aivancity
SCHOOL FOR
TECHNOLOGY, BUSINESS & SOCIETY
PARIS-CACHAN

04/10/2024

Natural Language Processing (NLP)

Introduction to NLP and Distributional Semantics

Part 0

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

Part 0

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)
- **Benoît 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

Part 0

Program for this semester

- Today: 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
- Oct 10: Practical Work 1 (2 sessions)
- Oct 16: Neural Network architectures used in Large Language Models:
  - Attention Mechanism
  - Transformers

Part 0

Program for this semester

- Oct 24: Practical Work 2 (2 sessions)
- Group Homework: deadline Monday 4th of November (after Toussaint)
  - Groups of 3
  - Report of max. 4 pages, Continuous assessment (50%)
- Nov 5: Large Language Models from Shannon to ChatGPT
  - pre-training and fine-tuning
  - alignment: reinforcement learning from human feedback (RLHF)
  - decoding/generation methods

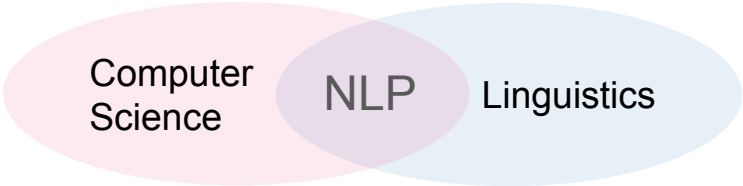
Part 0

Program for this semester

- Nov 19: Practical Work 3 (2 sessions)
- Nov 28:
  - Industrial applications and research benchmarks
  - Ethical, social, and environmental issues
- Dec 5: Practical Work 4 (2 sessions)
- Individual Final sitting Exam 50% (December, before Christmas)

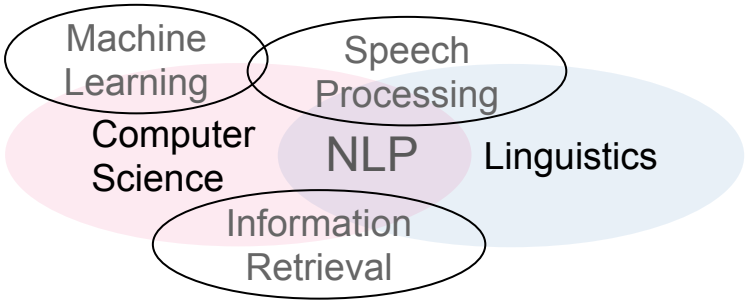
Part 1

Natural Language Processing (NLP)



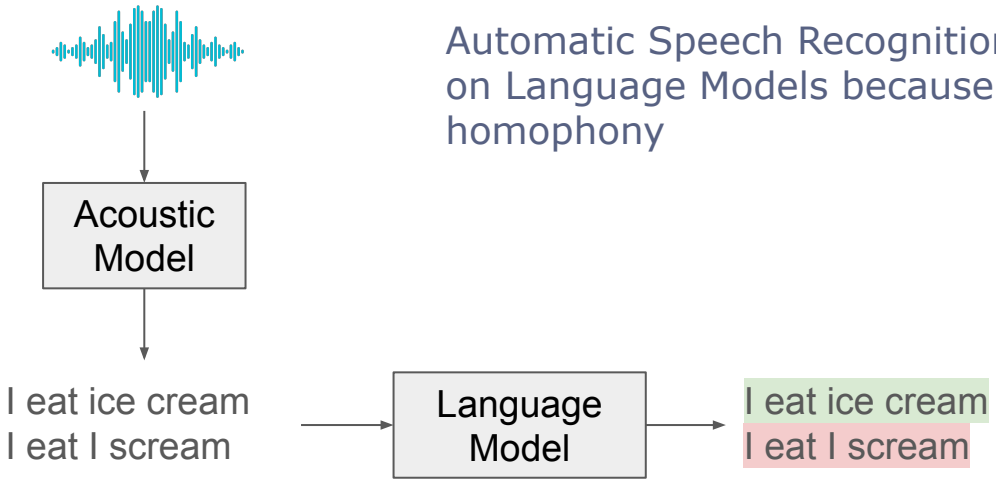
- Intersection of Computer Science and Linguistics:
  - Distributional Semantics: sense of a word from its context (today class)
  - Computational Linguistics, Computational Morphology, etc.:
    study of humans: how do we speak? how do we organize lexicon?

Natural Language Processing (NLP)



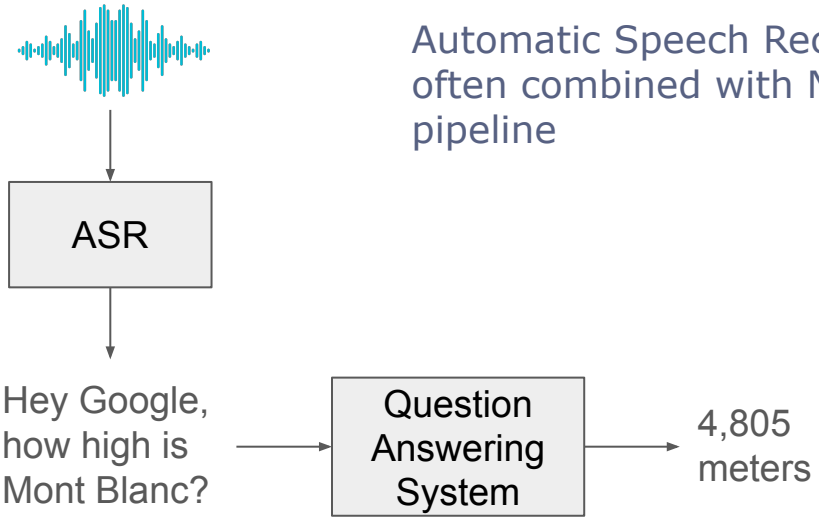
- Close to Speech Processing (Automatic Speech Recognition etc.)
- Close to Information Retrieval (Search engines like Google)
- Driven by Statistical/Machine Learning methods since the 90s (Brown, P. F., Della Pietra, S. A., Della Pietra, V. J., & Mercer, R. L. (1993). The Mathematics of Statistical Machine Translation : Parameter Estimation. Computational Linguistics, 19(2), 263-311.)
- Driven by Deep Learning since 2013 (Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. Advances in Neural Information Processing Systems)

Speech and Language Processing



Automatic Speech Recognition (ASR) relies on Language Models because of homophony

Speech and Language Processing



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

Part 1

Speech and Language Processing

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

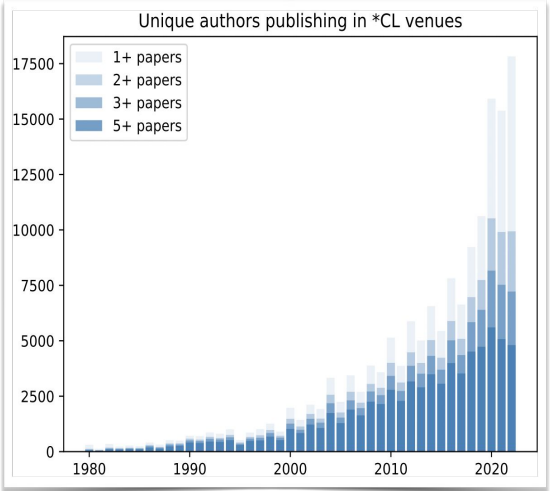
Part 1

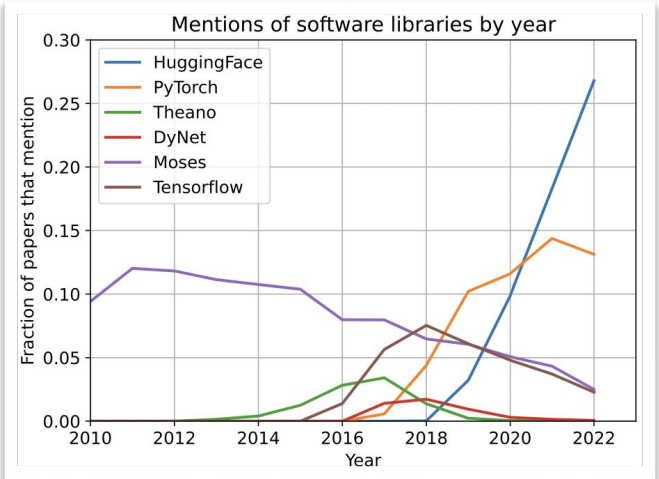
What is scientific research?

