EE P 500 D: LLMs and ChatGPT

History of LLMs | ChatGPT | Embeddings | Demos | Coding

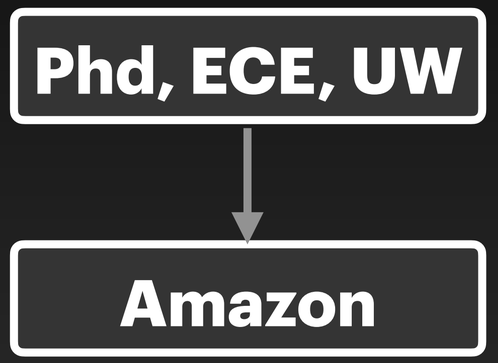
Dr. Karthik Mohan, Nov 11 2023 | LLM Short Course | PMP, ECE

Bit about Me

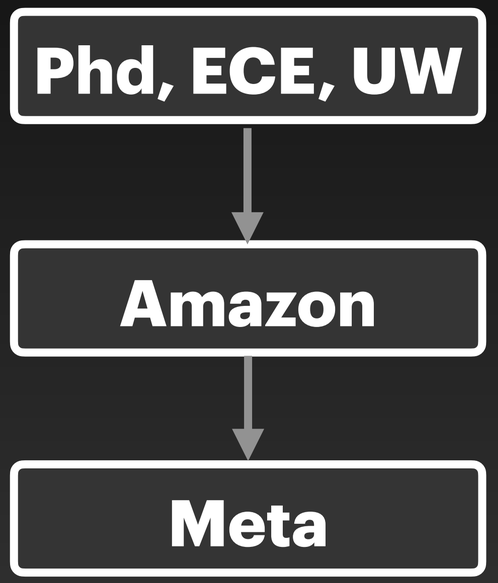
Bit about Me

Phd, ECE, UW

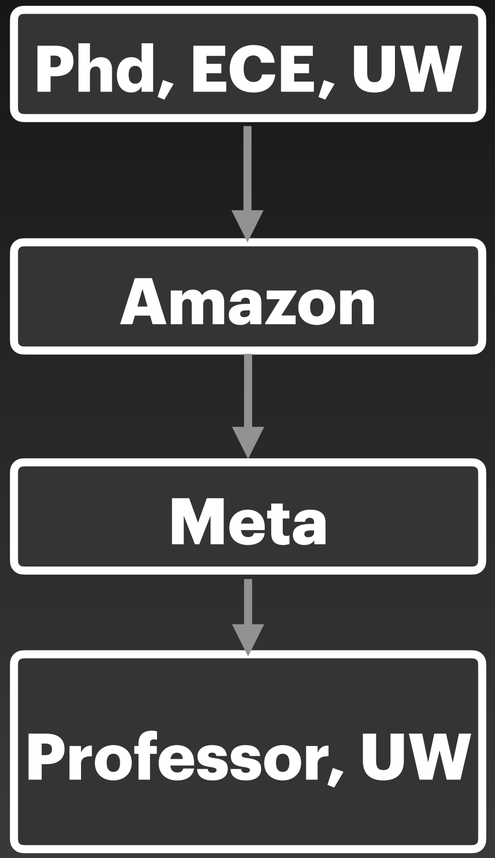
Bit about Me



Bit about Me



Bit about Me



Bit about Me

Phd, ECE, UW

↓

Amazon

↓

Meta

↓

Professor, UW → Staff ML Scientist

Course Outline

November 11

- Logistics and Motivation
- Introduction to LLMs
- Embeddings & Semantic Search

November 12

- LLM Models
- Fine-Tuning LLMs
- Sentiment Analysis
- Prompt Engineering

November 18

- Data Augmentation
- LLMs in production
- Question Answering

November 19

- LLM Ecosystem
- LangChain
- Recap
- Project Presentations

Course Outline

November 11

- Logistics and Motivation
- Introduction to LLMs
- Prompt Engineering Principles

November 12

- LLM Models
- Prompt Engineering
- Fine-Tuning LLMs
- Sentiment Analysis

November 18

- Data Augmentation
- LLMs in production
- Question Answering

November 19

- LLM Ecosystem
- LangChain
- Recap
- Project Presentations

Course Outline

November 11

- Logistics and Motivation
- Introduction to LLMs
- Prompt Engineering Principles

November 12

- LLM Models
- Fine-Tuning LLMs
- Sentiment Analysis
- Building your own chatbot

November 18

- Data Augmentation
- LLMs in production
- Question Answering

November 19

- LLM Ecosystem
- LangChain
- Recap
- Project Presentations

Course Outline

November 11

- Logistics and Motivation
- Introduction to LLMs
- Prompt Engineering Principles

November 12

- LLM Models
- Fine-Tuning LLMs
- Sentiment Analysis
- Building your own chatbot

November 18

- Data Augmentation
- LLMs in production
- Question Answering

November 19

- LLM Ecosystem
- LangChain
- Recap
- Project Presentations

Every Class

First 75 Minutes

- Theory
- Demos

Next 10 minutes

- In-Class Exercise

Next 1.5 hours

- In-class Coding Demo
- In-class Coding Exercise

Every Class

First 75 Minutes

- Theory
- Demos

Next 10 minutes

- In-Class Exercise

Next 1.5 hours

- In-class Coding Demo
- In-class Coding Exercise

Every Class

First 75 Minutes

- Theory
- Demos

Next 10 minutes

- In-Class Exercise

Next 1.5 hours

- In-class Coding Demo
- In-class Coding Exercise

What I would like you to take away!

Conceptually

- Better understanding of LLMs
- Of LLM application areas
- Of APIs

Implementation

- Coding up baselines in Colab
- Comfort with APIs
- Use of Hugging Face models
- Showcasing your work on webpage

Ideas

- Where can you apply LLMs next?
- ?

What I would like you to take away!

Conceptually

- Better understanding of LLMs
- Of LLM application areas
- Of APIs

Implementation

- Coding up baselines in Colab
- Comfort with APIs
- Fine-Tuning Hugging Face models
- Showcasing your work on webpage

Ideas

- Where can you apply LLMs next?
- ?

What I would like you to take away!

Conceptually

- Better understanding of LLMs
- Of LLM application areas
- Of APIs

Implementation

- Coding up baselines in Colab
- Comfort with APIs
- Use of Hugging Face models
- Showcasing your work on webpage

Ideas

- Where can you apply LLMs next?
- ?

What are you looking to learn/work on ?

Groups of 3 and connect for a few!

Course Webpage and Resources

https://bytesizeml.github.io/llm_short_course/

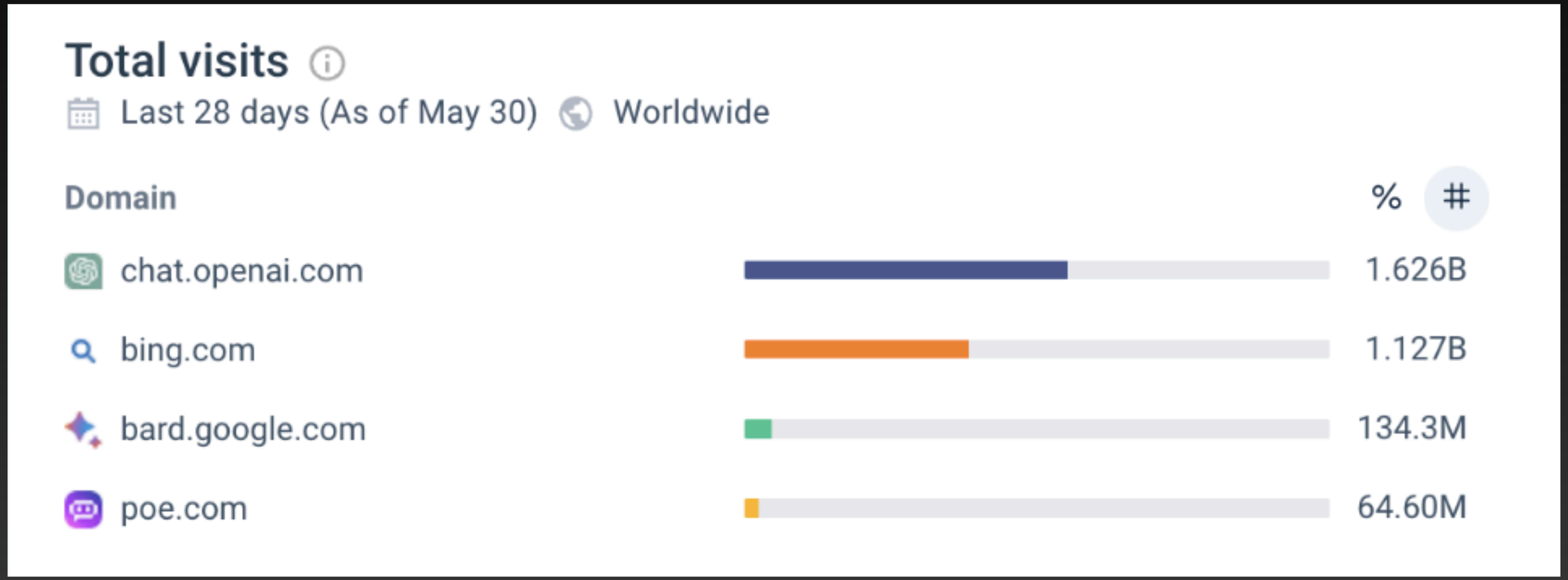
Assignments

Deadline	Assignment	Description
November 11th	Assignment 0	Prep, set up and getting hands-on
		with language models plus
		work a simple demo
		Example of a simple demo
November 18th	Conceptual	Test your understanding
	assignment	of the concepts and theory
		behind LLMs
November 18th	Mini-Project	Use of Chat GPT, LLMs
		on sentiment extraction or
		chat-bot simulation with a
		working demo hosted on a webpage
November 19th	Mini-Project	8 minutes per team
	Presentation	

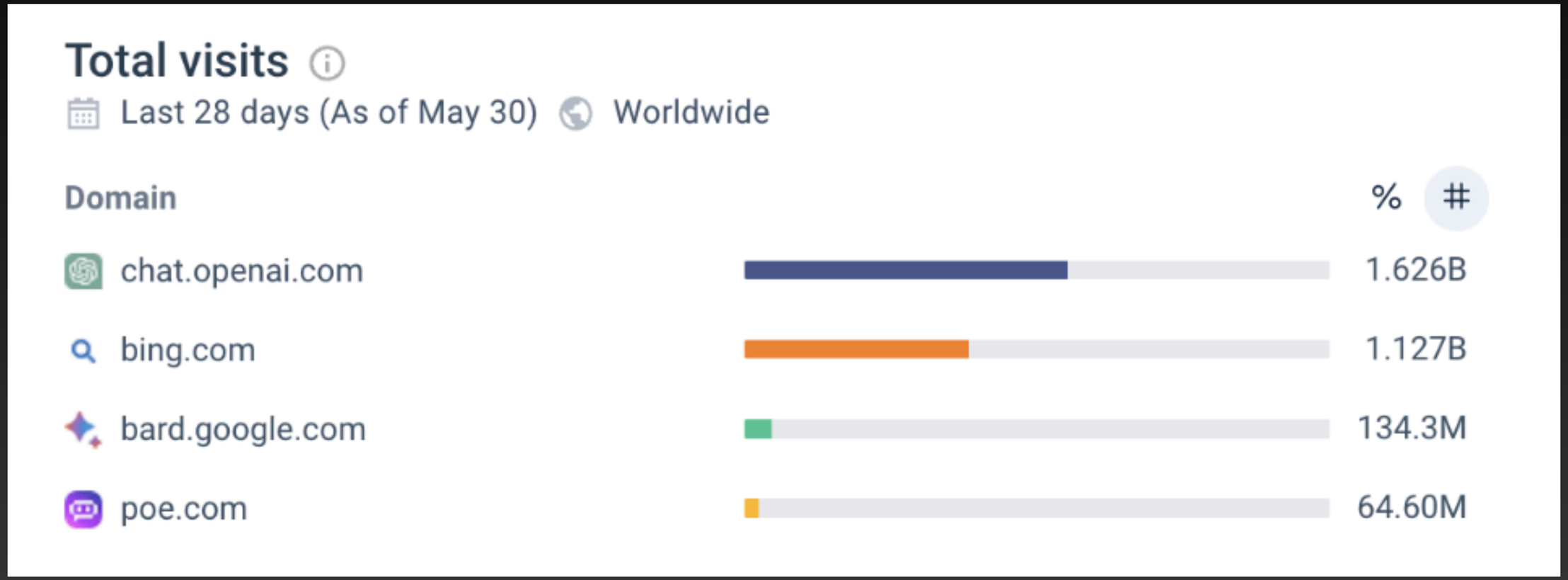
ChatGPT and LLMs are everywhere!

ChatGPT and LLMs are everywhere!

ChatGPT and LLMs are everywhere!



ChatGPT and LLMs are everywhere!



ChatGPT and LLMs are everywhere!

Let’s look at some examples!

ChatGPT and LLMs are everywhere!

Paraphrasing

ChatGPT and LLMs are everywhere!

Paraphrasing

Math

ChatGPT and LLMs are everywhere!

Paraphrasing

Math

Coding

ChatGPT and LLMs are everywhere!

Let’s go checkout ChatGPT live!

Engine behind ChatGPT

ChatGPT heavily relies on Large Language Models to power its responses to users!

Engine vs API

Engines are different from APIs and we shouldn’t confuse the two.

