Foundation Models & Embeddings

Lara J. Martin (she/they)

https://laramartin.net/neurosymbolic-text-gen/

Plagiarism example slides modified from Dr. Frank Ferraro

Learning Objectives

Try common prompting techniques like chain-of-thought

Recognize useful encoder-only, encoder-decoder, and decoder-only models

Understand the use & creation of dense vector embeddings

Calculate the distance between vector embeddings

Differentiate between encoder model embeddings and older dense embeddings

In-Class Activity

Think of something you're an expert in. It can be anything!

Pick an LLM that's hosted online such as ChatGPT (https://chatgpt.com/) or Claude (https://claude.ai).

Ask your LLM to give you information about that topic. Ask in different ways about different things and use different prompting techniques.

What does it do well with?

What does it not do well with?

Some Prompting Techniques:

Few-shot

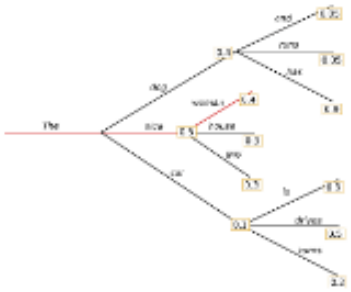
Zero-shot

Chain of thought

Decomposition

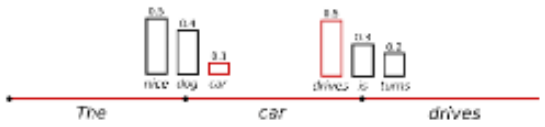
Self-Criticism

Review: Sampling

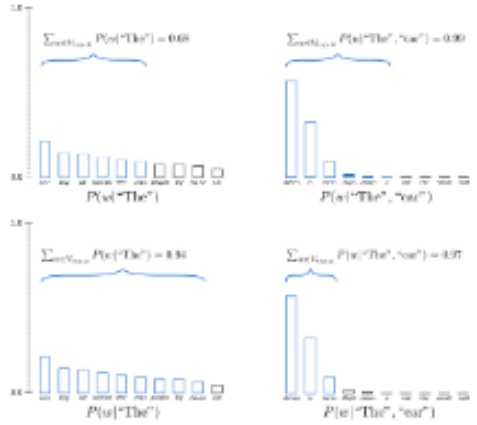


Greedy

Beam Search



Random Sampling



Top-K/P

Review

What’s the difference between finetuning and prompting?

What’s the difference between zero-shot and few-shot prompting?

Review: Tricks of the Trade

Instruction-tuned models like GPT-3.5 and Mistral-7B-Instruct like to be given a "role" first (e.g., "You are a helpful writing assistant.")

The more defined the task, the better

- More details
- One thing to do at a time

LLMs are overly confident (like people on the internet)

- To "objectively" have the model evaluate something, you should have another instance judge

Chain-of-thought prompting helps models come up with better answers

They will "Yes and…" your prompt

LLM Vocabulary

Finetuning: Training an LLM more to adapt it to your task

Pre-training: The training before finetuning; creating a foundation model

Prompting: Getting the output you want by just changing the input; the model doesn't change

Zero-shot prompting: Prompting without examples

Few-shot prompting: Prompting with a few examples

Prompt engineering: Figuring out the right prompt for a task

Prompt optimization: Automated prompt engineering

Prompt tuning: Automated creation of *latent* prompt (e.g., vector representation)

Foundation Models

Types of Foundation Models

Encoder Only

Decoder Only

Encoder-Decoder Models

Denotes what they use during pre-training

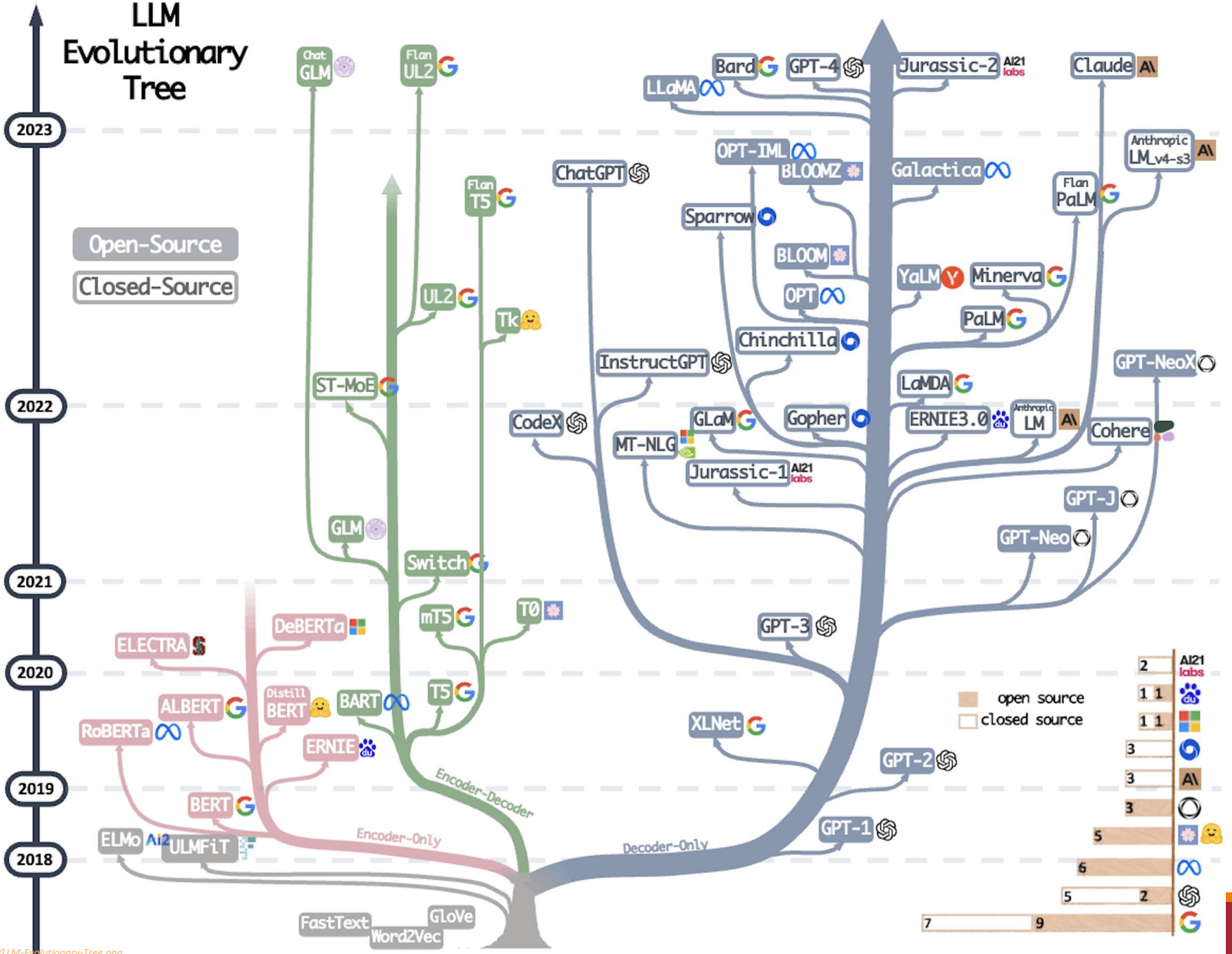
What is a foundation model?