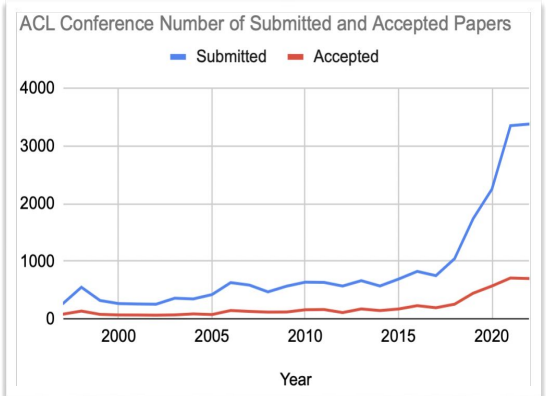
- General goal: Pushing the limits of our **knowledge**
- **Incrementally**! Find a limit/caveat in existing method and solve it!
- For example: lack of parallelization in Recurrent Neural Networks → **Transformers** (Vaswani et al. 2017)
- "Vaswani et al. 2017": a single publication that was **submitted**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

Part 1

The shape of today's NLP research

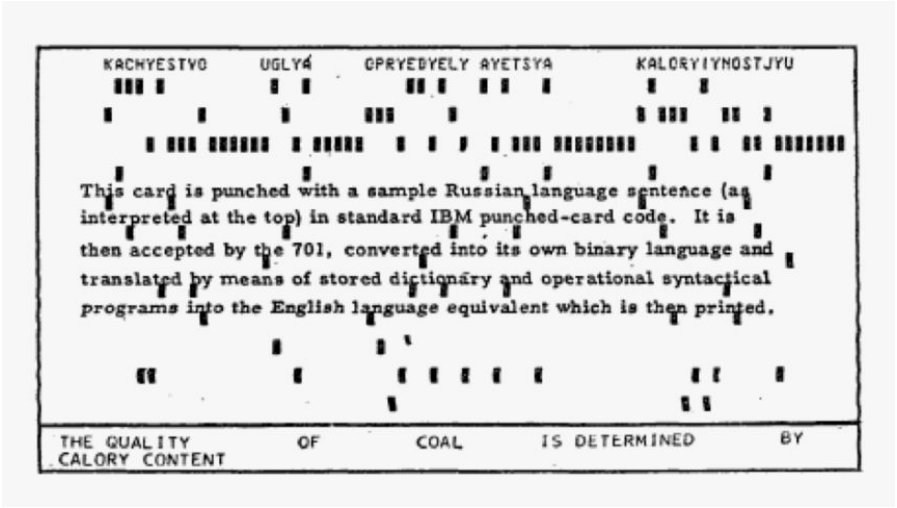






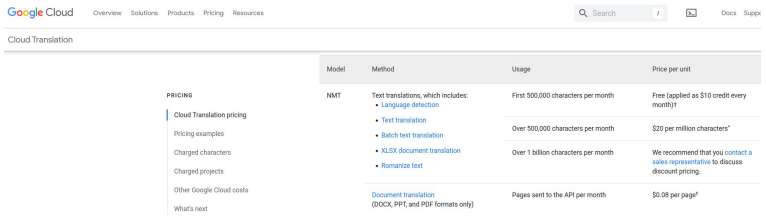
Part 1

NLP applications: Machine Translation



Georgetown–IBM experiment 1954

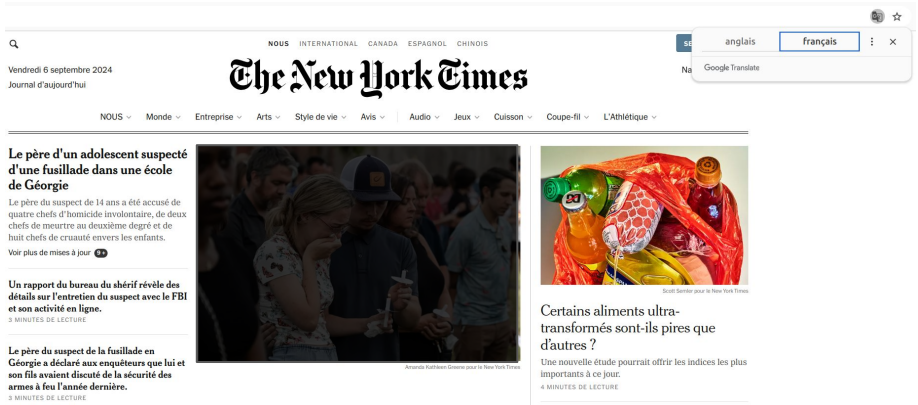
- Machine Translation is the first NLP application
- Google Translate supports 243 languages