Engine vs API

Engines are different from APIs and we shouldn’t confuse the two.

BERT and Llama are Engines/Foundation Models whereas ChatGPT 3.5 is an API

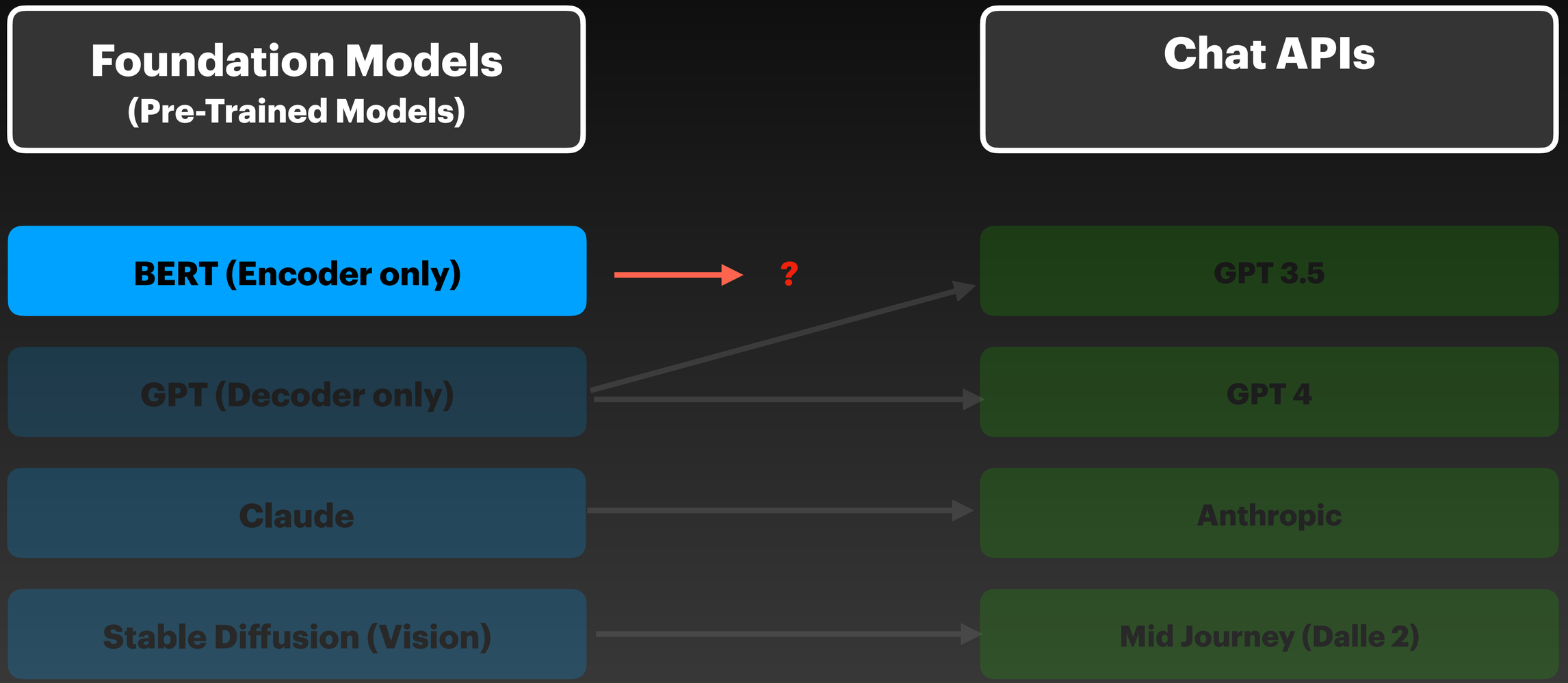
Engine vs API

Foundation Models
(Pre-Trained Models)

- BERT (Encoder only)
- GPT (Decoder only)
- Claude
- Stable Diffusion (Vision)

Chat APIs

Engine vs API



Engine vs API

Foundation Models
(Pre-Trained Models)

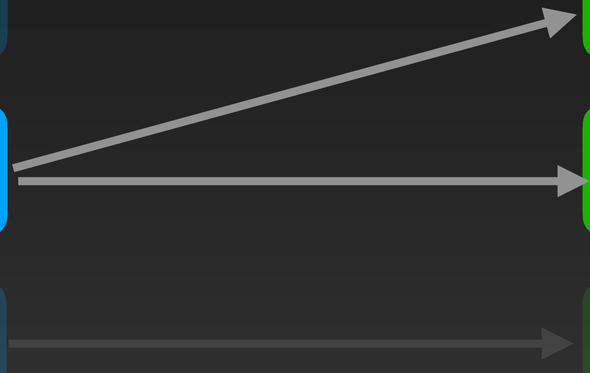
Chat APIs

BERT (Encoder only)

GPT (Decoder only)

Claude

Stable Diffusion (Vision)



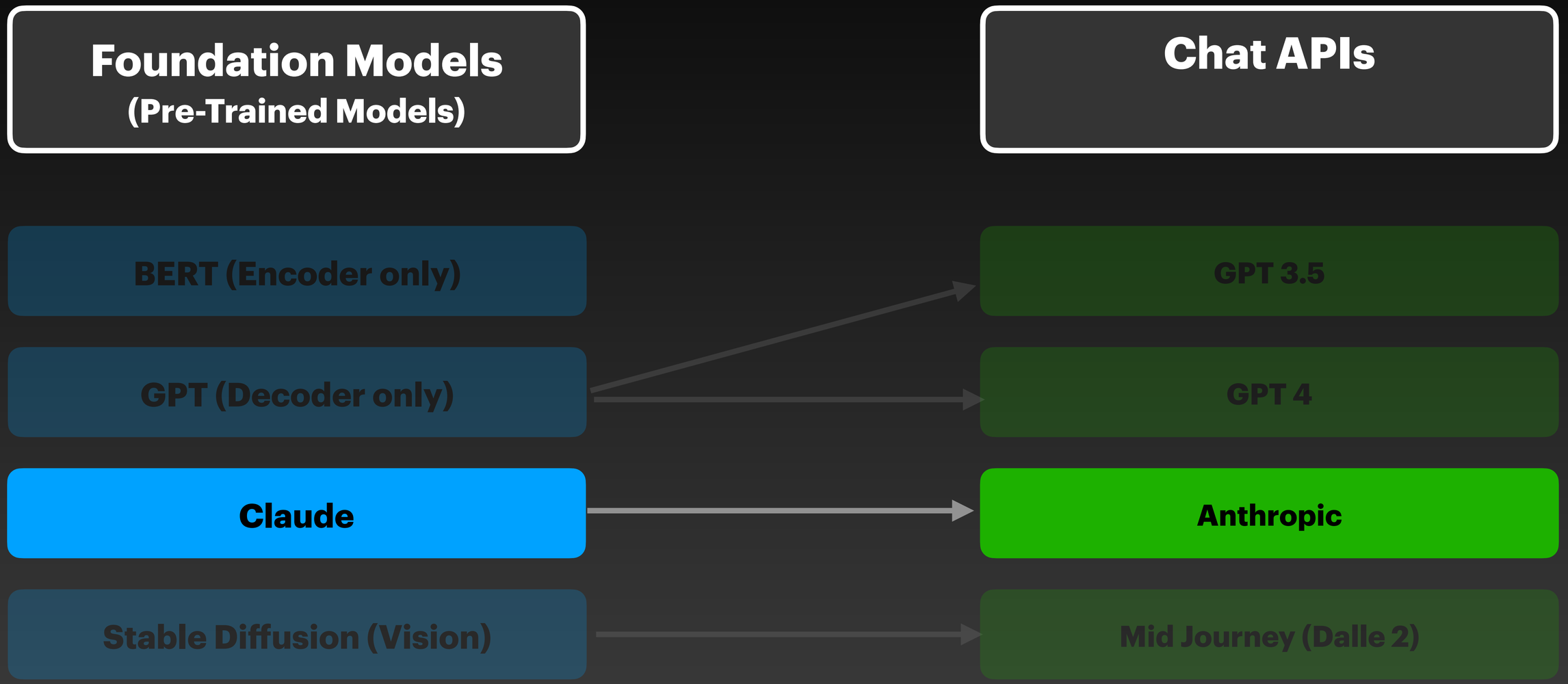
GPT 3.5

GPT 4

Anthropic

Mid Journey (Dalle 2)

Engine vs API



Engine vs API

Foundation Models
(Pre-Trained Models)

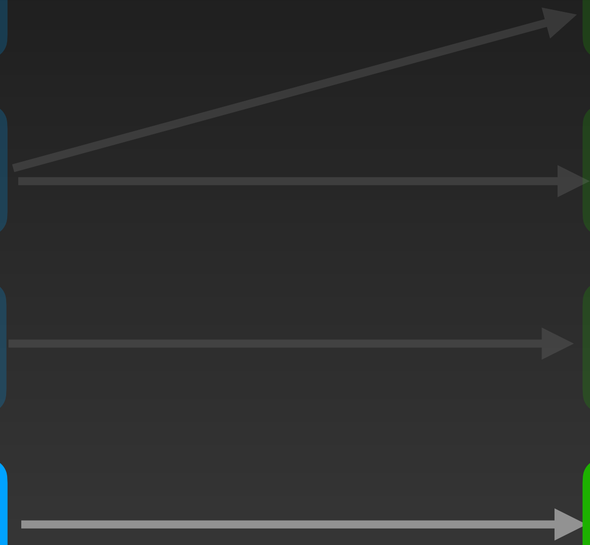
Chat APIs

BERT (Encoder only)

GPT (Decoder only)

Claude

Stable Diffusion (Vision)



GPT 3.5

GPT 4

Anthropic

Mid Journey (Dalle 2)

Engine vs API

Foundation Models
(Pre-Trained Models)

Chat APIs

BERT (Encoder only)

GPT (Decoder only)

Large Vision Model
Pre-Trained Model
Foundation Model

GPT 3.5

GPT 4

Claude

Anthropic

Stable Diffusion (Vision)

Mid Journey (Dalle 2)

Engine vs API

Foundation Models
(Pre-Trained Models)

t APIs

LLM - Large Language Model
Pre-Trained Model
Foundation Model

BERT (Encoder only)

GPT (Decoder only)

Claude

Stable Diffusion (Vision)

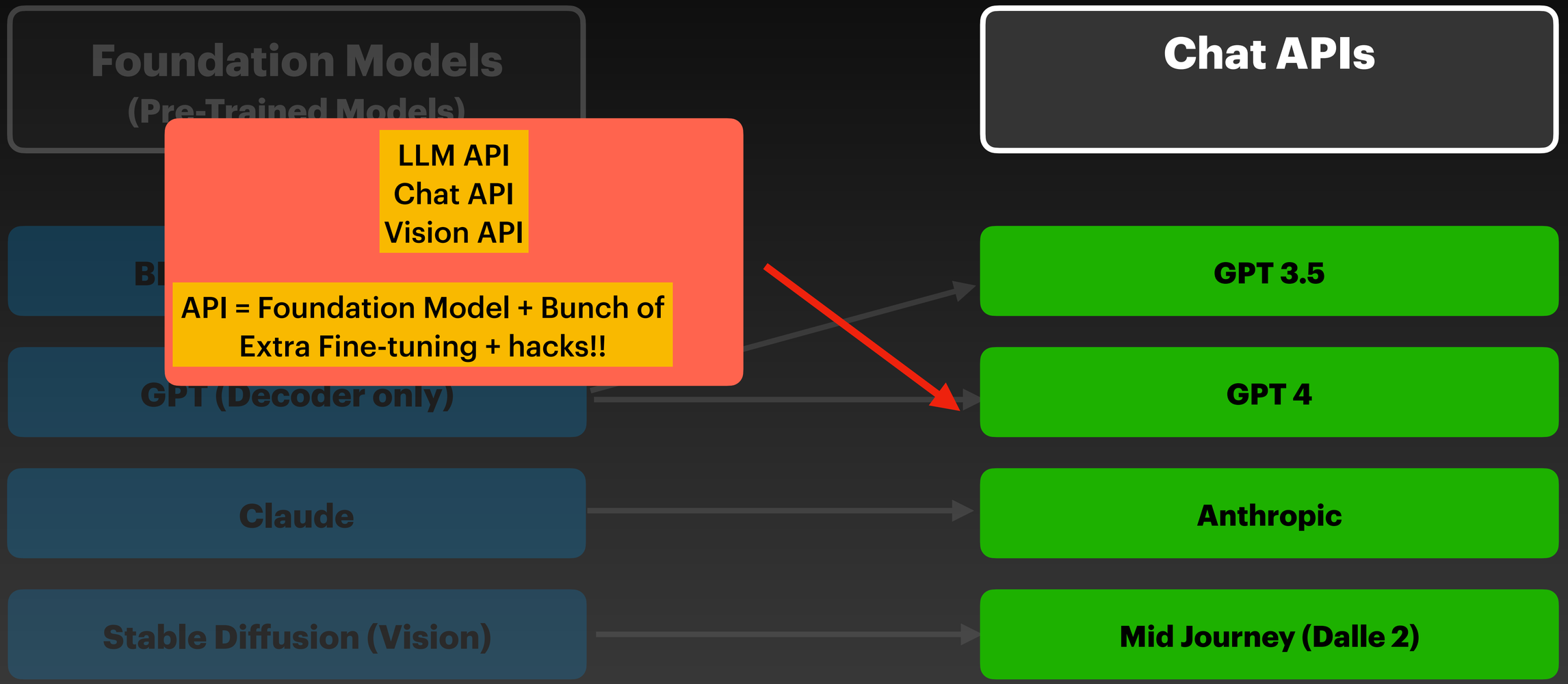
GPT 3.5

GPT 4

Anthropic

Mid Journey (Dalle 2)

Engine vs API



What is a Language Model?