A model that captures “foundational” or core information about a modality (e.g., text, speech, images)

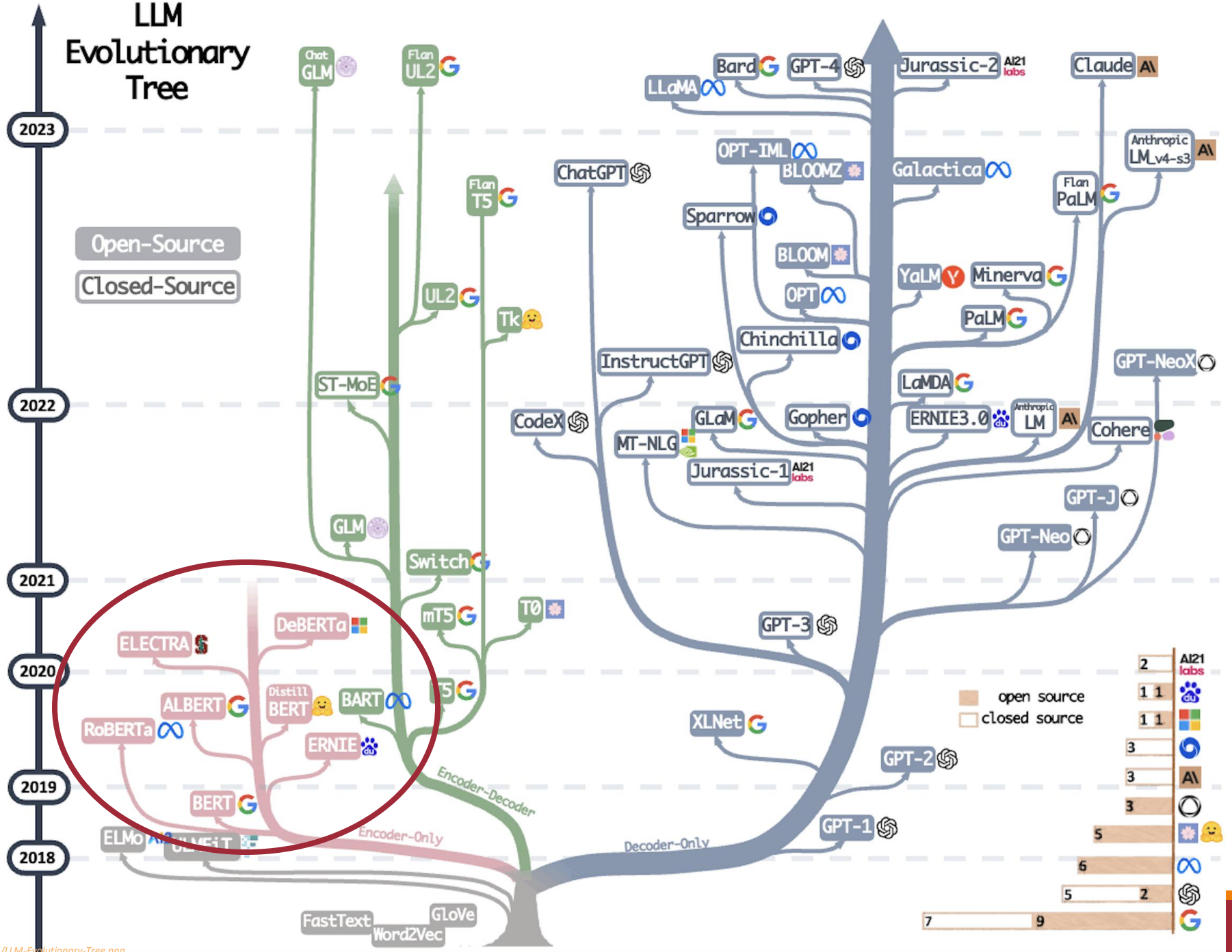
Pretrained on a large amount of data & able to be finetuned on a particular task

Self-supervised

All non-finetuned pretrained large language models (LLMs) are foundation models



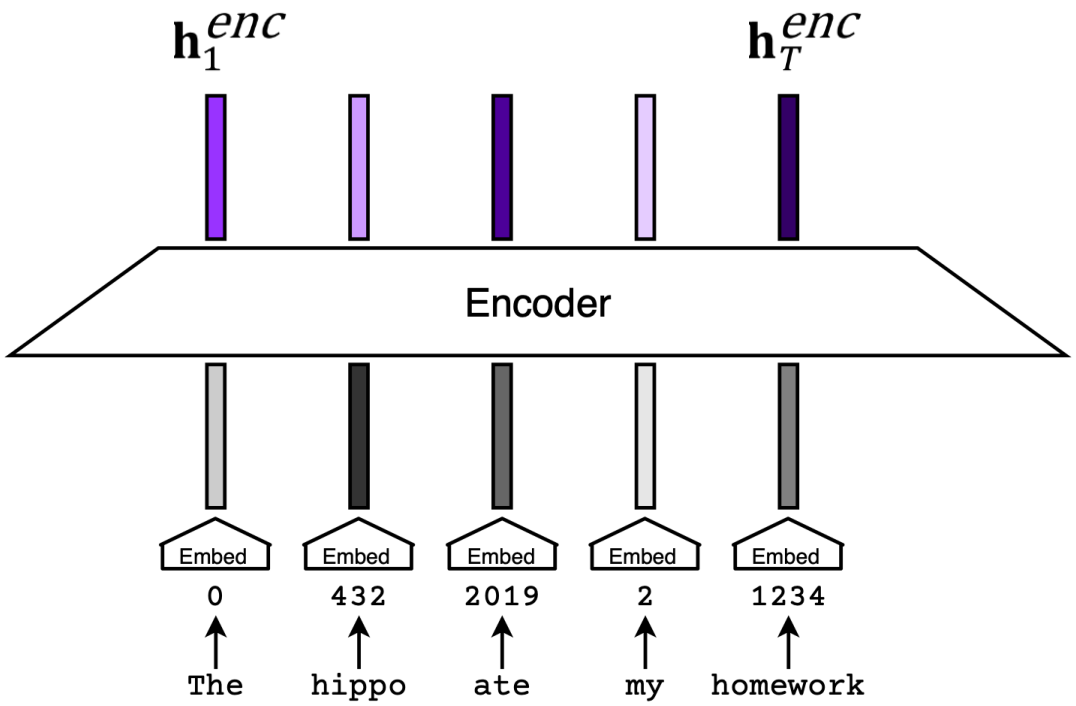
https://blog.biocomm.ai/wp-content/uploads/2023/05/LLM-Evolutionary-Tree.png