Part 1

NLP applications: Machine Translation

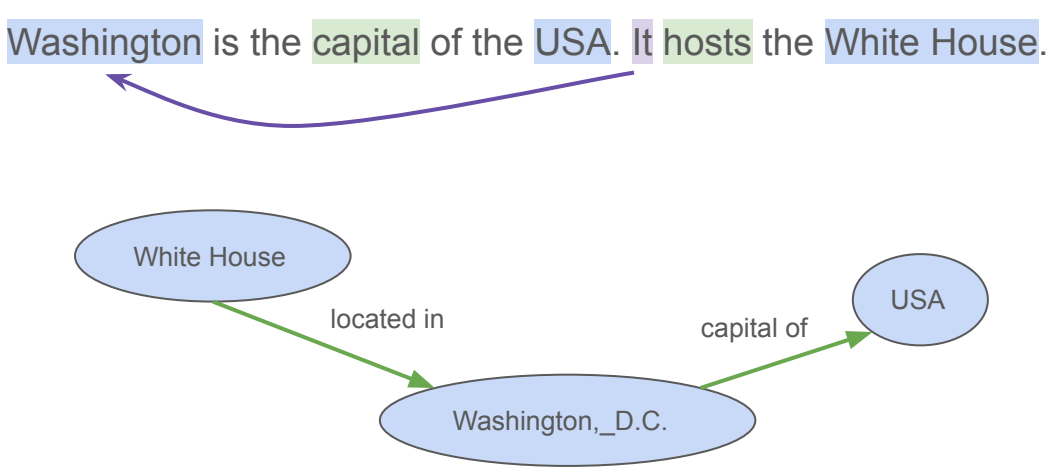
Ubiquitous on the web and social media



Part 1

NLP applications: Information Extraction

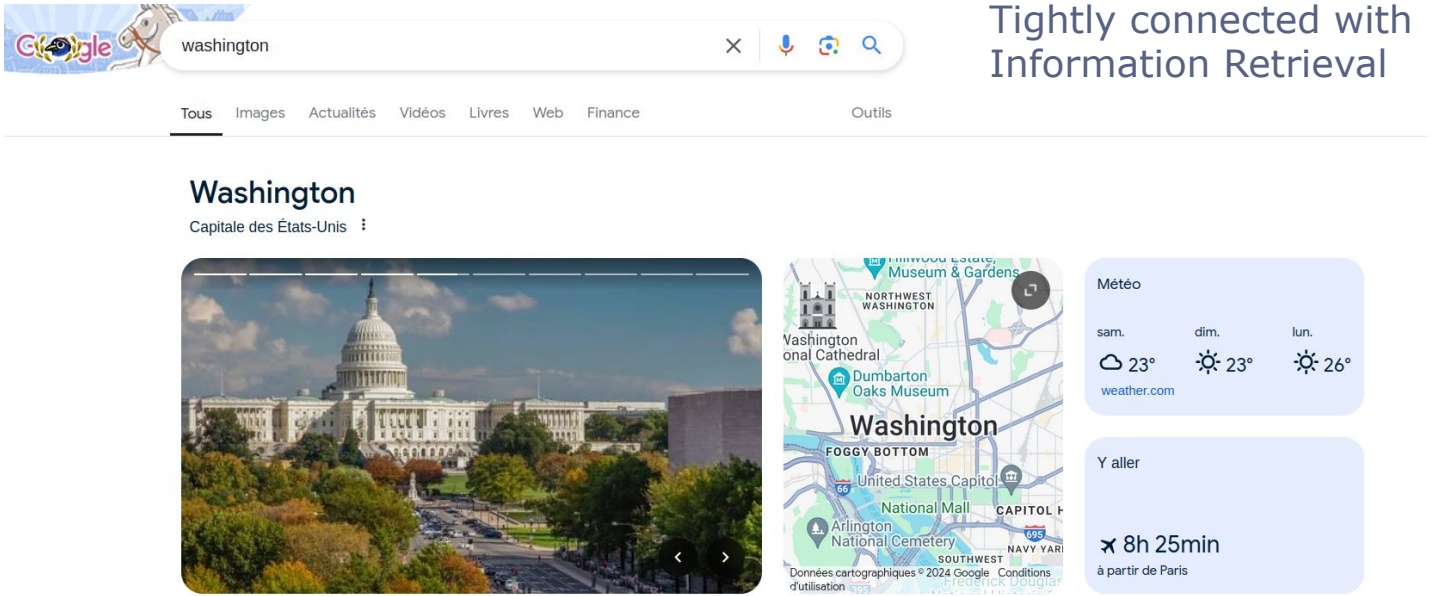
Washington,_D.C. != George_Washington



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

Part 1

NLP applications: Information Extraction

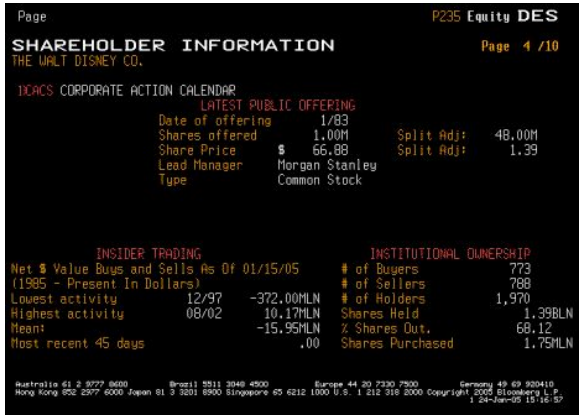


Tightly connected with Information Retrieval

Part 1

NLP applications: Information Extraction

Not only for advancing human knowledge



Bloomberg

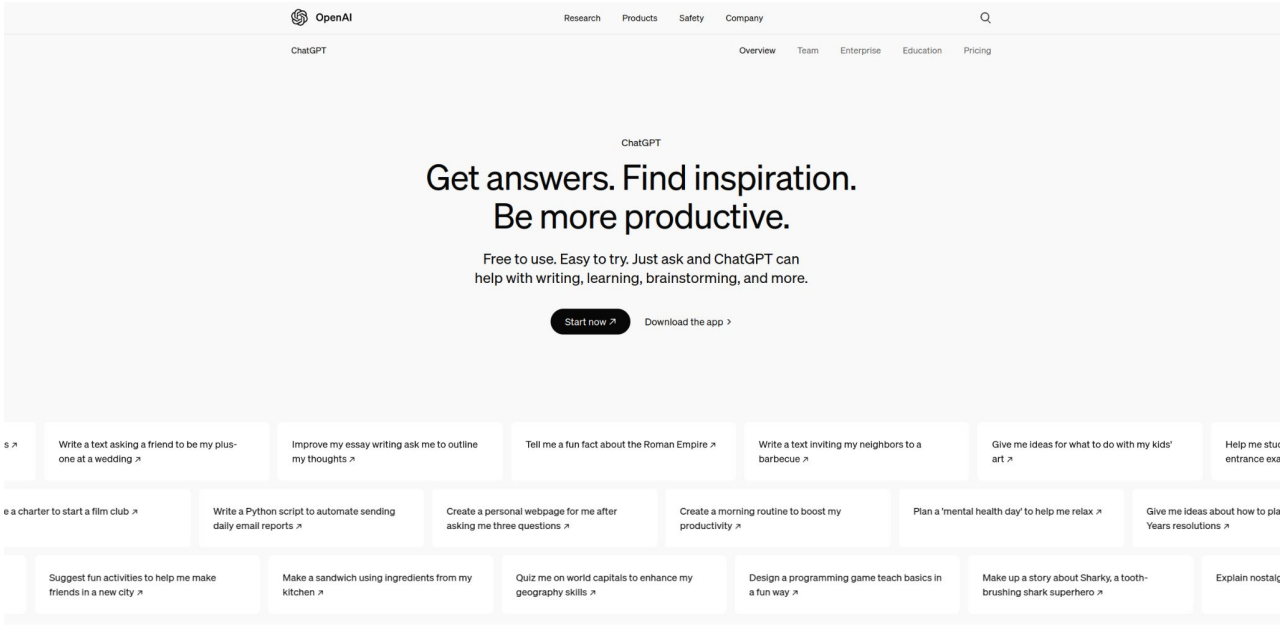
Part 1

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

Part 1

NLP applications: chatbots



Part 1

NLP applications: chatbots: demo

32 lines of python code

```python
from transformers import AutoModelForCausalLM, AutoTokenizer, BitsAndBytesConfig

def pipeline(text, model, tokenizer):
    messages = [
        {"role": "user", "content": text}
    ]
    inputs = tokenizer.apply_chat_template(messages, return_dict=True, tokenize=True, return_tensors="pt", add_generation_prompt=True)
    for k, v in inputs.items():
        inputs[k] = v.cuda()
    input_length = inputs["input_ids"].shape[1]
    output = model.generate(**inputs, max_new_tokens=128)
    output = tokenizer.batch_decode(output[:, input_length:], skip_special_tokens=True, clean_up_tokenization_spaces=True)
    return output


def user_loop(model, tokenizer):
    while True:
        answer = input(f">>> ").strip()
        output = pipeline(answer,model, tokenizer)[0]
        print(f"{output}\n")


def main():
    model_name = "google/gemma-2-2b-it"
    model  = AutoModelForCausalLM.from_pretrained(
        model_name,
        device_map="auto",
        token=True,
        quantization_config=BitsAndBytesConfig(load_in_4bit=True)
    )
    tokenizer = AutoTokenizer.from_pretrained(model_name)
    user_loop(model, tokenizer)
```

Part 1

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 b
etween 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 sema
ntics 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
```

Part 1

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 yo
ur 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 divers
e teaching methods.
```

Part 1

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 te
xt 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 p
roject?  I'm here to help!

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

Part 1

Big Money in LLMs/Chatbots

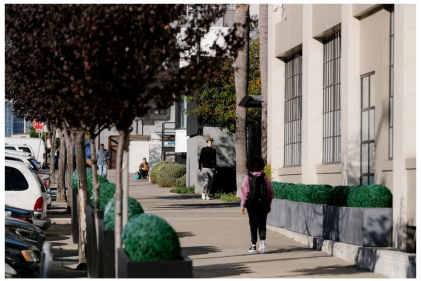
The New York Times

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.

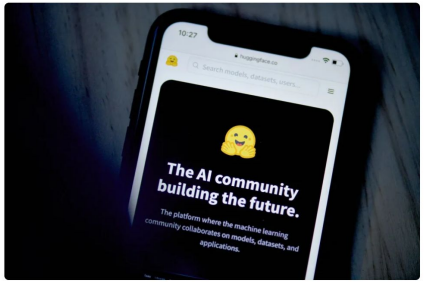
Share full article



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' | Quiz: 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 · 2:26 min Learn more | Share full article

Part 1

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

(David Paul Morris/Bloomberg via Getty Images)

Paul Lerner – *October 2024*

aivancity PARIS-CACHAN

Part 1

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.