Scientific Data-driven Model that helps machines understand language and patterns in sentence construction

What is a Language Model?

Example: I just got promoted. I am feeling so ———

What is a Language Model?

Example: I just got promoted. I am feeling so happy

What is a Language Model?

Example: I just checked my application status and it got ———. It’s frustrating!

What is a Language Model?

Example: I just checked my application status and it got rejected. It's frustrating!

What is a Large Language Model (LLM)?

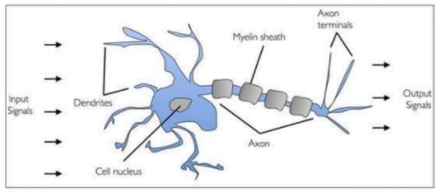
LLMs are language models that are learned from massive corpuses of text, that are mined from the web. They are known to be sophisticated in understanding language and can be **generative** in nature.

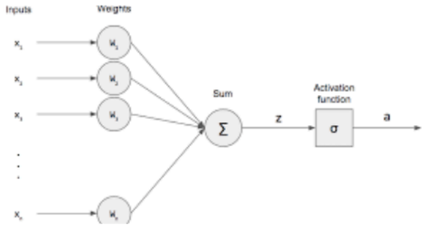
History of (Large) Language Models

How did machines work with language before and how we do it now?

History of (Large) Language Models

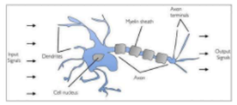
Perceptron



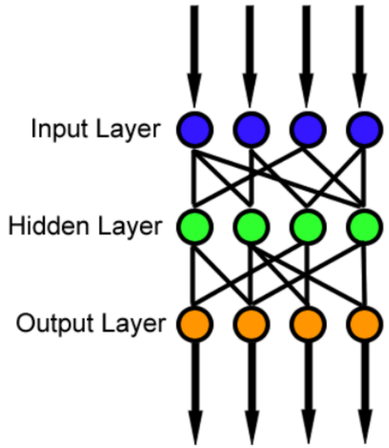


History of (Large) Language Models

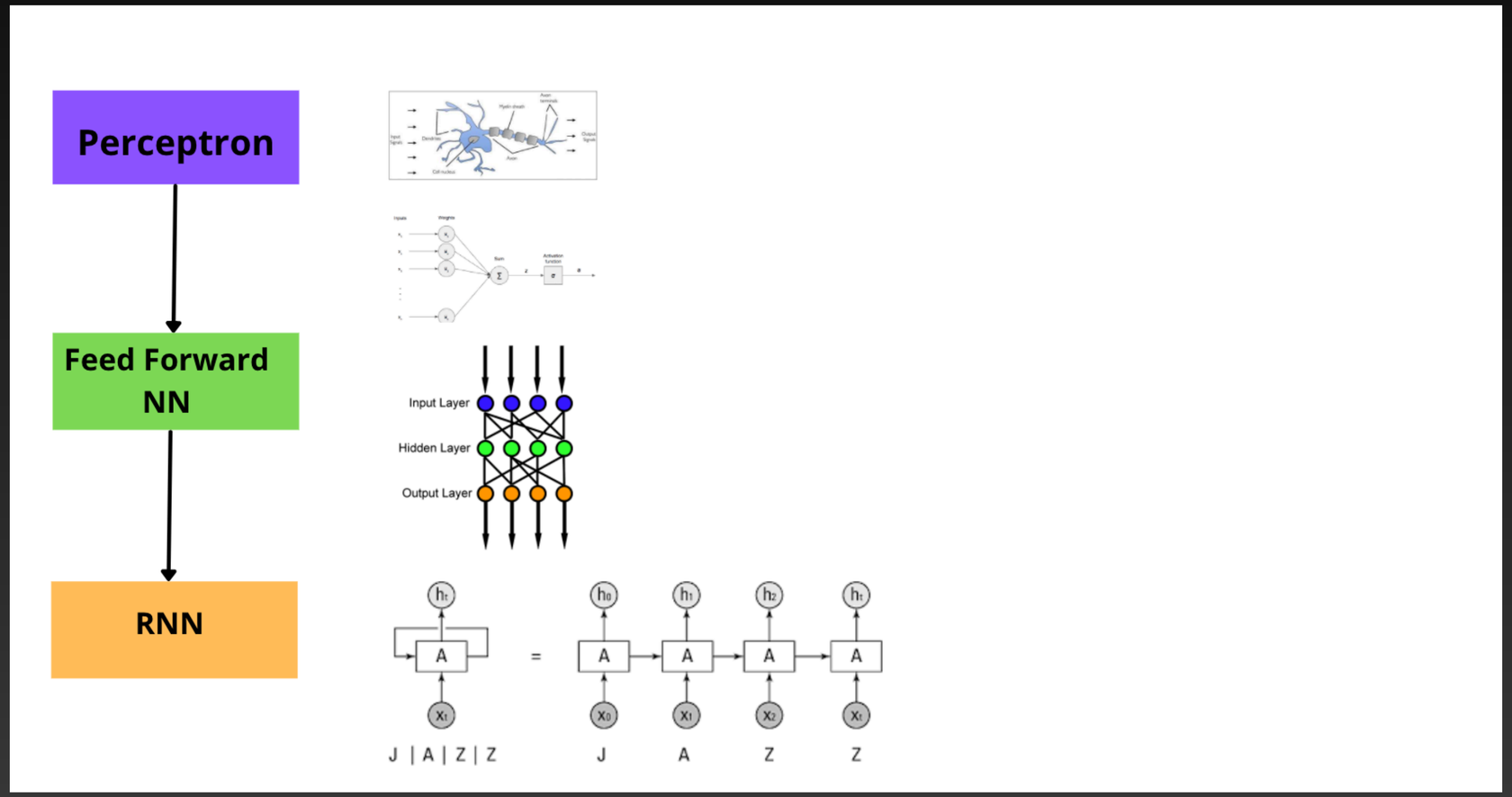
Perceptron



Feed Forward NN



History of (Large) Language Models



History of (Large) Language Models

RNN Issue:

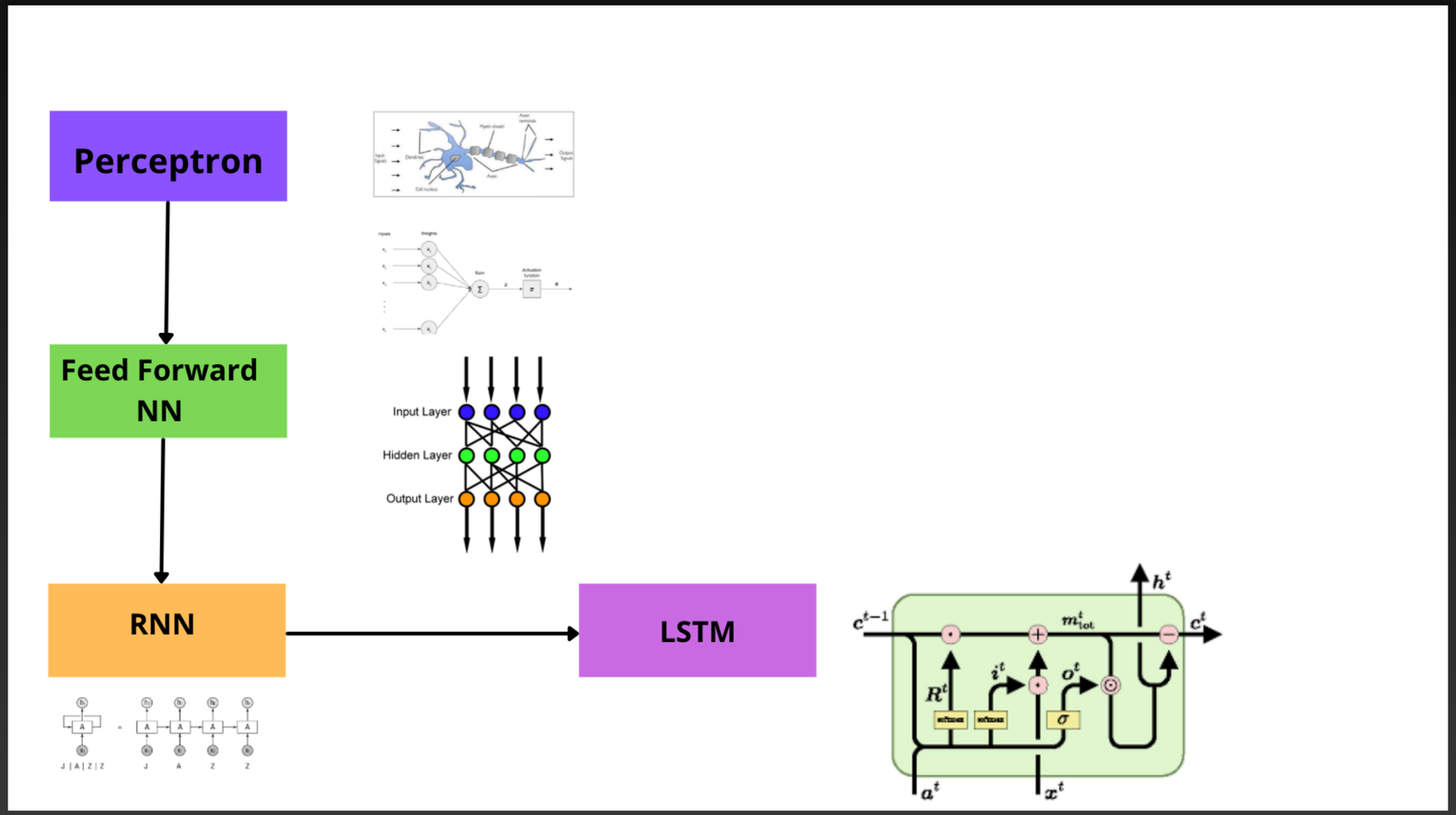
I just arrived in NY. In a few days, I would like to visit the city, ———

History of (Large) Language Models

RNN Issue:

I just arrived in NY. In a few days, I would like to visit the city, NY

History of (Large) Language Models



History of (Large) Language Models

LSTM

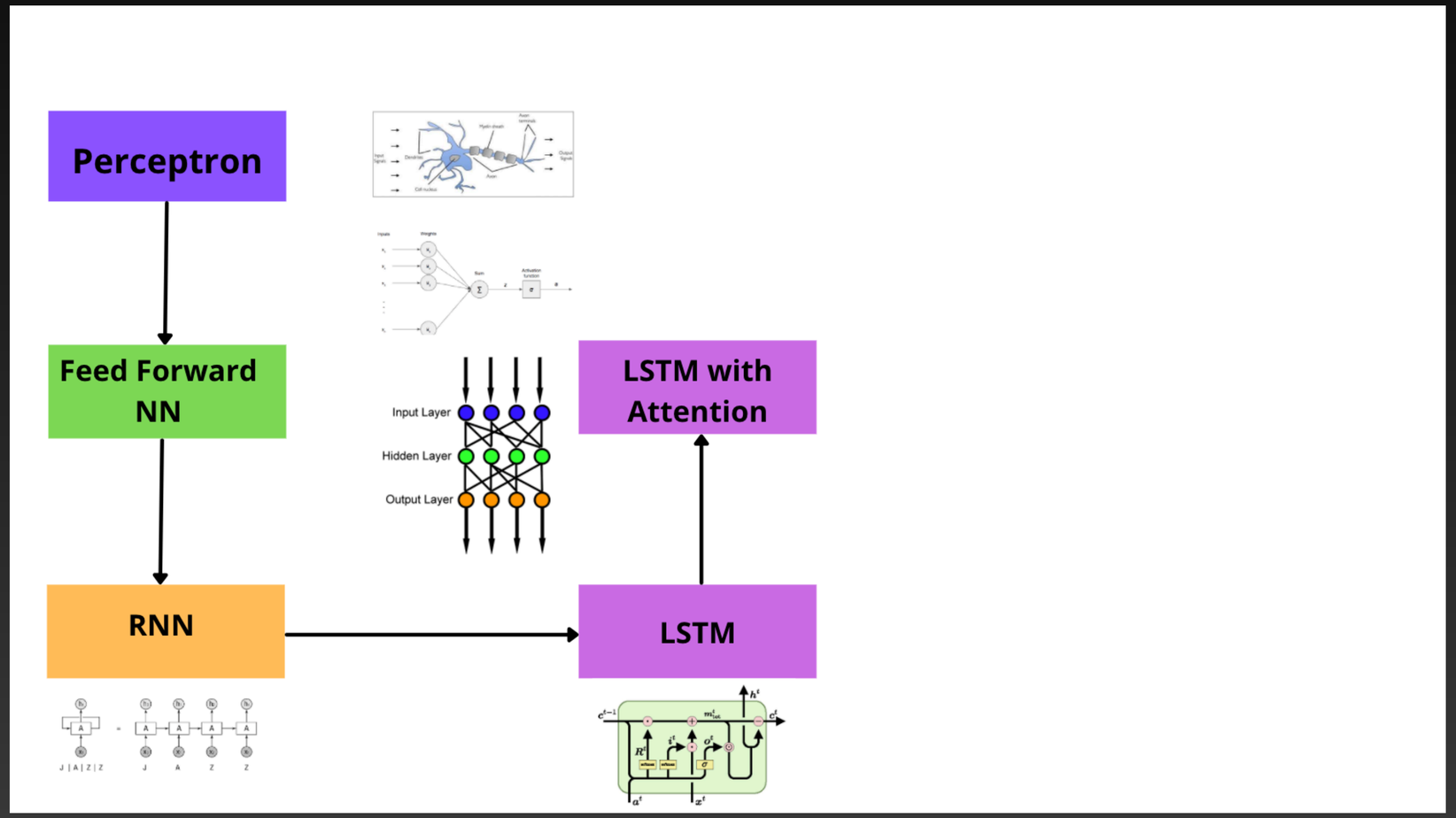
I just arrived in NY. In a few days, I would like to visit the city, ———

