0	60.1
CBOW <sup>†</sup>	300	6B	63.6	67.4	65.7
SG <sup>†</sup>	300	6B	73.0	66.0	69.1
GloVe	300	6B	77.4	67.0	71.7

Word analogy

# Alternatives to Skipgram: GloVe

studies ratio of co-occurrence instead of co-occurrence

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k ice)$	$1.9 \times 10^{-4}$	$6.6 \times 10^{-5}$	$3.0 \times 10^{-3}$	$1.7 \times 10^{-5}$
$P(k steam)$	$2.2 \times 10^{-5}$	$7.8 \times 10^{-4}$	$2.2 \times 10^{-3}$	$1.8 \times 10^{-5}$
$P(k ice)/P(k steam)$	8.9	$8.5 \times 10^{-2}$	1.36	0.96

$$\min_{\mathbf{u}, \mathbf{v}, b, \tilde{b}} \sum_{j=1}^V \sum_{i \in \mathcal{C}} f(M_{ij}) \left( \widehat{\log M_{ij}} - \log M_{ij} \right)^2$$

s.t.  $\widehat{\log M_{ij}} = \mathbf{u}_i \cdot \mathbf{v}_j + b_i + \tilde{b}_j,$

$\log \text{count}(i, j)$

# Empirical comparison

Model	Size	WS353	MC	RG	SCWS	RW
SVD	6B	35.3	35.1	42.5	38.3	25.6
SVD-S	6B	56.5	71.5	71.0	53.6	34.7
SVD-L	6B	65.7	72.7	75.1	56.5	37.0
CBOW <sup>†</sup>	6B	57.2	65.6	68.2	57.0	32.5
SG <sup>†</sup>	6B	62.8	65.2	69.7	58.1	37.2
GloVe	6B	65.8	72.7	77.8	53.9	38.1
SVD-L	42B	74.0	76.4	74.1	58.3	39.9
GloVe	42B	75.9	83.6	82.9	59.6	47.8
CBOW*	100B	68.4	79.6	75.4	59.4	45.5

Spearman's  $\rho$

Model	Dim.	Size	Sem.	Syn.	Tot.
ivLBL	100	1.5B	55.9	50.1	53.2
HPCA	100	1.6B	4.2	16.4	10.8
GloVe	100	1.6B	67.5	54.3	60.3
SG	300	1B	61	61	61
CBOW	300	1.6B	16.1	52.6	36.1
vLBL	300	1.5B	54.2	64.8	60.0
ivLBL	300	1.5B	65.2	63.0	64.0
GloVe	300	1.6B	80.8	61.5	70.3
SVD	300	6B	6.3	8.1	7.3
SVD-S	300	6B	36.7	46.6	42.1
SVD-L	300	6B	56.6	63.0	60.1
CBOW <sup>†</sup>	300	6B	63.6	67.4	65.7
SG <sup>†</sup>	300	6B	73.0	66.0	69.1
GloVe	300	6B	77.4	67.0	71.7

Word analogy

# Skipgram with character n-grams (fastText)

- brother: bro, rot, oth, the, her (trigrams)
- brothers: bro, rot, oth, the, her, ers : almost the same!
- also enables to model Out-of-Vocabulary words (OOV), e.g. brotha
- rough way of modelling **morphology**: relation between words
- same objective as skipgram:  $\log \left(1 + e^{-s(w_t, w_c)}\right) + \sum_{n \in \mathcal{N}_{t,c}} \log \left(1 + e^{s(w_t, n)}\right)$
- simply redefine similarity:  
sum over all n-grams of  
the word

$$s(w, c) = \sum_{g \in \mathcal{G}_w} \mathbf{z}_g^\top \mathbf{v}_c$$

# Empirical comparison

		sg	cbow	sisg-	sisg
AR	WS353	51	52	54	<b>55</b>
	GUR350	61	62	64	<b>70</b>
DE	GUR65	78	78	<b>81</b>	<b>81</b>
	ZG222	35	38	41	<b>44</b>
EN	RW	43	43	46	<b>47</b>
	WS353	72	<b>73</b>	71	71
Es	WS353	57	58	58	<b>59</b>
FR	RG65	70	69	<b>75</b>	<b>75</b>
RO	WS353	48	52	51	<b>54</b>
RU	HJ	59	60	60	<b>66</b>