Part 1

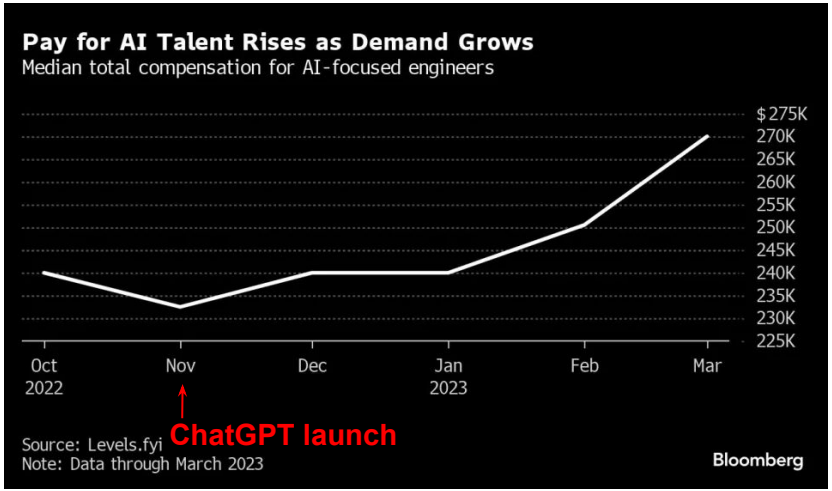
Jean Zay cluster

```
(matos) [usl47jg@jean-zay1: experiments]$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
cpu_p1*      up 4-04:00:00      2 drain$ r3i3n[8-9]
cpu_p1*      up 4-04:00:00      1  maint r2i1n16
cpu_p1*      up 4-04:00:00      1 drain* r2i7n9
cpu_p1*      up 4-04:00:00      4    mix r1i3n12,r3i0n[4,25,34]
cpu_p1*      up 4-04:00:00    711  alloc r1i0n[0-35],r1i1n[0-35],r1i2n[0-35],r1i3n[0-11,13-35],r1i4n[0-35],r1i5n[0-35],r1i6n[0-35],r1i7n[0-35],r2i0n[0-35],r2i1n[0-15,17-35],r2i2n[0-35],r2i3n[0-35],r2i4n[0-35],r2i5n[0-35],r2i6n[0-35],r2i7n[0-8,10-35],r3i0n[0-3,5-24,26-33,35],r3i1n[0-30,32-35],r3i2n[0-35],r3i3n[0-7,10-35]
cpu_p1*      up 4-04:00:00      1   idle r3i1n31
gpu_p13      up 4-04:00:00      1 drain$ r3i4n8
gpu_p13      up 4-04:00:00      3 drain* r6i0n0,r8i5n[2,5]
gpu_p13      up 4-04:00:00      1  drain r8i0n0
gpu_p13      up 4-04:00:00     99    mix r3i4n[1-2],r3i5n[0-4],r3i6n[2,4,6-7],r3i7n8,r6i0n7,r6i1n[2,4,6],r6i2n[0,2-5],r6i3n[1,6,8],r6i4n[0-1,3-5,7-8],r6i5n[1,3,5,8],r6i6n[0,7],r6i7n7,r7i0n[2-4],r7i1n5,r7i2n[1,4-5],r7i3n7,r7i4n[0-1,5,8],r7i5n3,r7i6n[2,4,7-8],r7i7n0,r8i0n1,r8i1n3,r8i2n[2-3],r8i3n6,r8i4n8,r8i5n[3,8],r8i7n[1,5,7],r9i0n4,r9i1n[0,2,7],r9i2n[0,5,7],r9i3n[2,4],r9i4n[1,5,8],r9i5n[6,8],r9i7n[0,4-5],r10i0n6,r10i1n5,r10i2n[0,2,5],r10i3n0,r10i5n[3-4],r10i6n[4,7-8],r10i7n[0,4-6]
gpu_p13      up 4-04:00:00    292  alloc r3i4n[0,3-7],r3i5n[5-8],r3i6n[0-1,3,5,8],r3i7n[0-7],r6i0n[1-6,8],r6i1n[0-1,3,5,7-8],r6i2n[1,6-8],r6i3n[0,2-5,7],r6i4n[2,6],r6i5n[0,2,4,6-7],r6i6n[1-6,8],r6i7n[0-6,8],r7i0n[0-1,5-8],r7i1n[0-4,6-8],r7i2n[0,2-3,6-8],r7i3n[0-6,8],r7i4n[2-4,6-7],r7i5n[0-2,4-8],r7i6n[0-1,3,5-6],r7i7n[1-8],r8i0n[2-8],r8i1n[0-2,4-8],r8i2n[0-1,4-8],r8i3n[0-5,7-8],r8i4n[0-7],r8i5n[0-1,4,6-7],r8i6n[0-8],r8i7n[0,2-4,6,8],r9i0n[0-3,5-8],r9i1n[1,3-6,8],r9i2n[1-4,6,8],r9i3n[0-1,3,5-8],r9i4n[0,2-4,6-7],r9i5n[0-5,7],r9i6n[0-8],r9i7n[1-3,6-8],r10i0n[0-5,7-8],r10i1n[0-4,6-8],r10i2n[1,3-4,6-8],r10i3n[1-8],r10i4n[0-8],r10i5n[0-2,5-8],r10i6n[0-3,5-6],r10i7n[1-3,7-8]
gpu_p2       up 4-04:00:00      1 drain$ jean-zay-ia810
gpu_p2       up 4-04:00:00     17    mix jean-zay-ia[802,805-807,809,812,815-816,820,822,824-827,829-831]
gpu_p2       up 4-04:00:00     12  alloc jean-zay-ia[801,803-804,811,813-814,817-819,821,823,828]
gpu_p2       up 4-04:00:00      1   idle jean-zay-ia808
gpu_p2l      up 4-04:00:00      1 drain$ jean-zay-ia810
gpu_p2l      up 4-04:00:00      5    mix jean-zay-ia[802,805-807,809]
gpu_p2l      up 4-04:00:00      4  alloc jean-zay-ia[801,803-804,811]
gpu_p2l      up 4-04:00:00      1   idle jean-zay-ia808
gpu_p2s      up 4-04:00:00     12    mix jean-zay-ia[812,815-816,820,822,824-827,829-831]
gpu_p2s      up 4-04:00:00      8  alloc jean-zay-ia[813-814,817-819,821,823,828]
gpu_p5       up 4-04:00:00     38    mix jean-zay-iam[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-iam[02,14,19,21,27,30-32,34,37,43,45,49-50]
visu         up    4:00:00      1 drain* jean-zay-visu5
visu         up    4:00:00      1   resv jean-zay-visu4
visu         up    4:00:00      1    mix jean-zay-visu1
visu         up    4:00:00      2   idle jean-zay-visu[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 idrsrv05
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 idrsrv05
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
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 550.54.15              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. |
|                                         |                        |               MIG 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 |
+-----------------------------------------+------------------------+----------------------+
```

Part 1

Big Money in LLMs/Chatbots

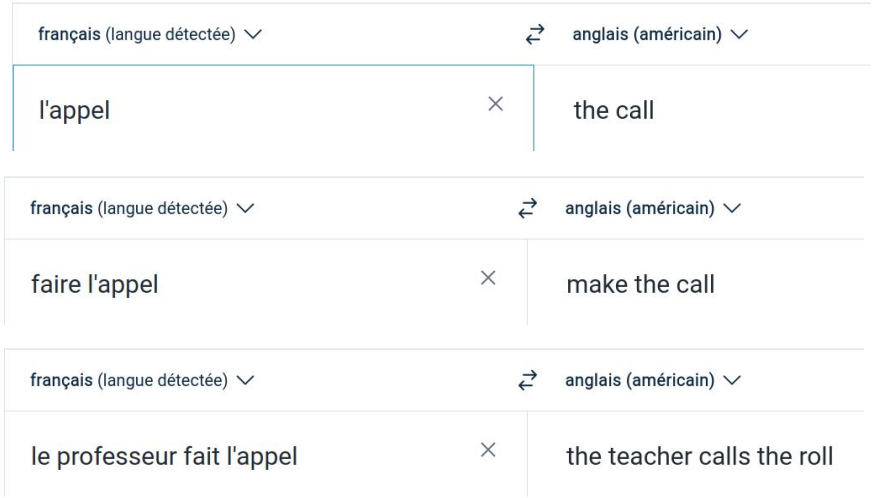


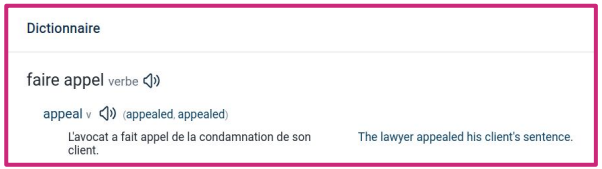
+568% in less than 2 years

Part 2

Break for questions and "appel"

"calls the roll"? (good evidence that Machine Translation is not a solved problem)





Part 2

What is a word?