History of (Large) Language Models

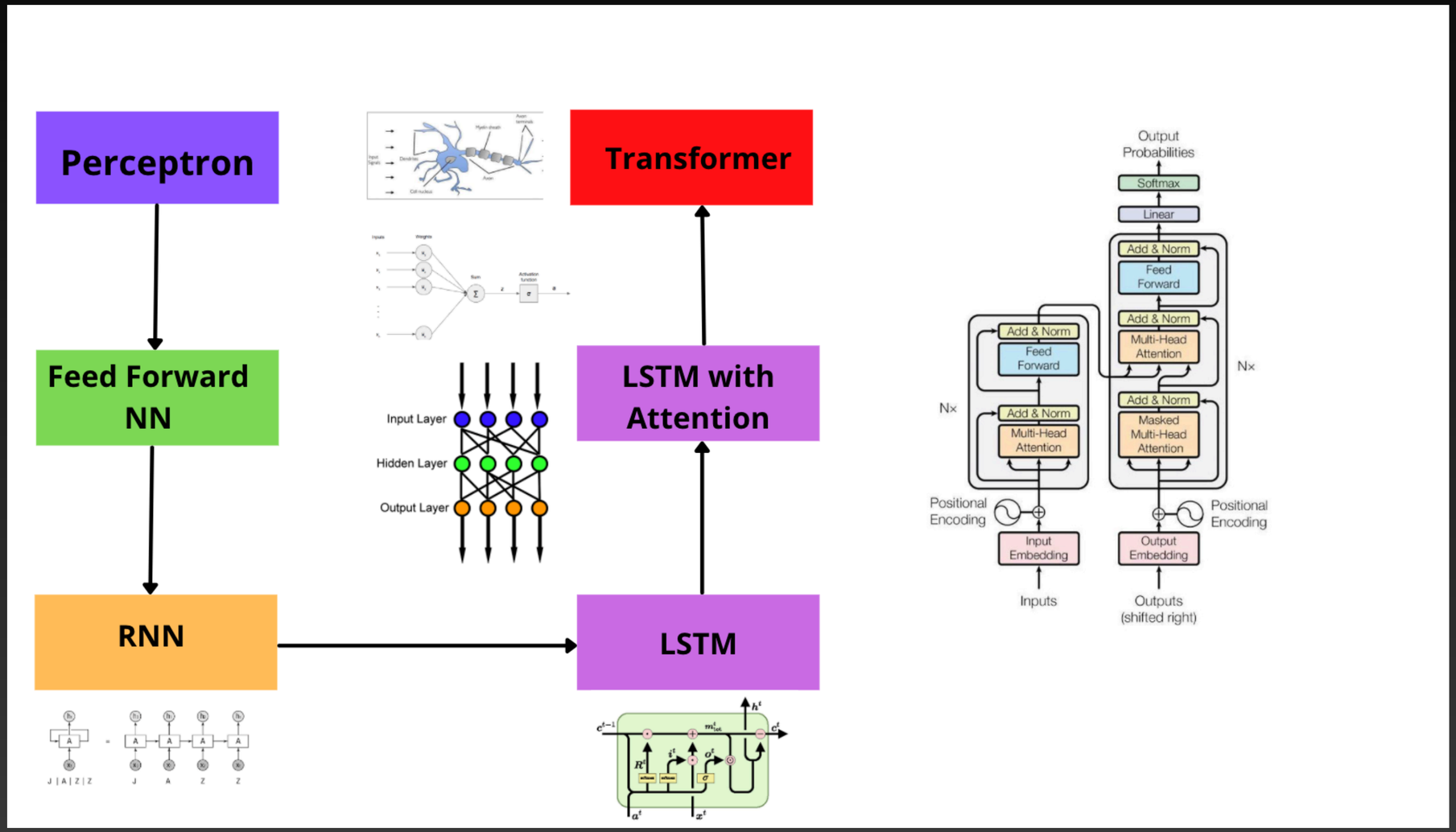
LSTM

I just arrived in NY. In a few days, I would like to visit the city, Seattle

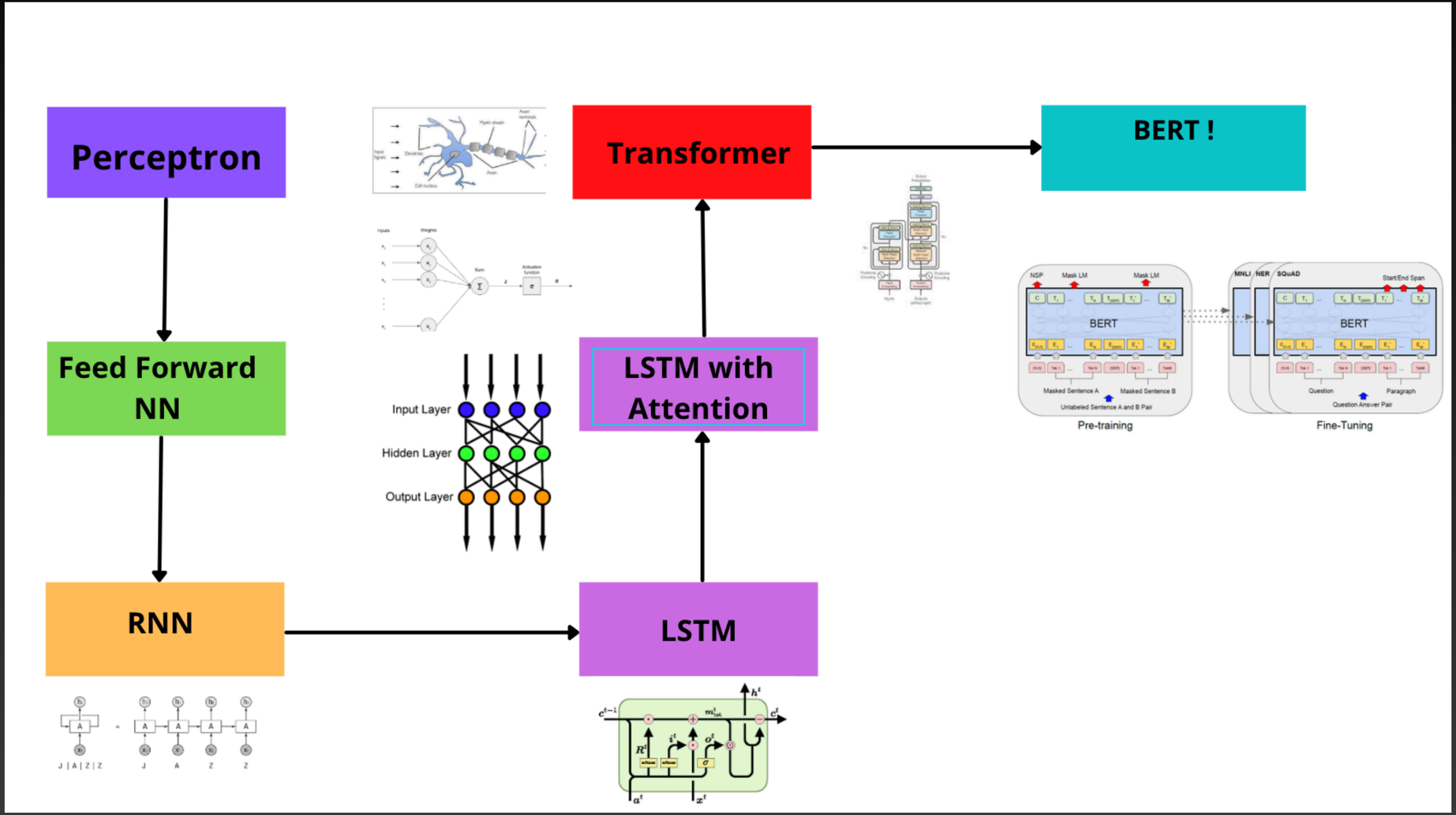
History of (Large) Language Models



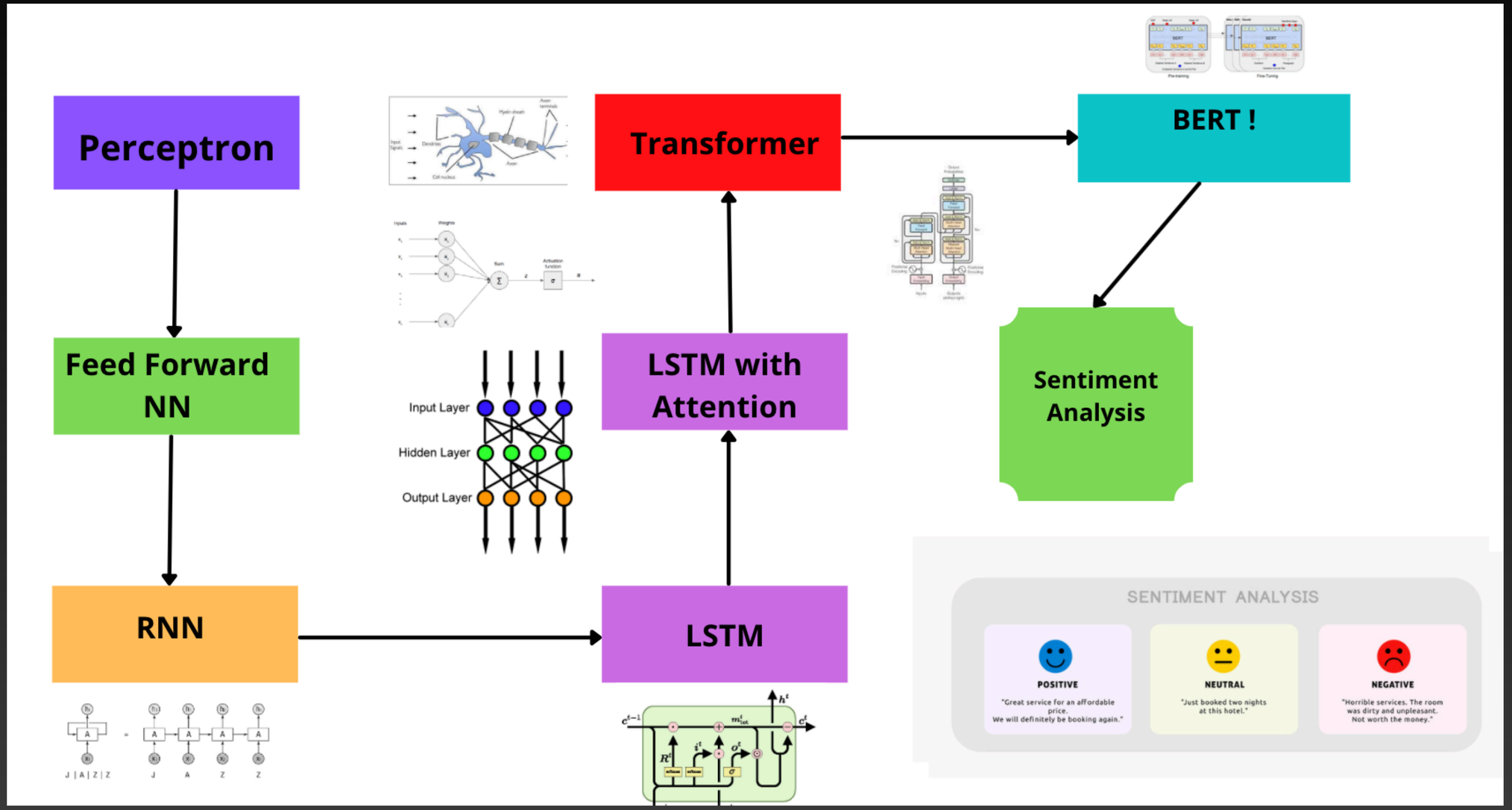
History of (Large) Language Models



History of (Large) Language Models



History of (Large) Language Models



History of (Large) Language Models

Perceptron

Feed Forward NN

RNN

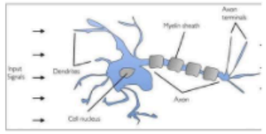
LSTM

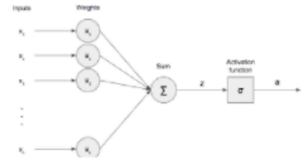
LSTM with Attention

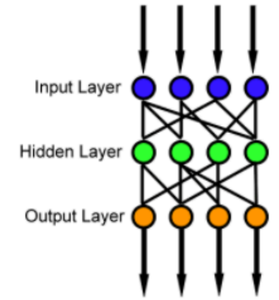
Transformer

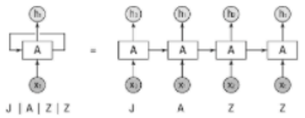
BERT !