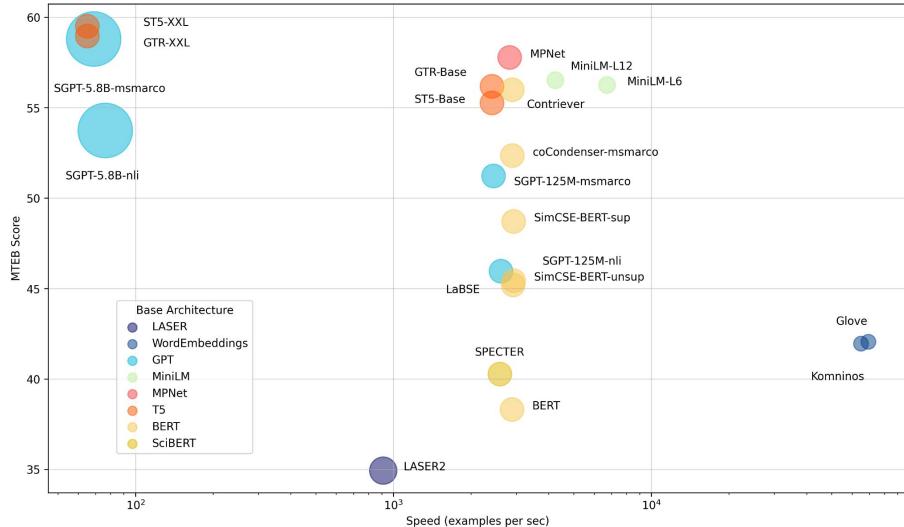
Spearman's  $\rho$

		sg	cbow	sisg
CS	Semantic	25.7	27.6	27.5
	Syntactic	52.8	55.0	77.8
DE	Semantic	66.5	66.8	62.3
	Syntactic	44.5	45.0	56.4
EN	Semantic	78.5	78.2	77.8
	Syntactic	70.1	69.9	74.9
IT	Semantic	52.3	54.7	52.3
	Syntactic	51.5	51.8	62.7

Word analogy

# Welcome LLMs, exit Embeddings?

- Large Language Models are effective but not so efficient
- Embeddings are very lightweight, relevant for many industrial applications
- fastText: efficient implementation
- LLMs build on similar hypothesis and methods as Embeddings

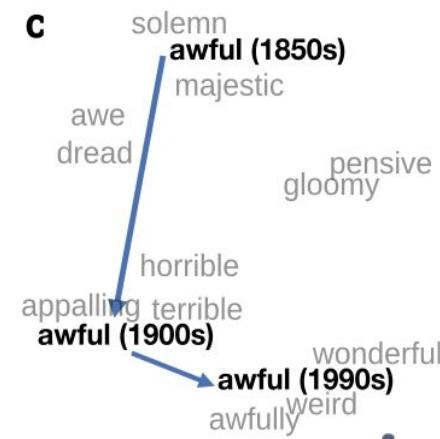
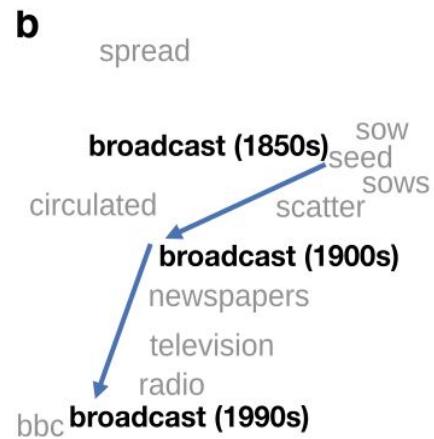
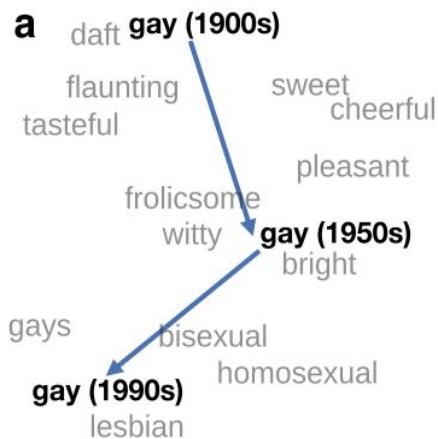