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

Part 2

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 brother**s**): 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'`

Part 2

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

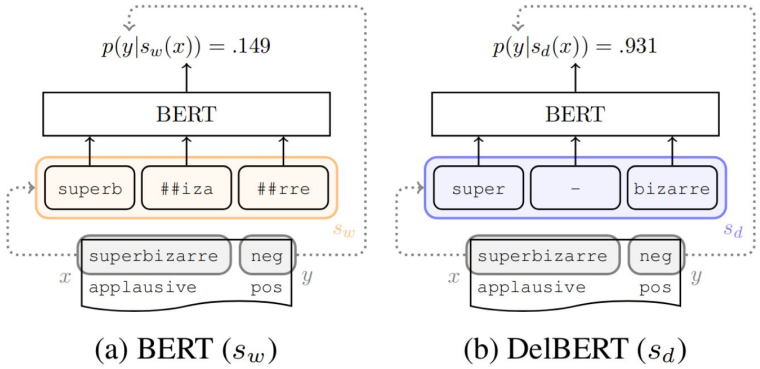
Inflection (brother vs brother**s**): "brother" is a *lemma* (singular, masc. form): useful for **indexing** (keyword-like) in Information Retrieval

Category	Infl.	Deri.	Comp.	Description	English example (input ==> output)
000	-	-	-	Root words (free morphemes)	progress ==> progress
100	✓	-	-	Inflection only	prepared ==> prepare @@ed
010	-	✓	-	Derivation only	intensive ==> intense @@ive
001	-	-	✓	Compound only	hotpot ==> hot @@pot
101	✓	-	✓	Inflection and Compound	wheelbands ==> wheel @@band @@s
011	-	✓	✓	Derivation and Compound	tankbuster ==> tank @@bust @@er
110	✓	✓	-	Inflection and Derivation	urbanizes ==> urban @@ize @@s
111	✓	✓	✓	Inflection, Derivation, Compound	trackworkers ==> track @@work @@er @@s

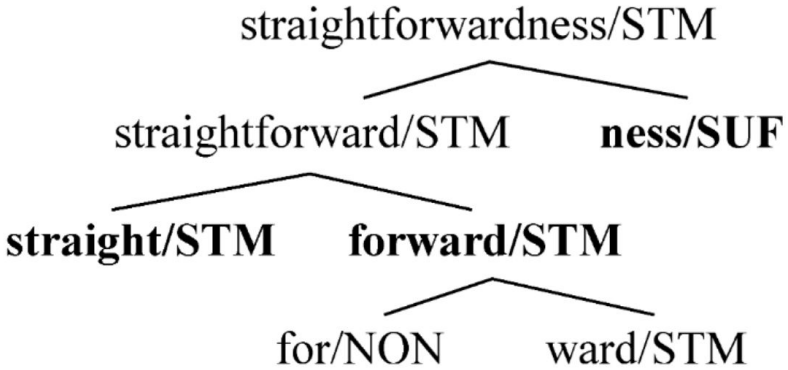
Part 2

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

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



(a) BERT (s_w) (b) DelBERT (s_d)



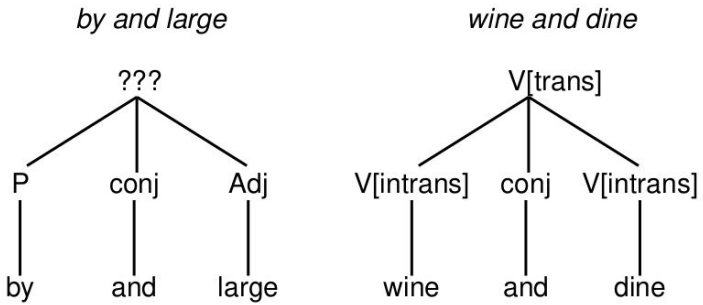
Part 2

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

Multi-word expressions ("mother in law")

Pointwise mutual information (PMI):

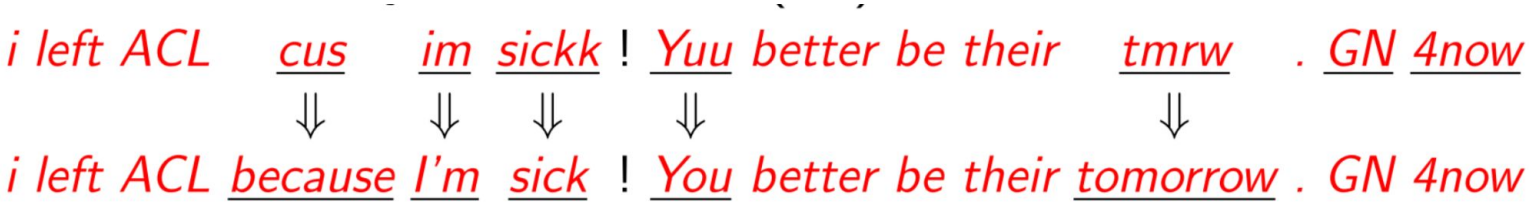
$$\log \frac{P(x,y)}{P(x)P(y)}$$



Part 2

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

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



Part 2

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"

Part 2

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

Part 2

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)

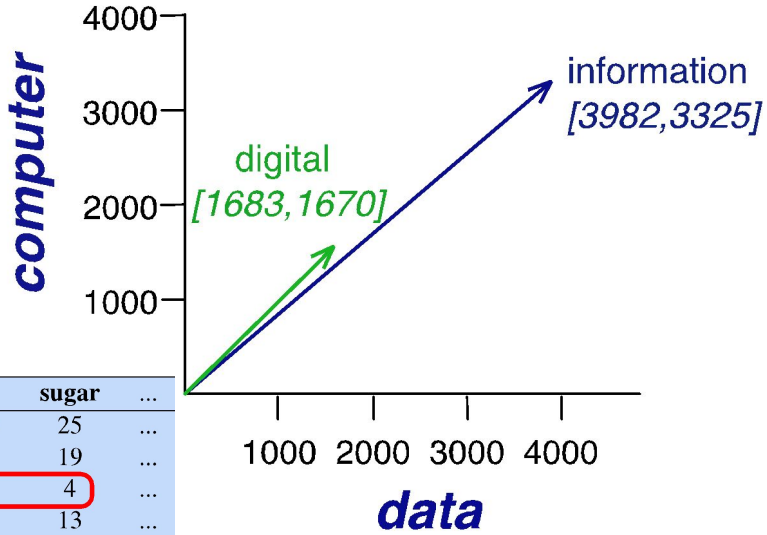
Two words are similar in meaning if their context vectors are similar

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	...	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	...

Computing word similarity: Dot product

The dot product between two vectors is a scalar:

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

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

Part 2

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

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

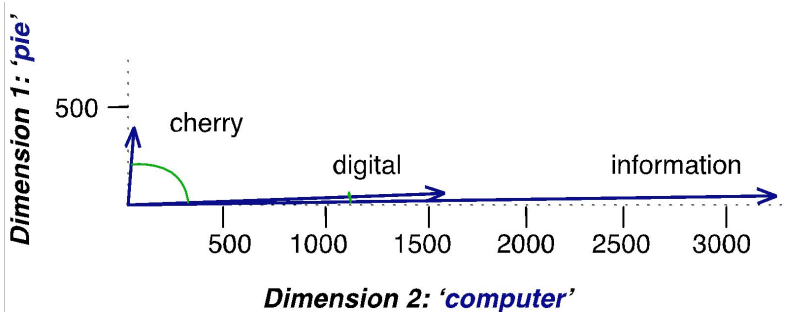
Based on the definition of the dot product between two vectors ***a*** and ***b***

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

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

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 vever co-occur together)
- Very **high dimension**! ***V***: vocabulary size (usually 20,000 - 200,000)

Part 2

How do we reduce dimensionality?

from **V** (vocabulary size) to **d** << **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)

Part 2

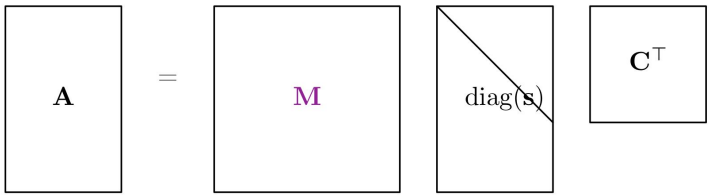
Latent Semantic Indexing/Analysis

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

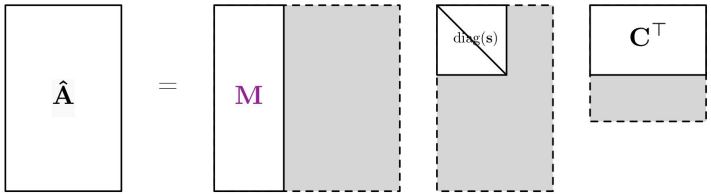
Singular Value Decomposition (SVD)

- 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:



Part 2

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 cats as the gold "correct answer" for supervised learning
  - No need for human labels

Part 2

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

Part 2

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:

$P(+|w, c)$

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

Part 2

Turning dot products into probabilities

Sim(w,c) ≈ w · c

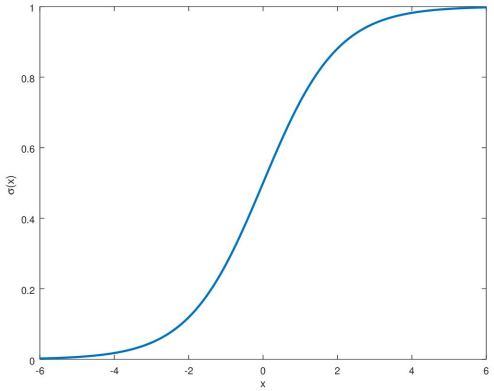
To turn this into a probability

We'll use the sigmoid from logistic regression:

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

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

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



Part 2

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.