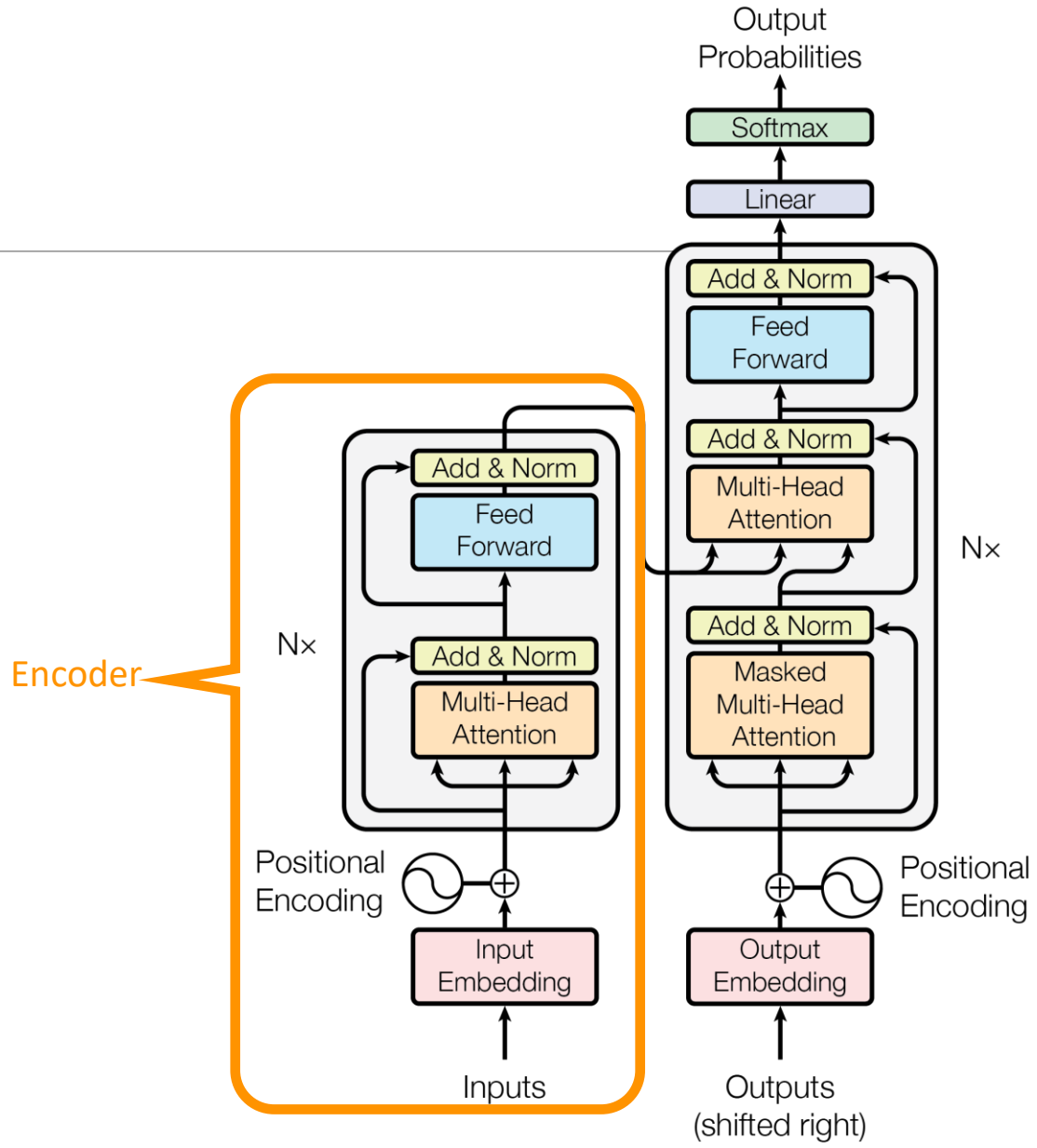


https://blog.biocomm.ai/wp-content/uploads/2023/05/LLM-Evolutionary-Tree.png

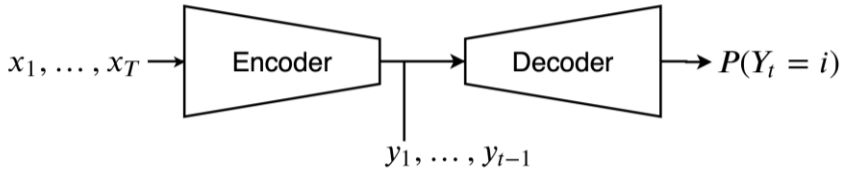

123
SESAME STREET

Encoder-only models



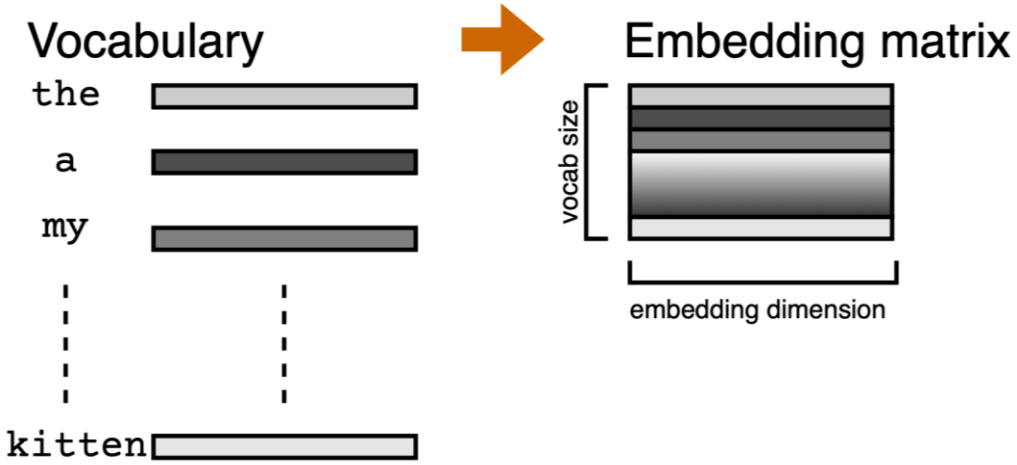


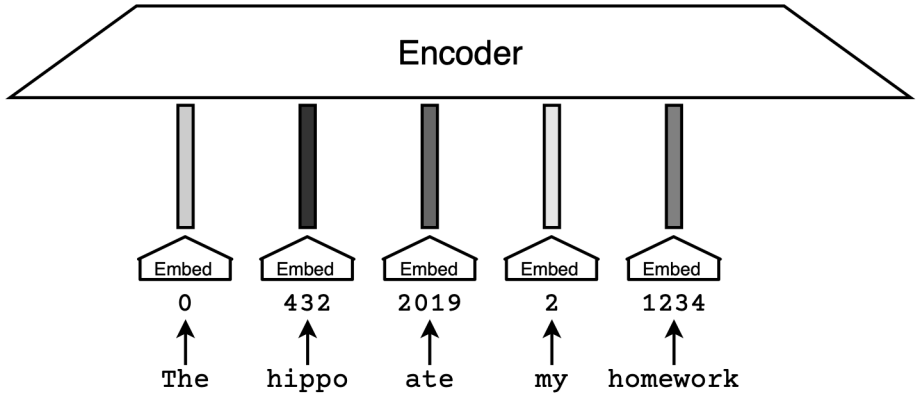
Word Embeddings



Review: Inputs to the Encoder

The encoder takes as input the embeddings corresponding to each token in the sequence.





How have we represented words?

Each word is a distinct item

- Bijection between the strings and unique integer ids:
- "cat" --> 3, "kitten" --> 792 "dog" --> 17394

- Are "cat" and "kitten" similar?

Equivalently: "One-hot" encoding

- Represent each word type w with a vector the size of the vocabulary
- This vector has V-1 zero entries, and 1 non-zero (one) entry

One-Hot Encoding Example

Let our vocab be {a, cat, saw, mouse, happy}

V = # types = 5

Assign:

a	4
cat	2
saw	3
mouse	0
happy	1

How do we represent "cat?"

$$e_{\text{cat}} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

How do we represent "happy?"

$$e_{\text{happy}} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

The Fragility of One-Hot Encodings
Case Study: Plagiarism Detector

Given two documents x_1, x_2, predict y = 1 (plagiarized) or y = 0 (not plagiarized)

What is/are the:

- Method/steps for predicting?
- General formulation?
- Features?

Image: http://3.bp.blogspot.com/_MWJDzvrnu7Y/TTcXpCWZerI/AAAAAAAAWY/zxu1GwrIBz0/s1600/excellent-mr-burns.gif

Plagiarism Detection: Word Similarity?

MAINFRAMES

Mainframes are primarily referred to large computers with rapid, advanced processing capabilities that can execute and perform tasks equivalent to many Personal Computers (PCs) machines networked together. It is characterized with high quantity Random Access Memory (RAM), very large secondary storage devices, and high-speed processors to cater for the needs of the computers under its service.