# In Summary

- NLP = research field at the intersection of Computer Science and Linguistics
- NLP = Many industrial applications, from Machine Translation to chatbots like ChatGPT or Information Extraction
- Meaning of a word is its use in the language: distributional semantics
- Skip-gram (word2vec): compute embeddings of words by predicting their context (**self-supervised learning**)
- Use as building block (**pre-training**) or solve analogies or measure word semantic similarity

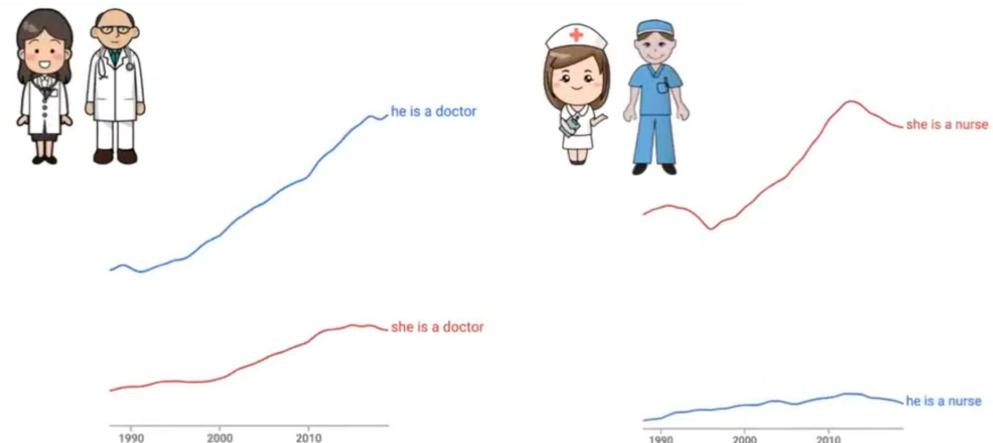
# Limitations

- Cannot model polysemy: chair [furniture] vs chair [person] has only one embedding "chair"
- Meaning changes through time/domain...



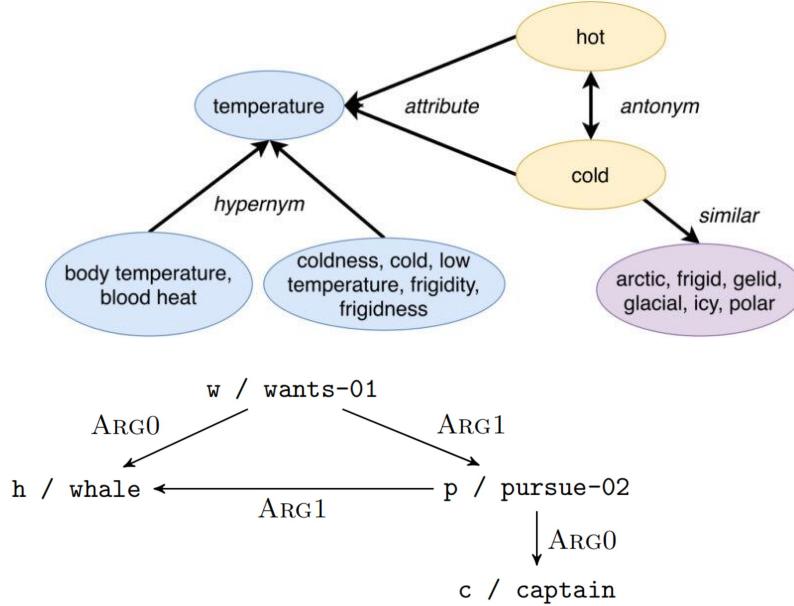
# Embeddings reflect cultural bias!