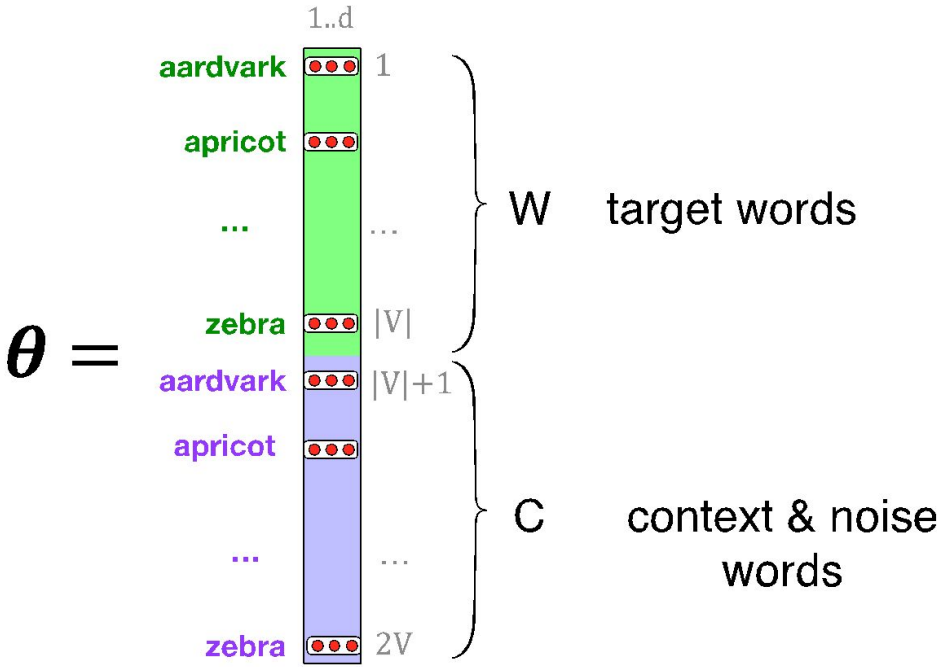
Part 2

Loss function for one w

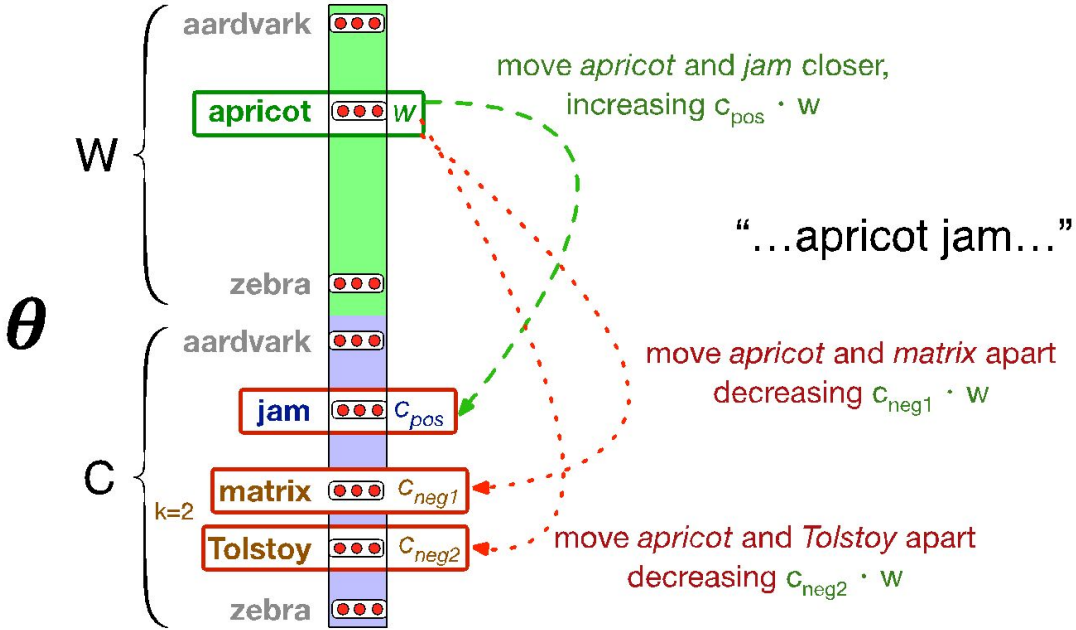
- Maximize the similarity of the target word, context word pairs ***(w, c+)*** drawn from the positive data
- Minimize the similarity of the ***(w, c-)*** pairs drawn from the negative data.

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

Learning with Stochastic gradient descent



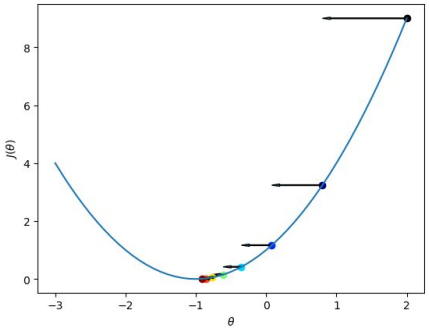
Learning with Stochastic gradient descent



Part 2

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)

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

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

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

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

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

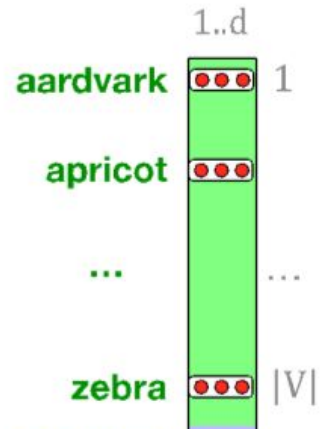
$$\frac{\partial L_{CE}}{\partial w} = [\sigma(c_{pos} \cdot w) - 1]c_{pos} + \sum_{i=1}^{k} [\sigma(c_{neg_i} \cdot w)]c_{neg_i}$$

Stochastic gradient descent update

$$
\begin{aligned}
c_{pos}^{t+1} &= c_{pos}^{t} - \eta\left[\sigma\left(c_{pos}^{t} \cdot w^{t}\right) - 1\right] w^{t} \\
c_{neg}^{t+1} &= c_{neg}^{t} - \eta\left[\sigma\left(c_{neg}^{t} \cdot w^{t}\right)\right] w^{t} \\
w^{t+1} &= w^{t} - \eta\left[\left[\sigma\left(c_{pos} \cdot w^{t}\right) - 1\right] c_{pos} + \sum_{i=1}^{k}\left[\sigma\left(c_{\mathrm{neg}_i} \cdot w^{t}\right)\right] c_{neg_i}\right]
\end{aligned}
$$

Part 2

Embedding = lookup table or linear layer?