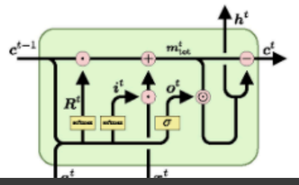
Extractive Summarization

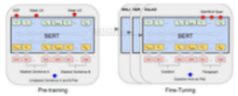


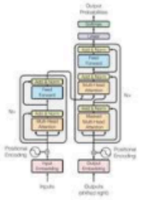


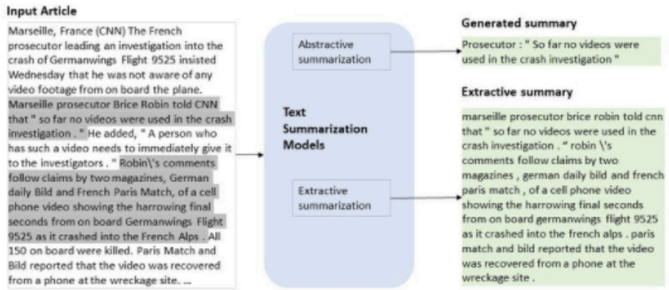




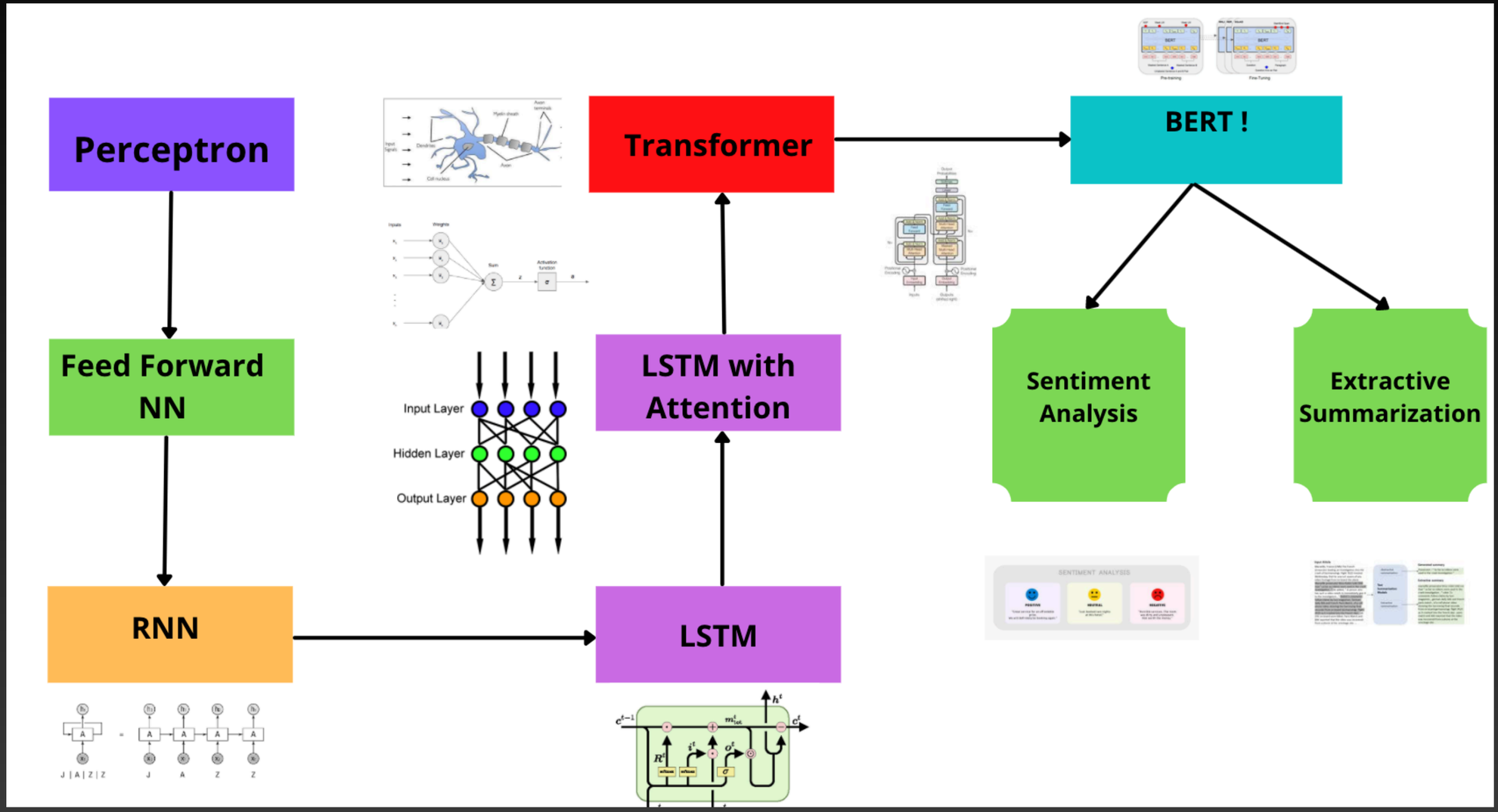








History of (Large) Language Models



History of (Large) Language Models

GPT vs BERT

While BERT is purely about encoding and is called an encoding Transformer. GPT is purely a decoder and is called a decoding transformer.

History of (Large) Language Models

GPT-x

GPT-x (GPT, GPT-2, GPT-2.5, etc) are decoding transformers that are trained to predict the next token given the past and do a very good job at it! That's how they can generate entire paragraphs that look logical, grammatical and structured.

ChatGPT Framework

ChatGPT Framework

Pre-Trained LLM (GPT-x)

ChatGPT Framework

Pre-Trained LLM (GPT-x)

+

ChatGPT Framework

Pre-Trained LLM (GPT-x)

+

Supervised Fine-Tuning on High Quality Data

ChatGPT Framework

Pre-Trained LLM (GPT-x)

+

Supervised Fine-Tuning on High Quality Data

+

ChatGPT Framework

Pre-Trained LLM (GPT-x)

+

Supervised Fine-Tuning on High Quality Data

+

Reinforcement Learning With Human Feedback

ChatGPT Framework

Pre-Trained LLM (GPT-x)

+

Supervised Fine-Tuning on High Quality Data

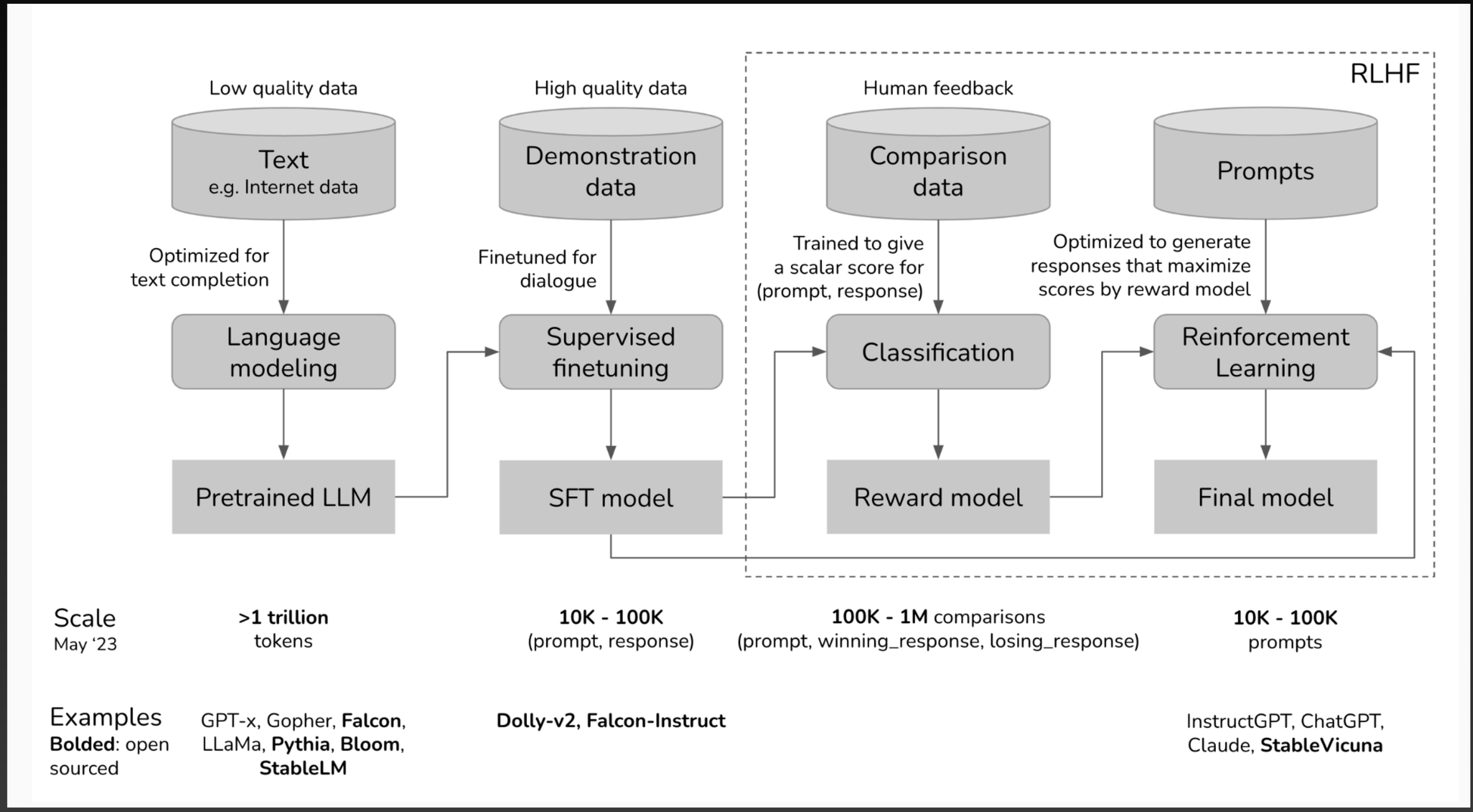
+

Reinforcement Learning With Human Feedback

=

ChatGPT

ChatGPT Framework



1 Trillion Tokens!

	RedPajama	LLaMA*
CommonCrawl	878 billion	852 billion
C4	175 billion	190 billion
Github	59 billion	100 billion
Books	26 billion	25 billion
ArXiv	28 billion	33 billion
Wikipedia	24 billion	25 billion
StackExchange	20 billion	27 billion
Total	1.2 trillion	1.25 trillion

1 Trillion Tokens requires how many books?

	RedPajama	LLaMA*
CommonCrawl	878 billion	852 billion
C4	175 billion	190 billion
Github	59 billion	100 billion
Books	26 billion	25 billion
ArXiv	28 billion	33 billion
Wikipedia	24 billion	25 billion
StackExchange	20 billion	27 billion
Total	1.2 trillion	1.25 trillion

1 Trillion Tokens requires how many books?

	RedPajama	LLaMA*
CommonCrawl	878 billion	852 billion
C4	175 billion	190 billion
Github	59 billion	100 billion
Books	26 billion	25 billion
ArXiv	28 billion	33 billion
Wikipedia	24 billion	25 billion
StackExchange	20 billion	27 billion
Total	1.2 trillion	1.25 trillion

1 Book ~ 50k Tokens

1 Trillion Tokens requires how many books?

	RedPajama	LLaMA*
CommonCrawl	878 billion	852 billion
C4	175 billion	190 billion
Github	59 billion	100 billion
Books	26 billion	25 billion
ArXiv	28 billion	33 billion
Wikipedia	24 billion	25 billion
StackExchange	20 billion	27 billion
Total	1.2 trillion	1.25 trillion

1 Book ~ 50k Tokens

15 Million Books ~ 1 Trillion Tokens

ChatGPT use cases for NLP

ChatGPT use cases for NLP

Table 1: Distribution of use case categories from our API prompt dataset.

Use-case	(%)
Generation	45.6%
Open QA	12.4%
Brainstorming	11.2%
Chat	8.4%
Rewrite	6.6%
Summarization	4.2%
Classification	3.5%
Other	3.5%
Closed QA	2.6%
Extract	1.9%

Table 2: Illustrative prompts from our API prompt dataset. These are fictional examples inspired by real usage—see more examples in Appendix A.2.1.

Use-case	Prompt
Brainstorming	List five ideas for how to regain enthusiasm for my career
Generation	Write a short story where a bear goes to the beach, makes friends with a seal, and then returns home.
Rewrite	This is the summary of a Broadway play: """ {summary} """ This is the outline of the commercial for that play: """

The distribution of prompts used to finetune InstructGPT

ChatGPT use cases for NLP

Prompt Engineering for information retrieval

Data Augmentation

Transfer Learning to smaller models

ChatGPT use cases for NLP

Prompt Engineering for information retrieval

Data Augmentation

Transfer Learning to smaller models

ChatGPT use cases for NLP

Prompt Engineering for information retrieval

Data Augmentation

Transfer Learning to smaller models

ChatGPT use cases for NLP

Prompt Engineering for information retrieval

Data Augmentation

Transfer Learning to smaller models

ChatGPT use cases for NLP

Prompt Engineering for information retrieval

Data Augmentation

Transfer Learning to smaller models

More use cases!