- Statistical patterns in text reflect both **intrinsic meaning** and **extrinsic use**
- Ask “father : doctor :: mother : x”  
 $x = \text{nurse}$
- Ask “man : computer programmer :: woman : x”  
 $x = \text{homemaker}$



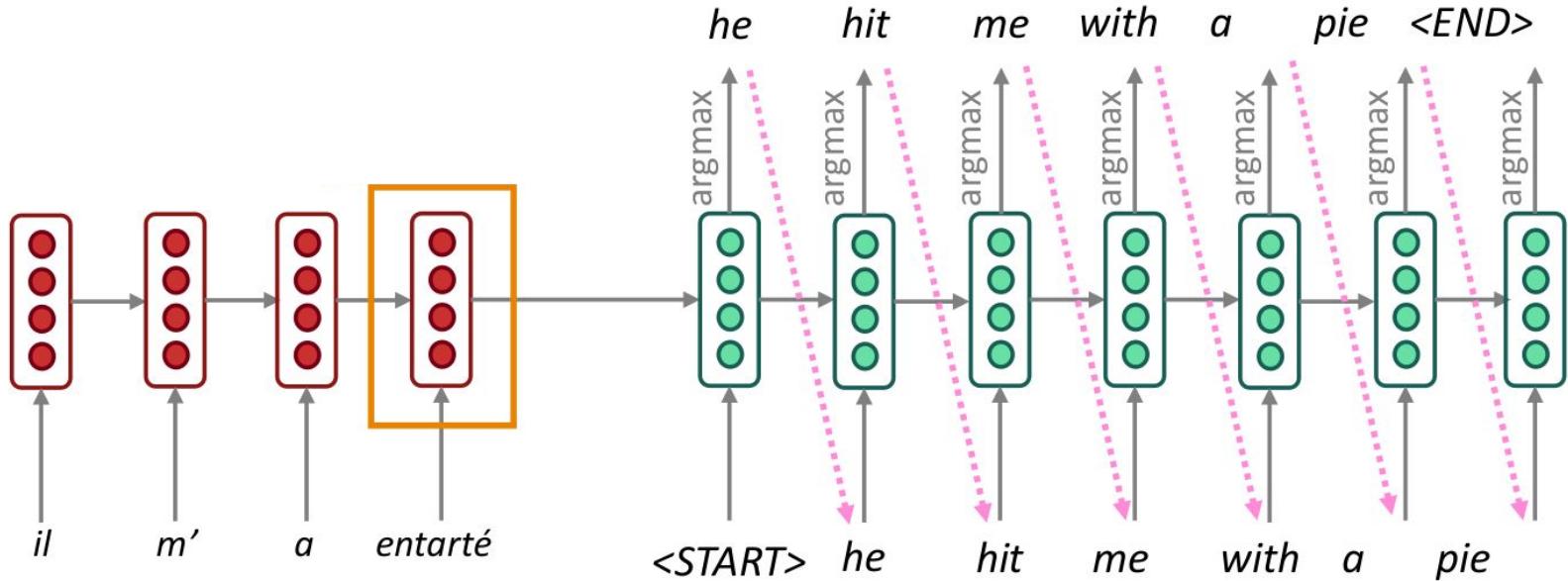
# Alternatives to distributional semantics

- Not mainstream but may prove useful... Cf. Natural Language Processing by Jacob Eisenstein (2018).
- Chapter 12: Logical semantics
- Chapter 13: Predicate-argument semantics



*"The whale wants the captain to pursue him"*

# Next class: models for sequences!





# aivancity

PARIS-CACHAN

**advancing education  
in artificial intelligence**