Consisting of advanced components, mainframes have the capability of running multiple large applications required by many and most enterprises and organizations. This is one of its advantages. Mainframes are also suitable to cater for those applications (programs) or files that are of very high demand by its users (clients). Examples of such organizations and enterprises using mainframes are online shopping websites such as

MAINFRAMES

Mainframes usually are referred those computers with fast, advanced processing capabilities that could perform by itself tasks that may require a lot of Personal Computers (PC) Machines. Usually mainframes would have lots of RAMs, very large secondary storage devices, and very fast processors to cater for the needs of those computers under its service.

Due to the advanced components mainframes have, these computers have the capability of running multiple large applications required by most enterprises, which is one of its advantage. Mainframes are also suitable to cater for those applications or files that are of very large demand by its users (clients). Examples of these include the large online shopping websites -i.e. : Ebay, Amazon, Microsoft, etc.

Case Study: Plagiarism Detector (Feature Example)

Given two documents x_1, x_2, predict y = 1 (plagiarized) or y = 0 (not plagiarized)

Intuition: documents are more likely to be plagiarized if they have words in common

$$f_{\text{any-common-word,Plag.}}(x_1, x_2) = ???$$

$$f_{\text{<word v>,Plag.}}(x_1, x_2) = ???$$

Example:

$$e_{\text{cat}} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

image: http://3.bp.blogspot.com/_MWJDzvrnu7Y/TTcXpCWZerI/AAAAAAAAWY/zxu1GwrIBz0/s1600/excellent-mr-burns.gif

Case Study: Plagiarism Detector (Feature Example)

Given two documents x_1, x_2, predict y = 1 (plagiarized) or y = 0 (not plagiarized)

Intuition: documents are more likely to be plagiarized if they have words in common

n # adjacent words

$$f_{\text{any-common-word,Plag.}}(x_1, x_2) = ???$$

$$f_{\text{<word v>,Plag.}}(x_1, x_2) = ???$$

$$f_{\text{<ngram Z>,Plag.}}(x_1, x_2) = ???$$

Example:

$$e_{\text{fluffy cat}} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

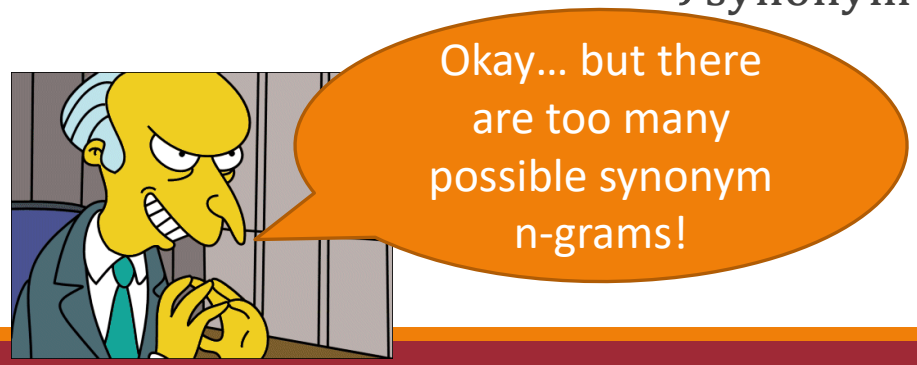
Image: http://3.bp.blogspot.com/_MWJDzvrnu7Y/TTcXpCWZerI/AAAAAAAAWY/zxu1GwrIBz0/s1600/excellent-mr-burns.gif