ChatGPT use cases for NLP

Prompt Engineering for information retrieval

Data Augmentation

Transfer Learning to smaller models

Open AI embeddings for Semantic Search

Today we focus on Embeddings

Tomorrow - We move to prompt engineering and fine-tuning LLMs

Embeddings for Semantic Search

Men in Black

Arrival

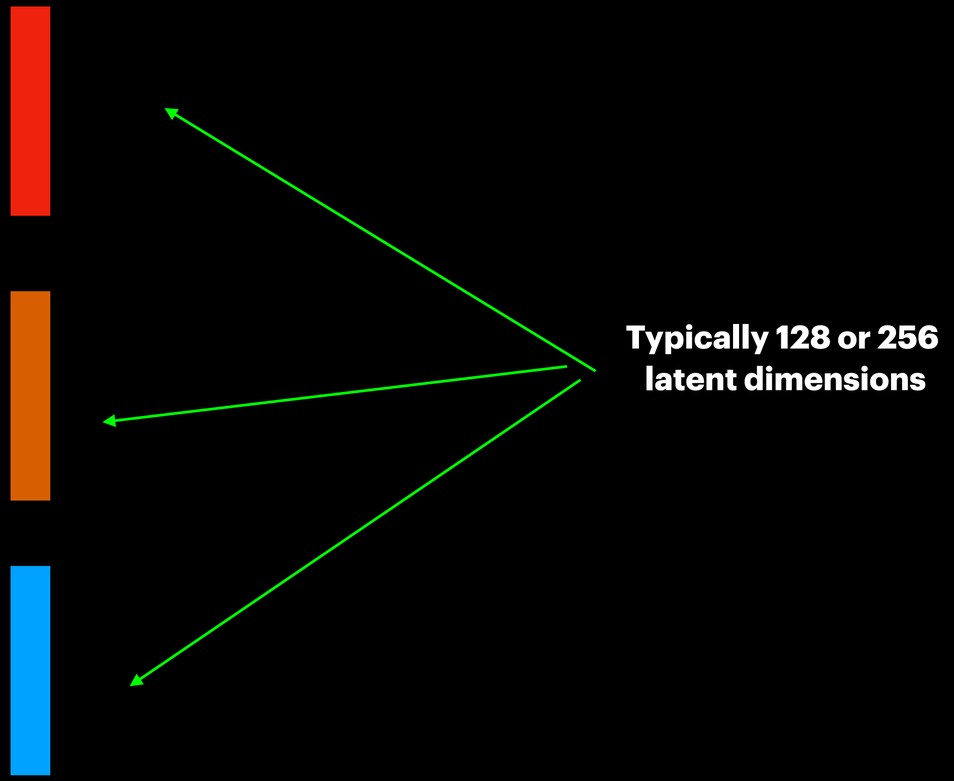
When Harry met

Embeddings for Semantic Search

Men in Black

Arrival

When Harry met



Embeddings for Semantic Search

Men in Black

Arrival

When Harry met

Embeddings

Latent Dimensions → **Interpretation**

? ? ? ? ⋮

- Comedy + Romance
- Thriller + Adventure
- Comedy + Thriller
- Adventure

⋮

Embeddings

Minions

Men in Black

Arrival

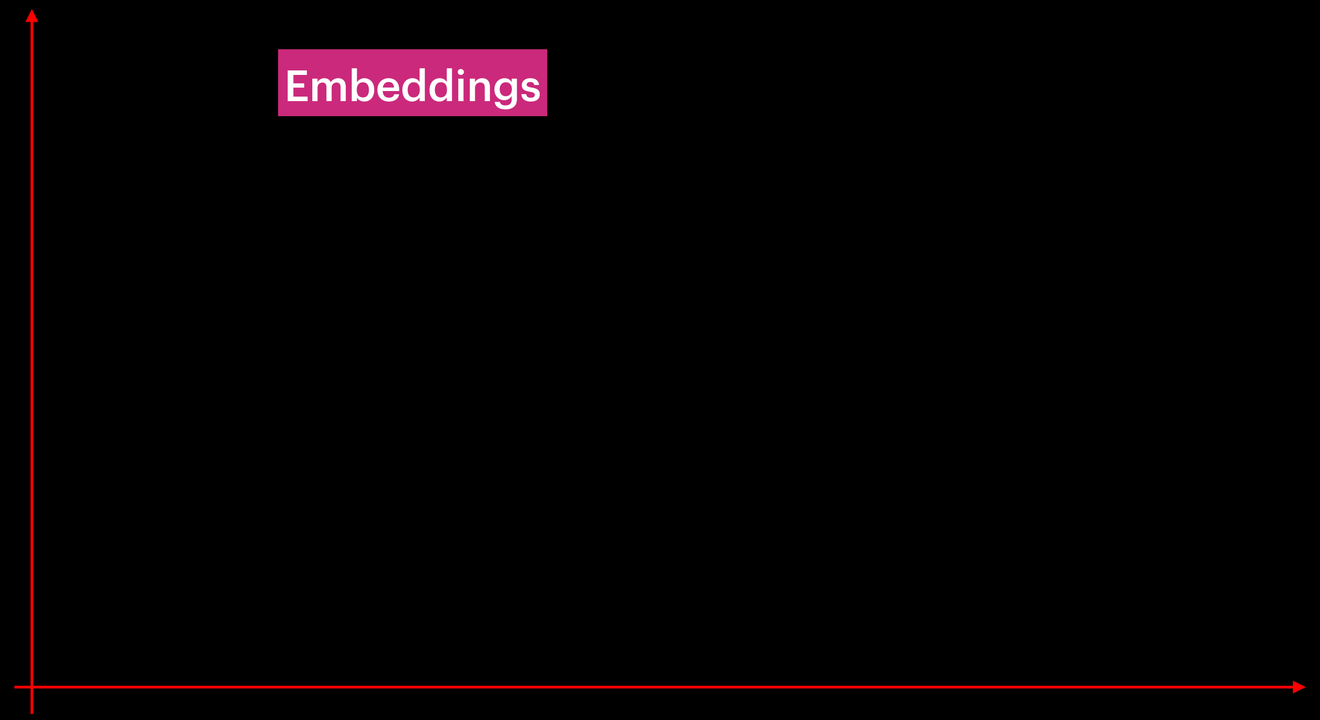