lookup table

One-hot encoding

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

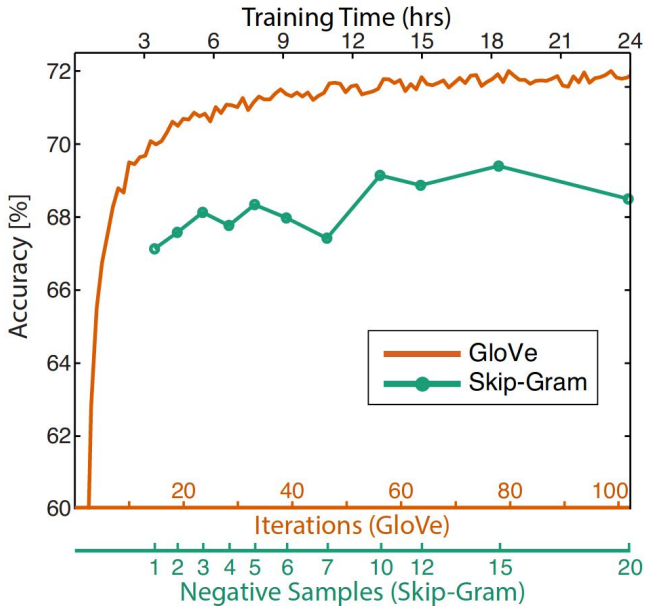
$$|\mathcal{V}| = n\text{:}$$

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

Part 2

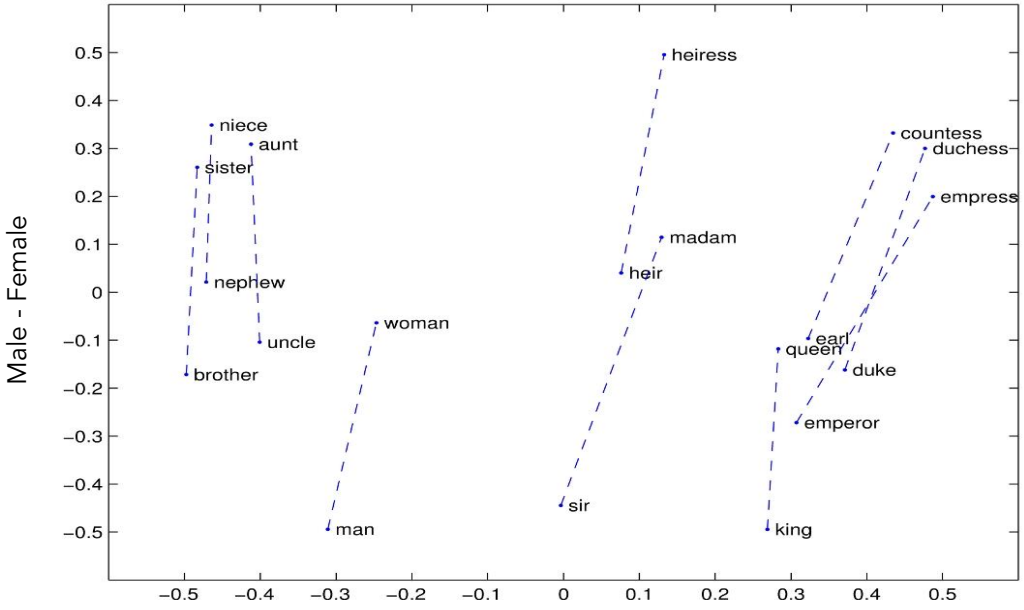
As always, hyperparameters

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



Part 2

What now?



(2-d projections of embedding space) https://nlp.stanford.edu/projects/glove/

Part 2

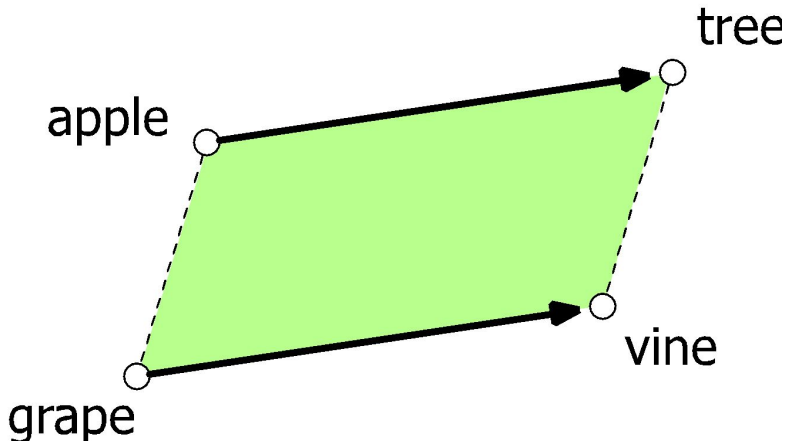
Intrinsic evaluation

- 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

Part 2

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:



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

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
SGNS	(k=15)	0.666	SPPMI	(k=15)	0.719
SVD	(k=15)	0.661	SGNS	(k=15)	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 ρ 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

Part 2

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
SGNS	(k=15)	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

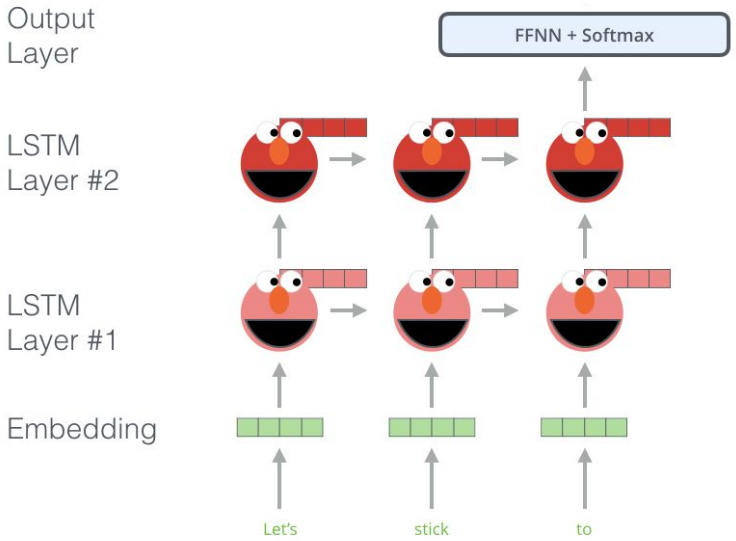
Word Pair 1		Word Pair 2	
Athens	Greece	Oslo	Norway
Astana	Kazakhstan	Harare	Zimbabwe
Angola	kwanza	Iran	rial
Chicago	Illinois	Stockton	California
brother	sister	grandson	granddaughter

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:___

Part 2

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



Part 2

Named Entity Recognition with pretrained embeddings

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

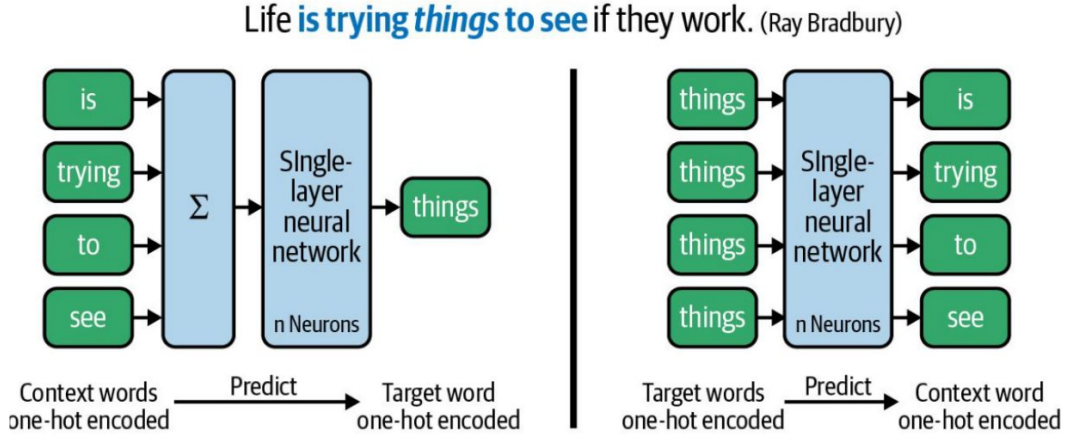
Model	Dev	Test	ACE	MUC7
Discrete	91.0	85.4	77.4	73.4
SVD	90.8	85.7	77.3	73.7
SVD-S	91.0	85.5	77.6	74.3
SVD-L	90.5	84.8	73.6	71.5
HPCA	92.6	**88.7**	81.7	80.7
HSMN	90.5	85.7	78.7	74.7
CW	92.2	87.4	81.7	80.2
CBOW	93.1	88.2	82.2	81.1
GloVe	**93.2**	88.3	**82.9**	**82.2**

F1 score

Part 2

Alternatives to Skipgram: continuous bag of words (CBOW)

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



Part 2

Alternatives to Skipgram: continuous bag of words (CBOW)

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

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

average of embeddings for words in the immediate neighborhood (**m−h**, ... , **m+h**)

$\boldsymbol{x}_1$: `yes , we have no bananas`

$\boldsymbol{x}_2$: `say yes for bananas`

$\boldsymbol{x}_3$: `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

Part 2

Alternatives to Skipgram: continuous bag of words (CBOW)

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

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†	6B	57.2	65.6	68.2	57.0	32.5
SG†	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 ρ

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†	300	6B	63.6	67.4	65.7
SG†	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	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

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

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

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

Part 2

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†	6B	57.2	65.6	68.2	57.0	32.5
SG†	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 ρ

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†	300	6B	63.6	67.4	65.7
SG†	300	6B	73.0	66.0	69.1
GloVe	300	6B	77.4	67.0	71.7