Case Study: Plagiarism Detector (Feature Example)

Given two documents x_1, x_2, predict y = 1 (plagiarized) or y = 0 (not plagiarized)

Intuition: documents are more likely to be plagiarized if they have words in common

$$f_{\text{any-common-word,Plag.}}(x_1, x_2) = ???$$
$$f_{\text{<word v>,Plag.}}(x_1, x_2) = ???$$
$$f_{\text{<ngram Z>,Plag.}}(x_1, x_2) = ???$$
$$f_{\text{synonym-of-<word v>,Plag.}}(x_1, x_2) = ???$$



Example:

$$e_{\text{feline}} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

Image: http://3.bp.blogspot.com/_MWJDzvrnu7Y/TTcXpCWZerI/AAAAAAAAWY/zxu1GwrIBz0/s1600/excellent-mr-burns.gif

Case Study: Plagiarism Detector (Feature Example)

Given two documents x_1, x_2, predict y = 1 (plagiarized) or y = 0 (not plagiarized)

Intuition: documents are more likely to be plagiarized if they have words in common

$$f_{\text{any-common-word,Plag.}}(x_1, x_2) = ???$$
$$f_{\text{<word v>,Plag.}}(x_1, x_2) = ???$$
$$f_{\text{<ngram Z>,Plag.}}(x_1, x_2) = ???$$
$$f_{\text{synonym-of-<word v>,Plag.}}(x_1, x_2) = ???$$
$$f_{\text{synonym-of-<ngram Z>,Plag.}}(x_1, x_2) = ???$$

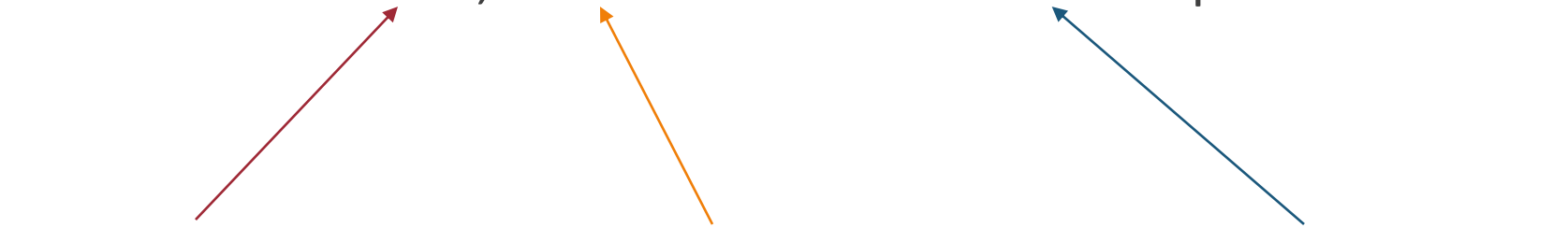
What is the synonym of

$$e_{\text{fluffy cat}} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} ?$$

image: http://3.bp.blogspot.com/_MWJDzvrnu7Y/TTcXpCWZerI/AAAAAAAAWY/zxu1GwrIBz0/s1600/excellent-mr-burns.gif

Word Embeddings

A dense, “low”-dimensional vector representation



Many values are not 0 (or at least less sparse than one-hot)

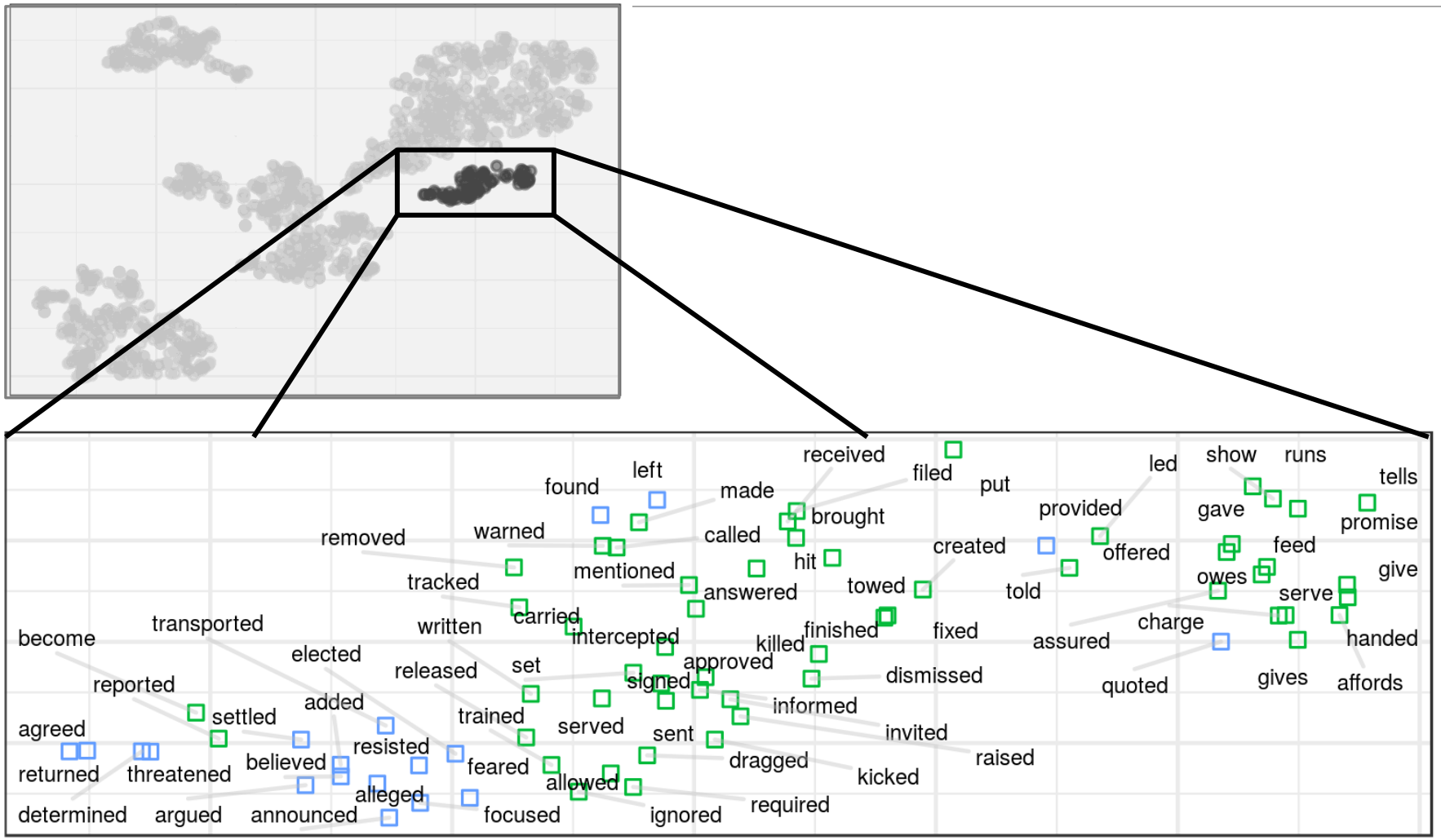
Up till ~2013: E could be any size
2013-present: E << vocab