When Harry met

Men in Black

Arrival

When Harry met

Embeddings for Words



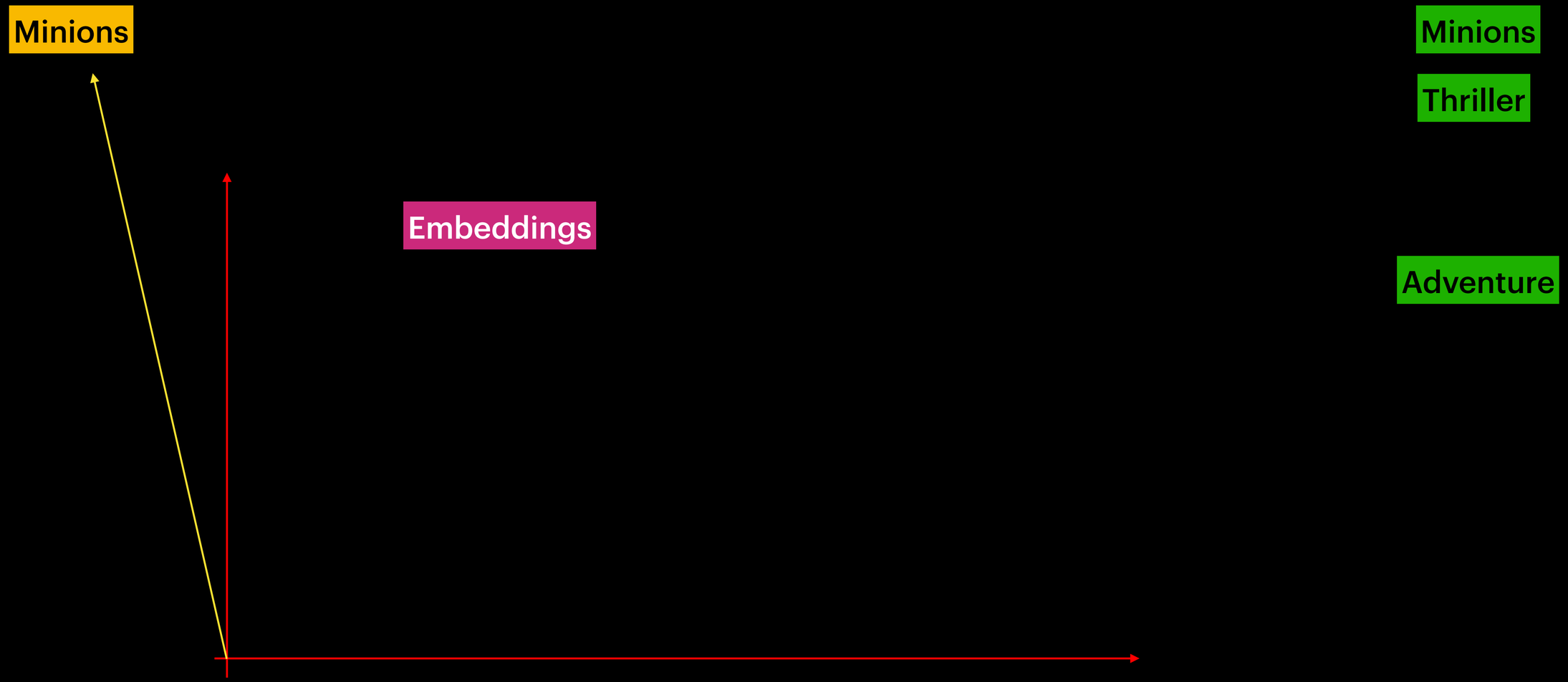
Minions

Thriller

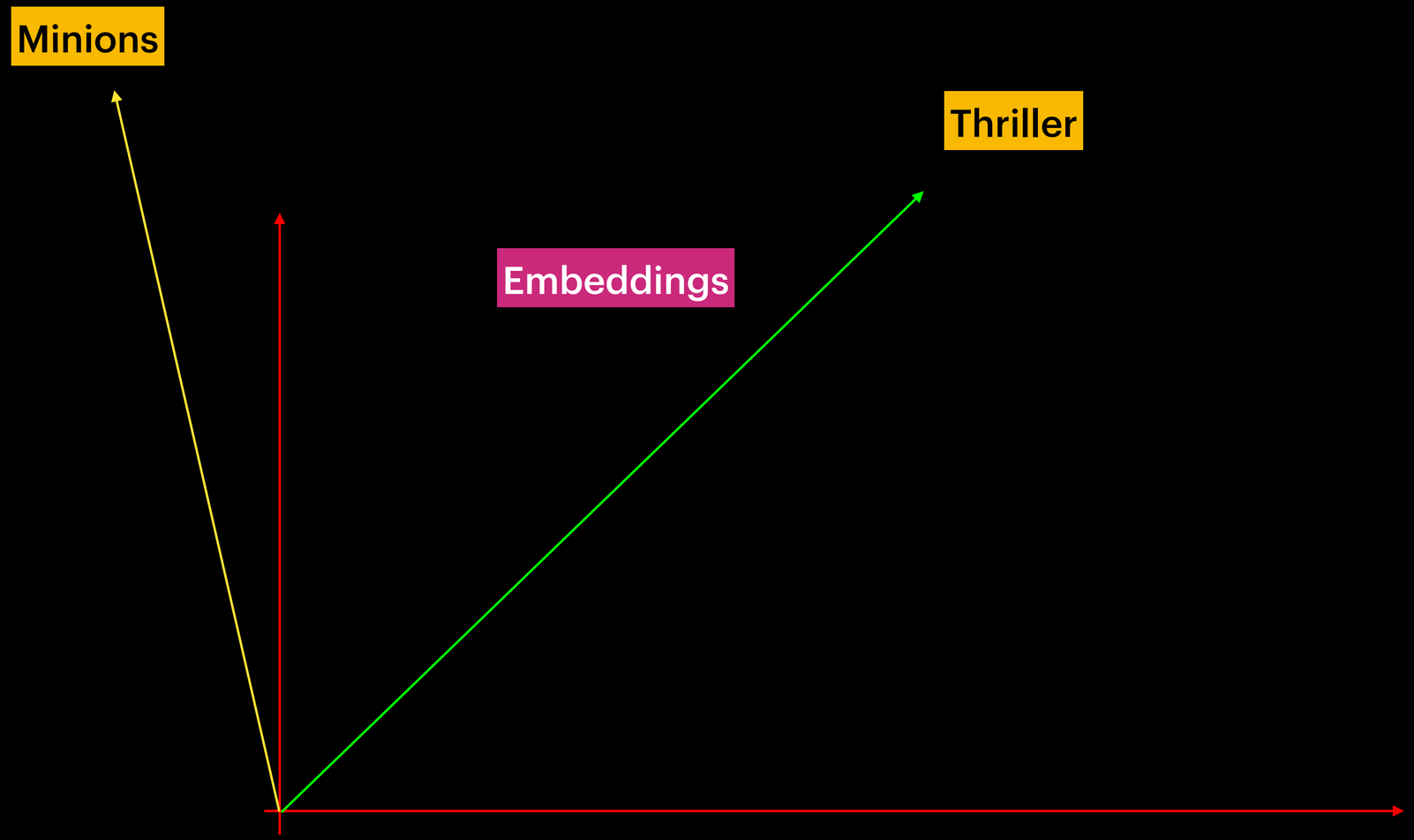
Adventure

Romance

Embeddings for Words



Embeddings for Words



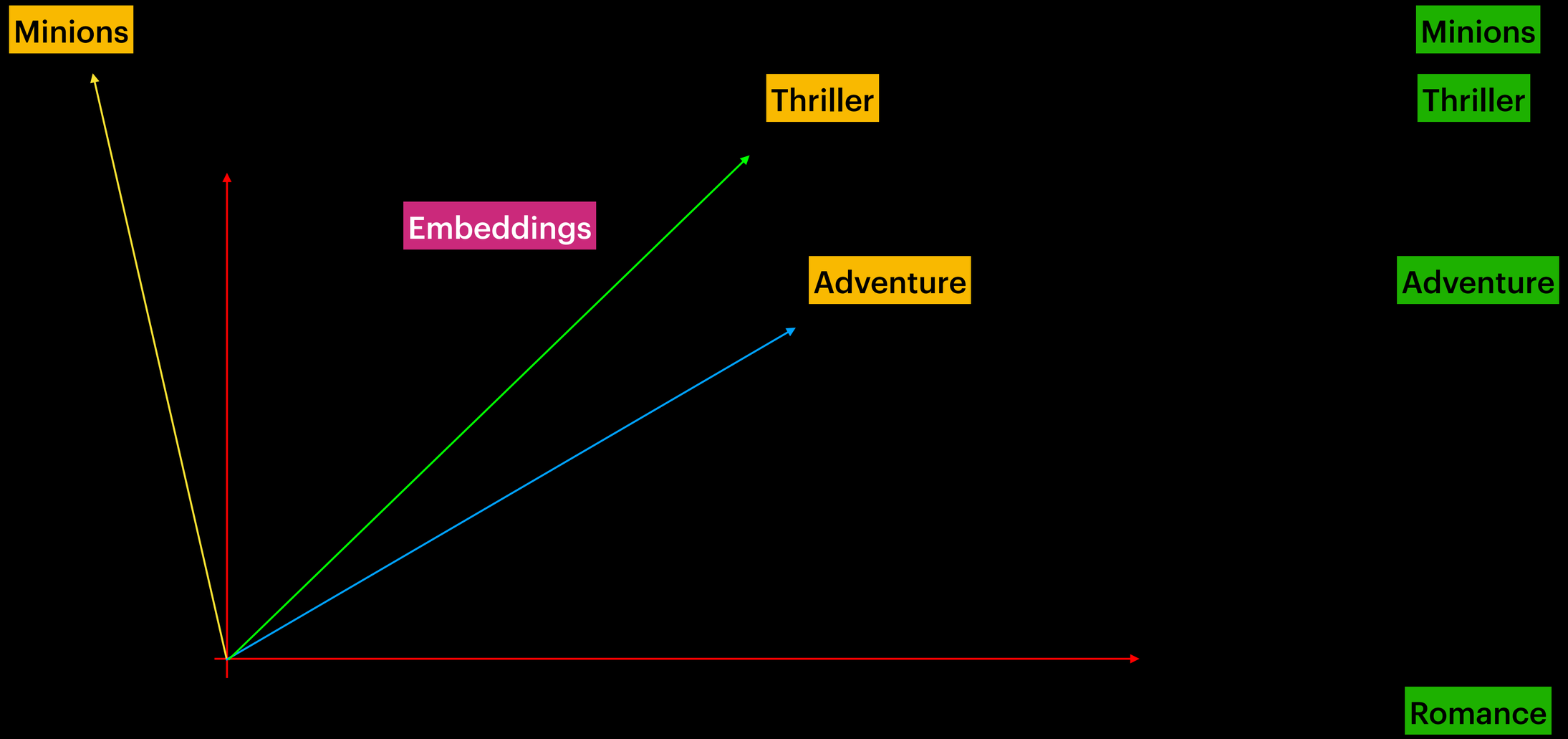
Minions

Thriller

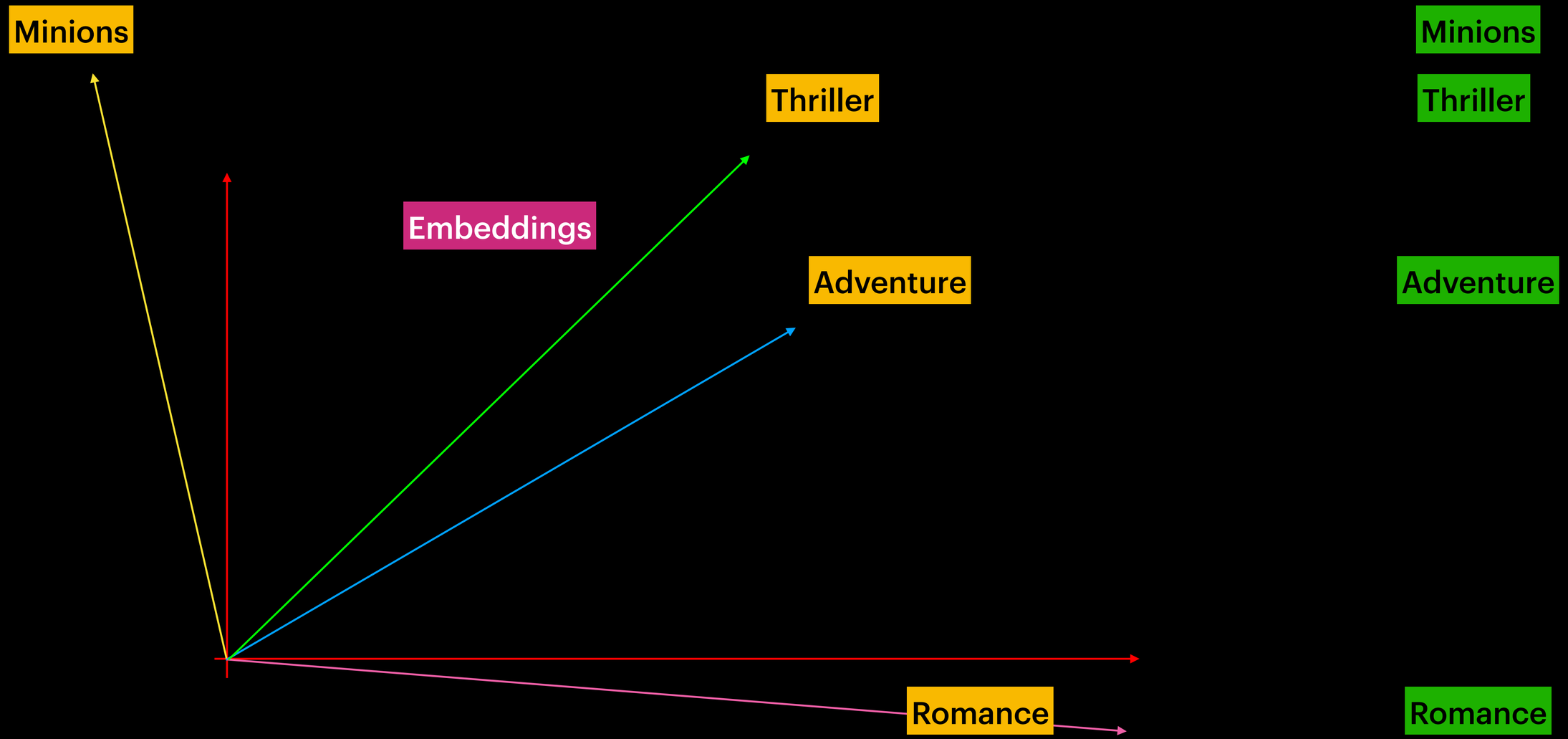
Adventure

Romance

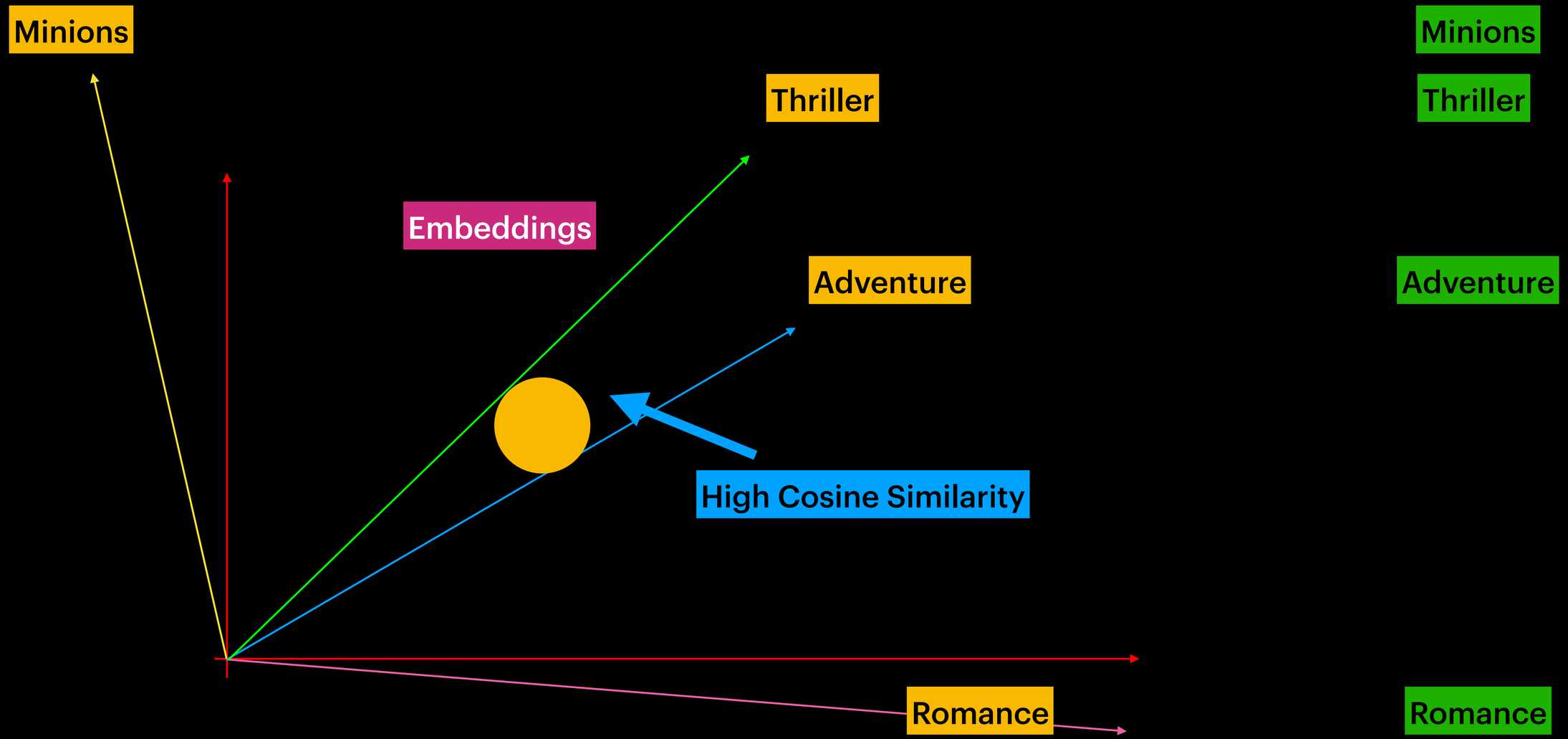
Embeddings for Words



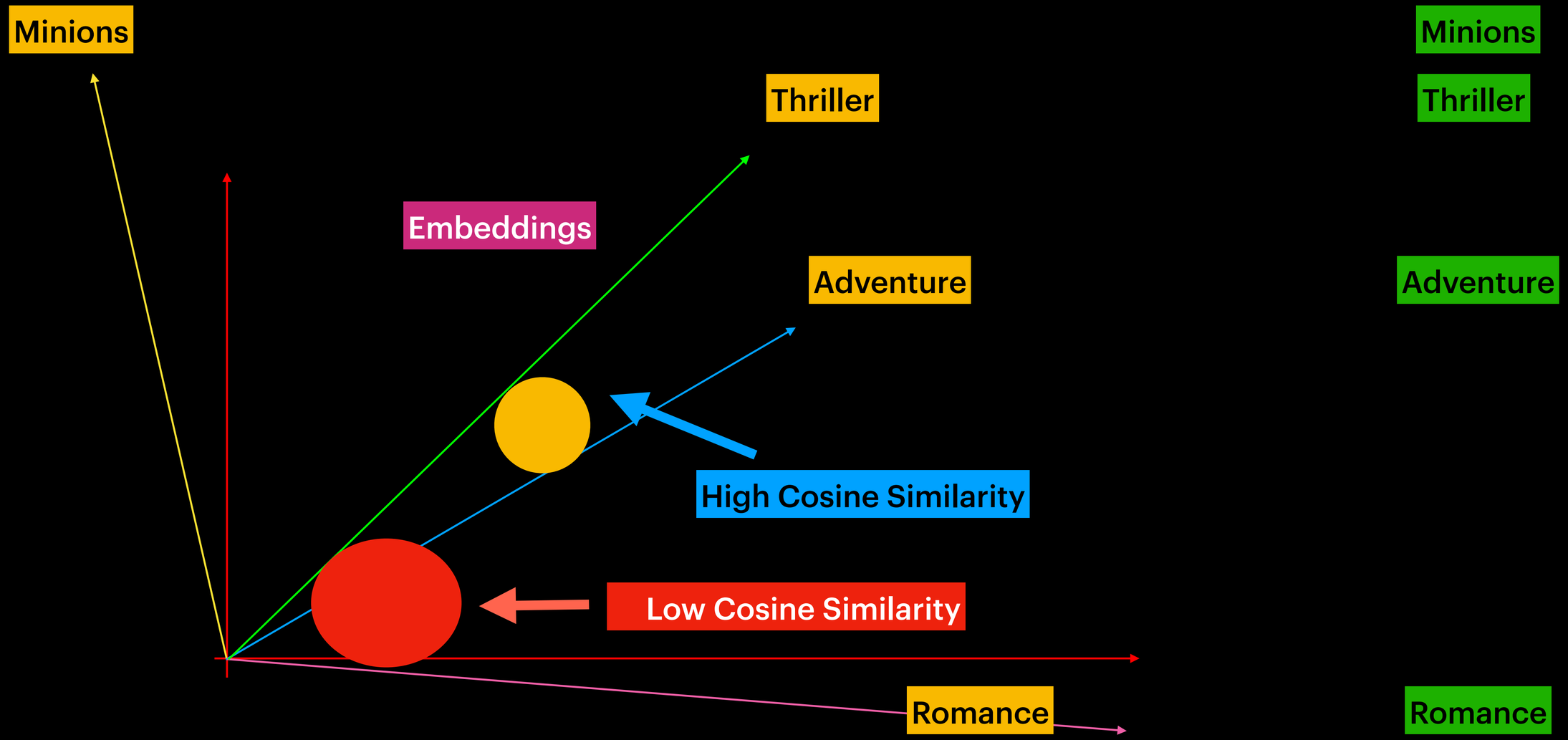
Embeddings for Words



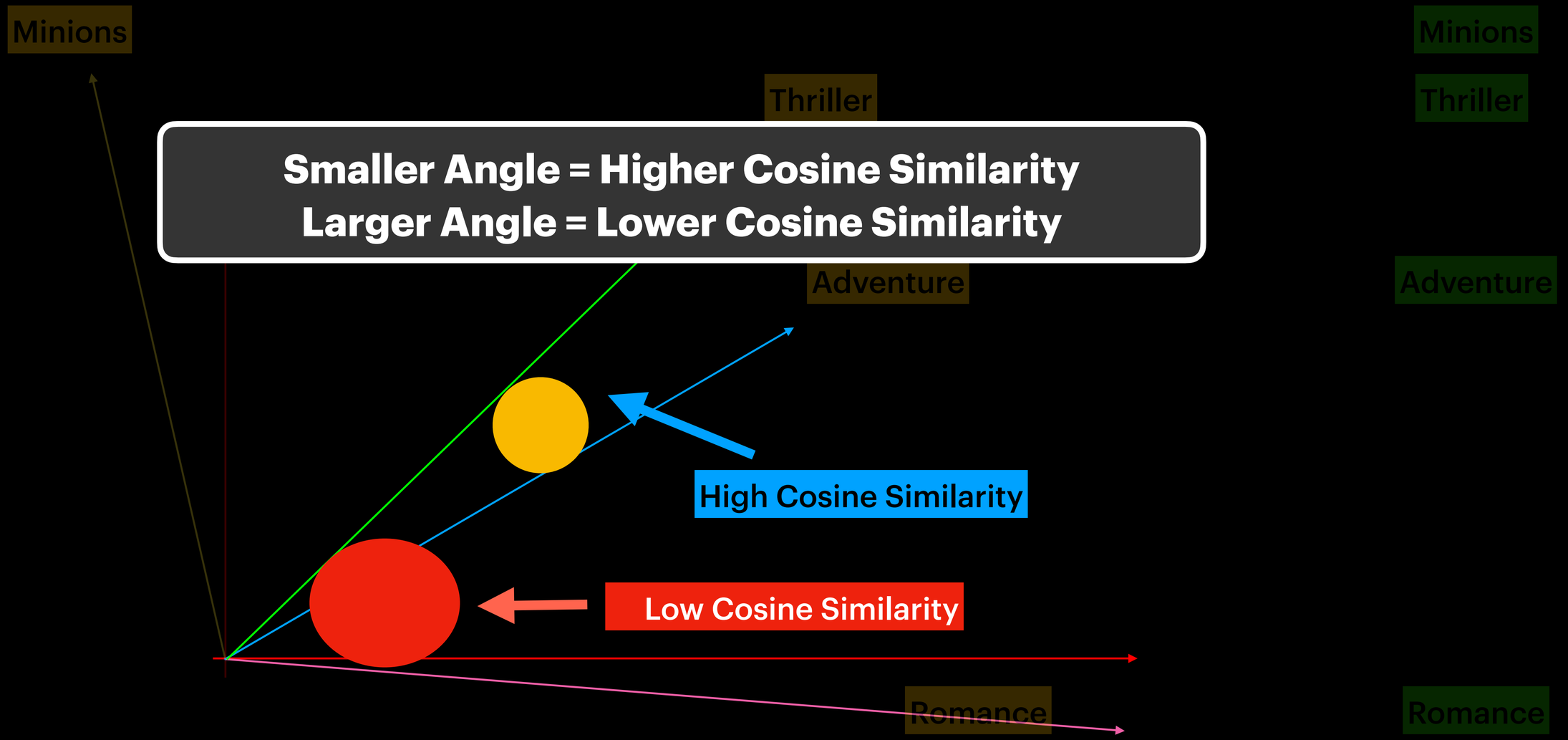
Embeddings for Words



Embeddings for Words



Embeddings for Words



Embeddings for Sentences

Embeddings for Sentences

Sentence: I love Minions

Embeddings for Sentences

Sentence: I love Minions