Word analogy

Part 2

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

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

Part 2

Empirical comparison

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

Spearman's ρ

		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

Part 2

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

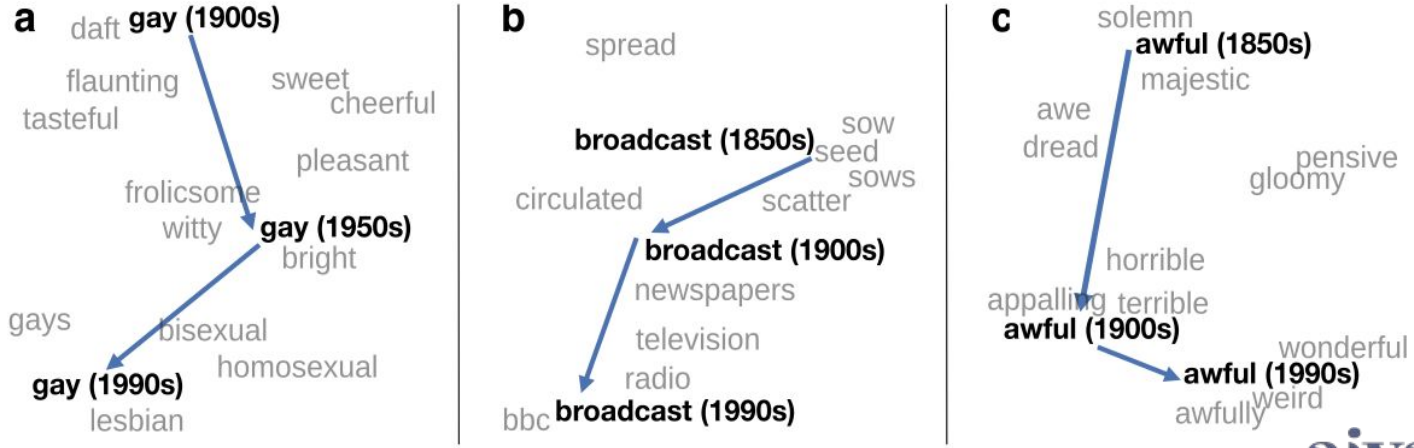
In Summary

- 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

Part 2

Limitations

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



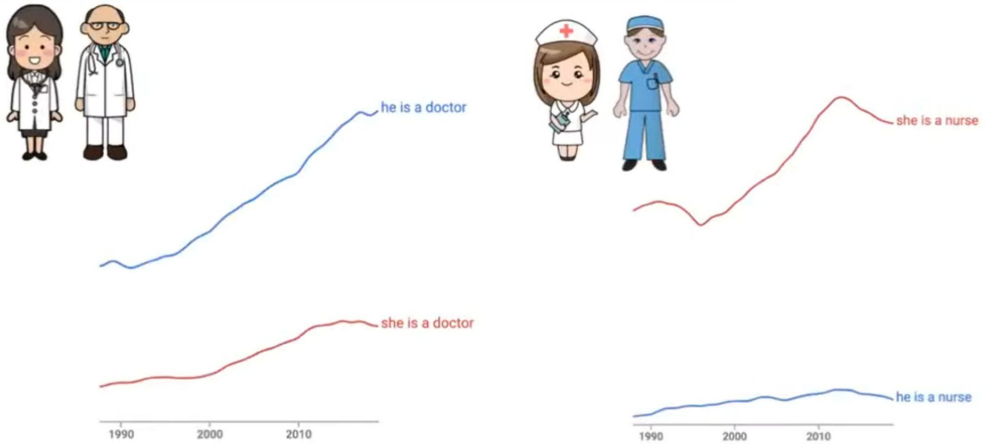
Part 2

Embeddings reflect cultural bias!

- Statistical patterns in text reflect both **intrinsic meaning** and **extrinsic use**
- Ask "father : doctor :: mother : x"

  x = nurse
- Ask "man : computer programmer :: woman : x"

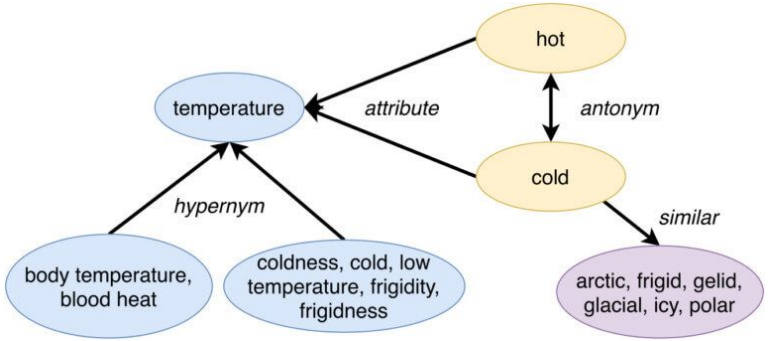
  x = homemaker

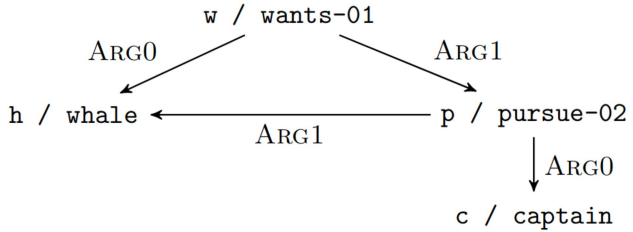


Part 2

Alternatives to distributional semantics

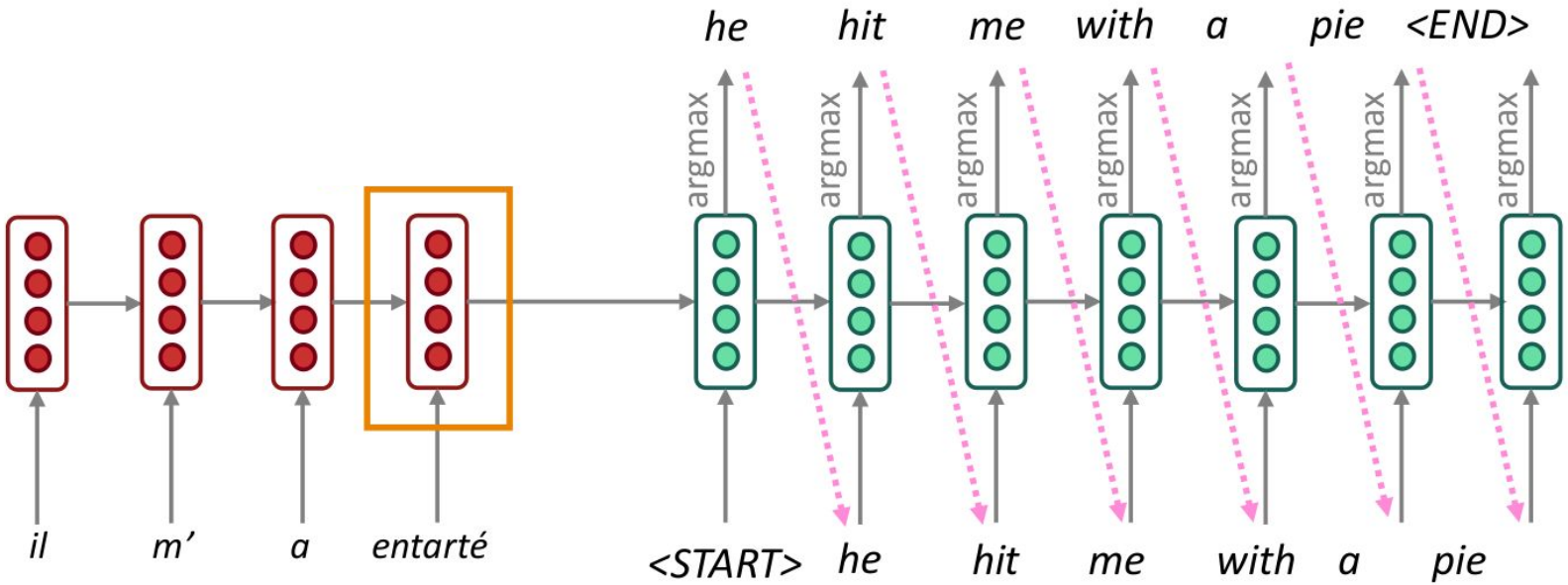
- Not mainstream but may prove useful… Cf. Natural Language Processing by Jacob Eisenstein (2018).
- Chapter 12: Logical semantics
- Chapter 13: Predicate-argument semantics





"The whale wants the captain to pursue him"

Next class: models for sequences!



aivancity

PARIS-CACHAN

advancing education
in artificial intelligence