An E-dimensional vector, often (but not always) real-valued

These are also called

- **embeddings**
- **Continuous representations**
- **(word/sentence/…) vectors**
- **Vector-space models**

A Dense Representation

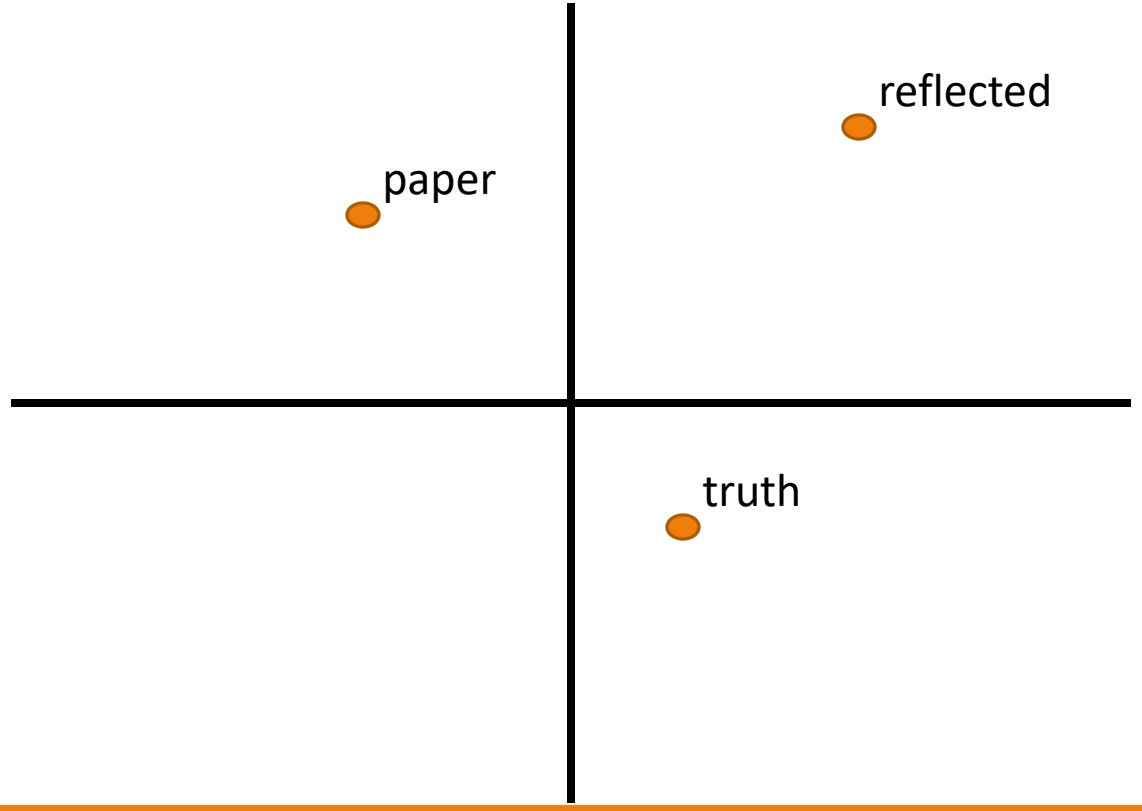


Continuous Meaning

The paper reflected the truth.

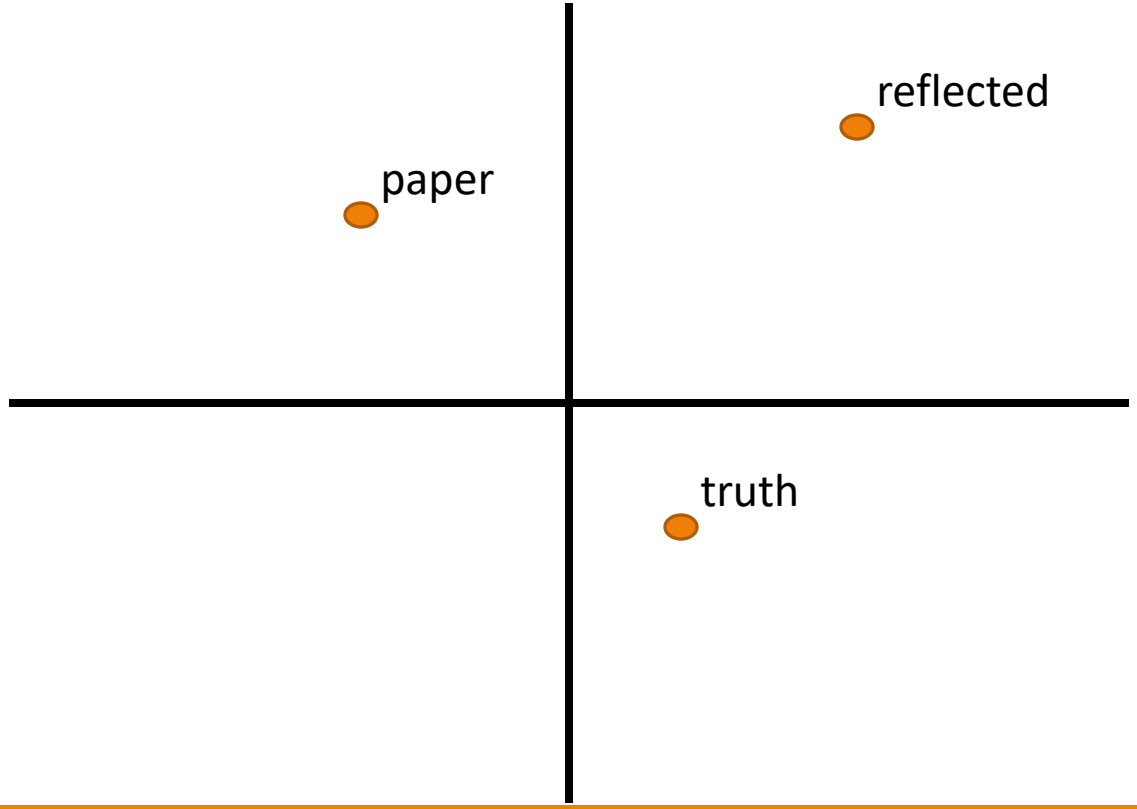
Continuous Meaning

The paper reflected the truth.



Continuous Meaning

The paper reflected the truth.



glean

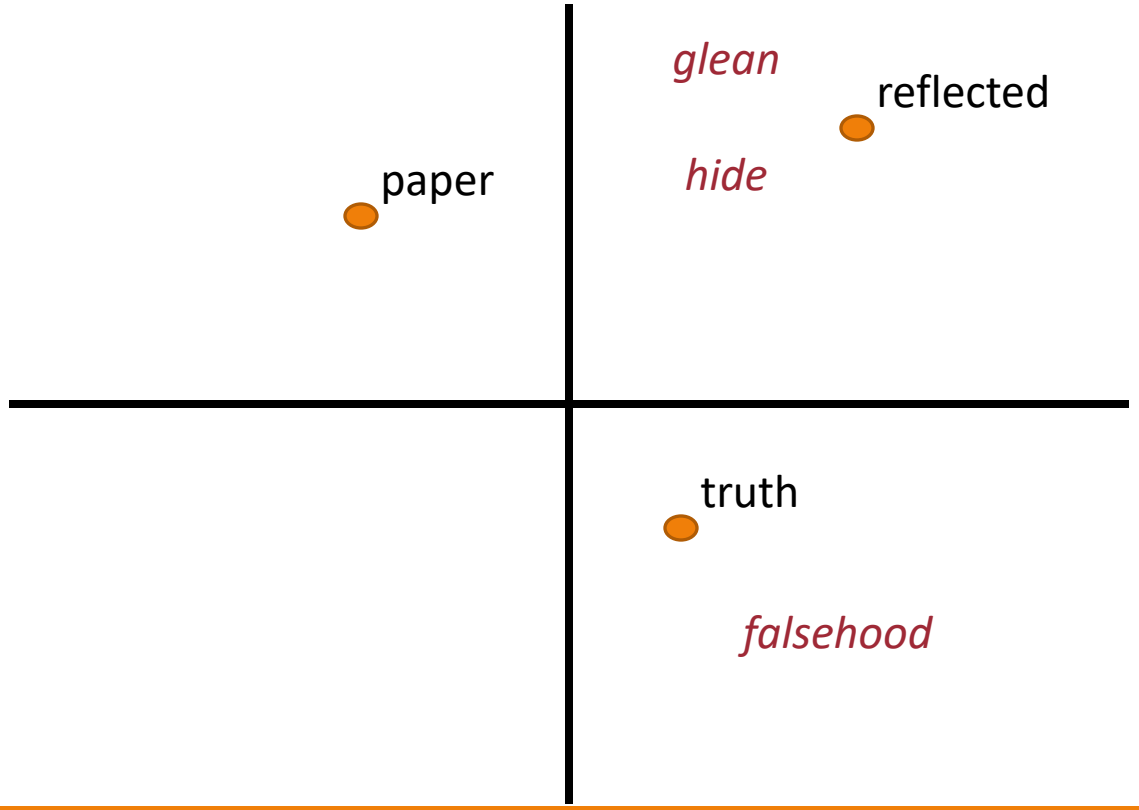
hide

falsehood

where might these go in this space?

Continuous Meaning

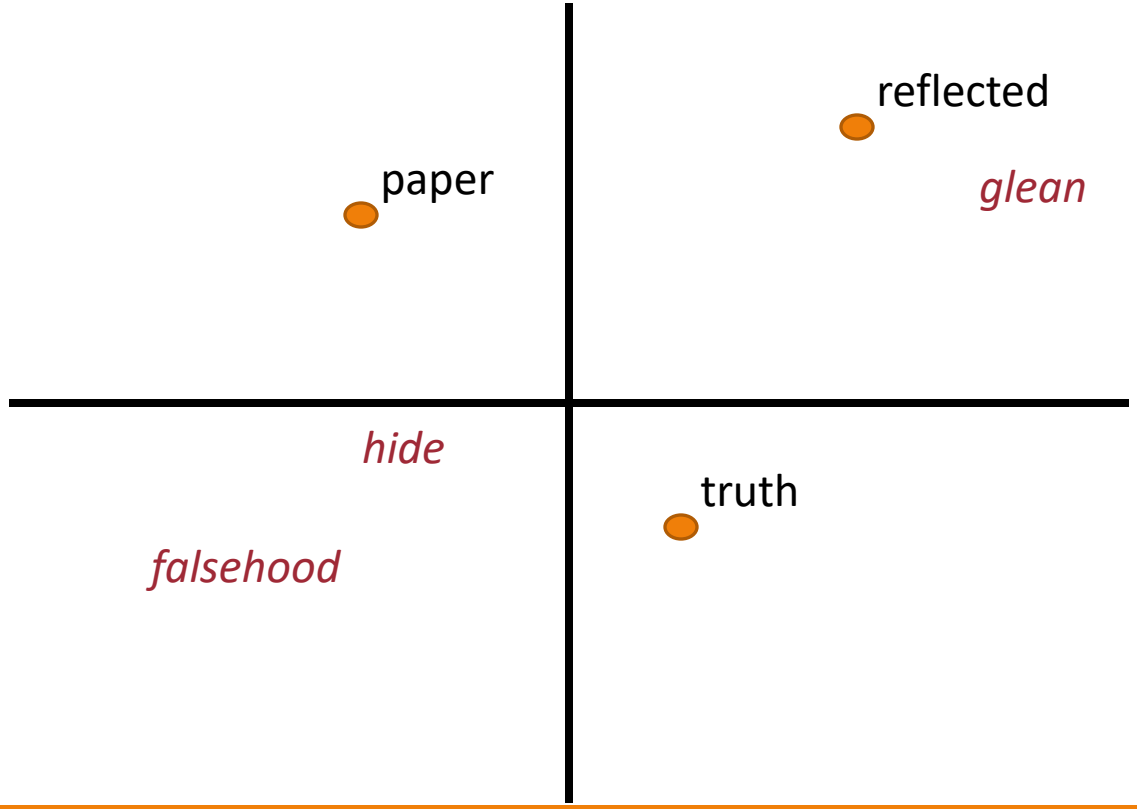
The paper reflected the truth.



One option

Continuous Meaning

The paper reflected the truth.



Another option

(Some) Properties of Embeddings

https://media3.giphy.com/media/3orif0M8U1E7NfpFzq/200_s.gif

Capture "like" (similar) words

target:	Redmond	Havel	ninjutsu	graffiti	capitulate
	Redmond Wash.	Vaclav Havel	ninja	spray paint	capitulation
	Redmond Washington	president Vaclav Havel	martial arts	grafitti	capitulated
	Microsoft	Velvet Revolution	swordsmanship	taggers	capitulating

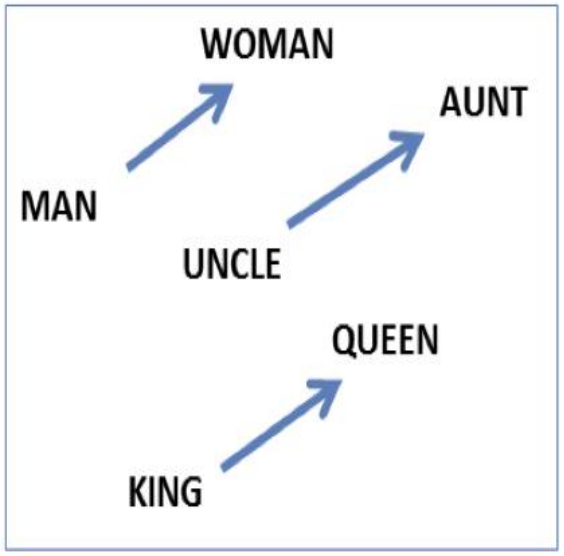
(Some) Properties of Embeddings

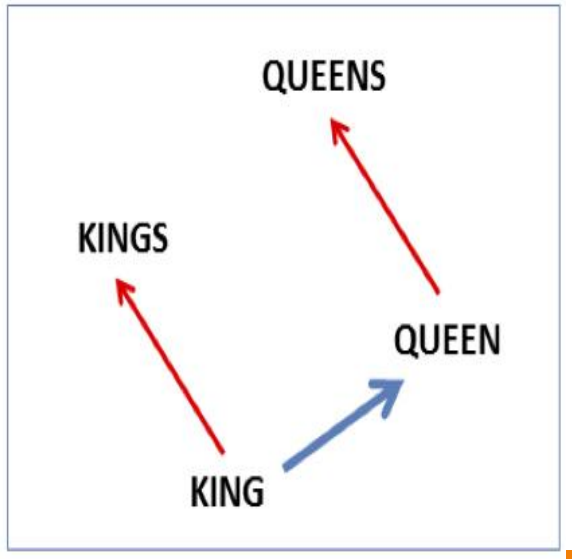
https://media3.giphy.com/media/3orif0M8U1E7NfpFzq/200_s.gif

Capture "like" (similar) words

target:	Redmond	Havel	ninjutsu	graffiti	capitulate
	Redmond Wash.	Vaclav Havel	ninja	spray paint	capitulation
	Redmond Washington	president Vaclav Havel	martial arts	grafitti	capitulated
	Microsoft	Velvet Revolution	swordsmanship	taggers	capitulating

Capture relationships





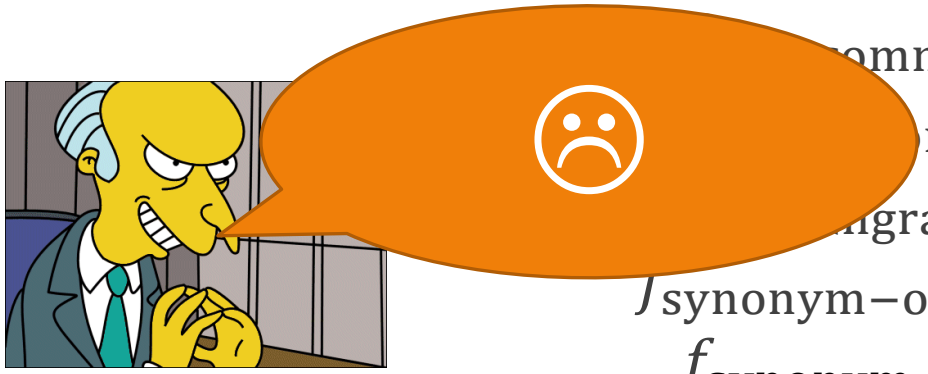
vector(*'king'*) – vector(*'man'*) + vector(*'woman'*) ≈ vector('queen')

vector(*'Paris'*) - vector(*'France'*) + vector(*'Italy'*) ≈ vector('Rome')

Case Study: Plagiarism Detector (Feature Example)

Given two documents x_1, x_2, predict y = 1 (plagiarized) or y = 0 (not plagiarized)

Intuition: documents are more likely to be plagiarized if they have words in common



$f_{\text{ommon-word,Plag.}}(x_1, x_2) = ???$

$f_{\text{rd v>,Plag.}}(x_1, x_2) = ???$

$f_{\text{ngram Z>,Plag.}}(x_1, x_2) = ???$

$f_{\text{synonym-of-<word v>,Plag.}}(x_1, x_2) = ???$

$f_{\text{synonym-of-<ngram Z>,Plag.}}(x_1, x_2) =$

get_similarity_with_embeddings()

Image: http://3.bp.blogspot.com/_MWJDzvrnu7Y/TTcXpCWZerI/AAAAAAAAAWY/zxu1GwrIBz0/s1600/excellent-mr-burns.gif

Think-Pair-Share: Embedding Similarity

https://vectors.nlpl.eu/explore/embeddings/en/

These embeddings are created from Wikipedia. Consider how the words that are similar to them might have been calculated. That is, what articles do you think the words were found in?

Vector Embeddings: Key Ideas

Vector embeddings can be used for phrases, paragraphs, or even whole documents!

1. Acquire basic contextual statistics (often counts) for each word type v

2. Extract a real-valued vector e_v for each word v from those statistics

[0.00315225, 0.00315225, 0.00547597, 0.00741556, 0.00912817, 0.01068435, 0.01212381, 0.01347162, 0.01474487, 0.0159558]

3. Use the vectors to represent each word in later tasks