How do we Embed a Sentence ?

Embeddings for Sentences

Sentence: I love Minions

How do we Embed a Sentence ?

One idea is to average word embeddings in Sentence

Embeddings for Sentences

Sentence: I love Minions

Embedding[Sentence] = Avg(Embedding[I] , Embedding[love], Embedding[Minions])

Embeddings for Sentences

Sentence: I love Minions

Embedding[Sentence] = Avg(Embedding[I] , Embedding[love], Embedding[Minions])

What's the downside of this approach?

Consider Two sentences

Sentence 1: Milk Chocolate

Sentence 2: Chocolate Milk

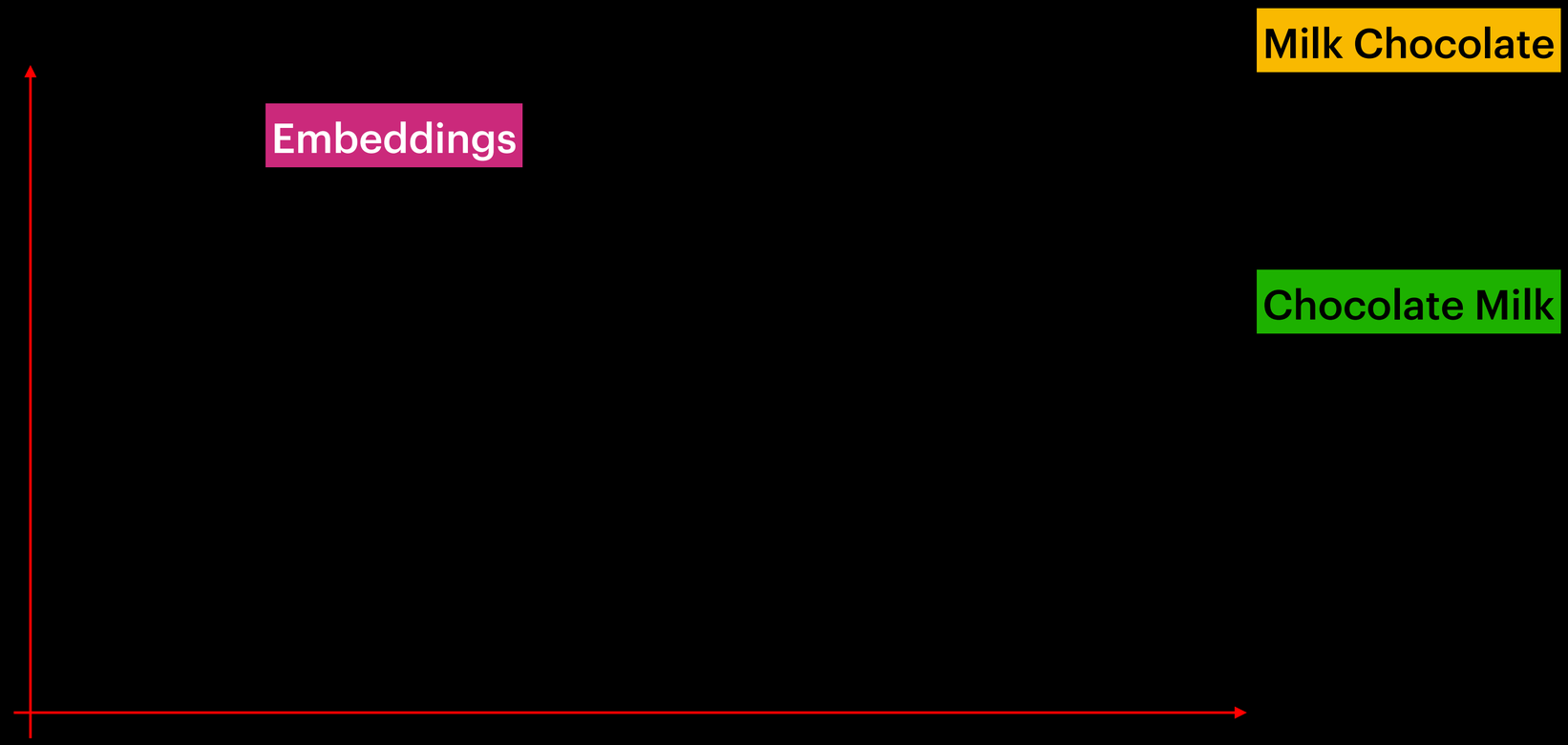
Consider Two sentences

Sentence 1: Milk Chocolate

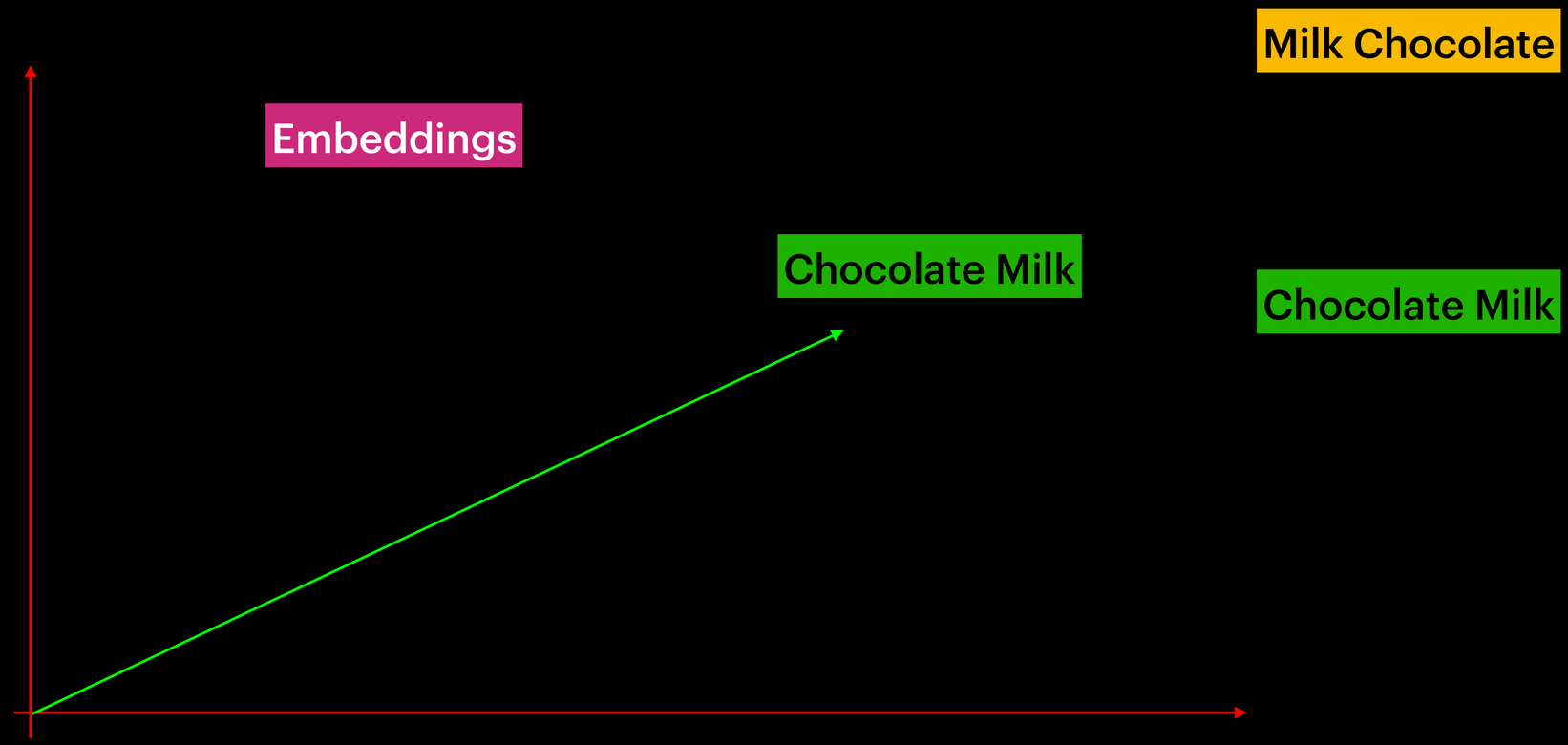
Sentence 2: Chocolate Milk

Will the averaged word embeddings be the same?

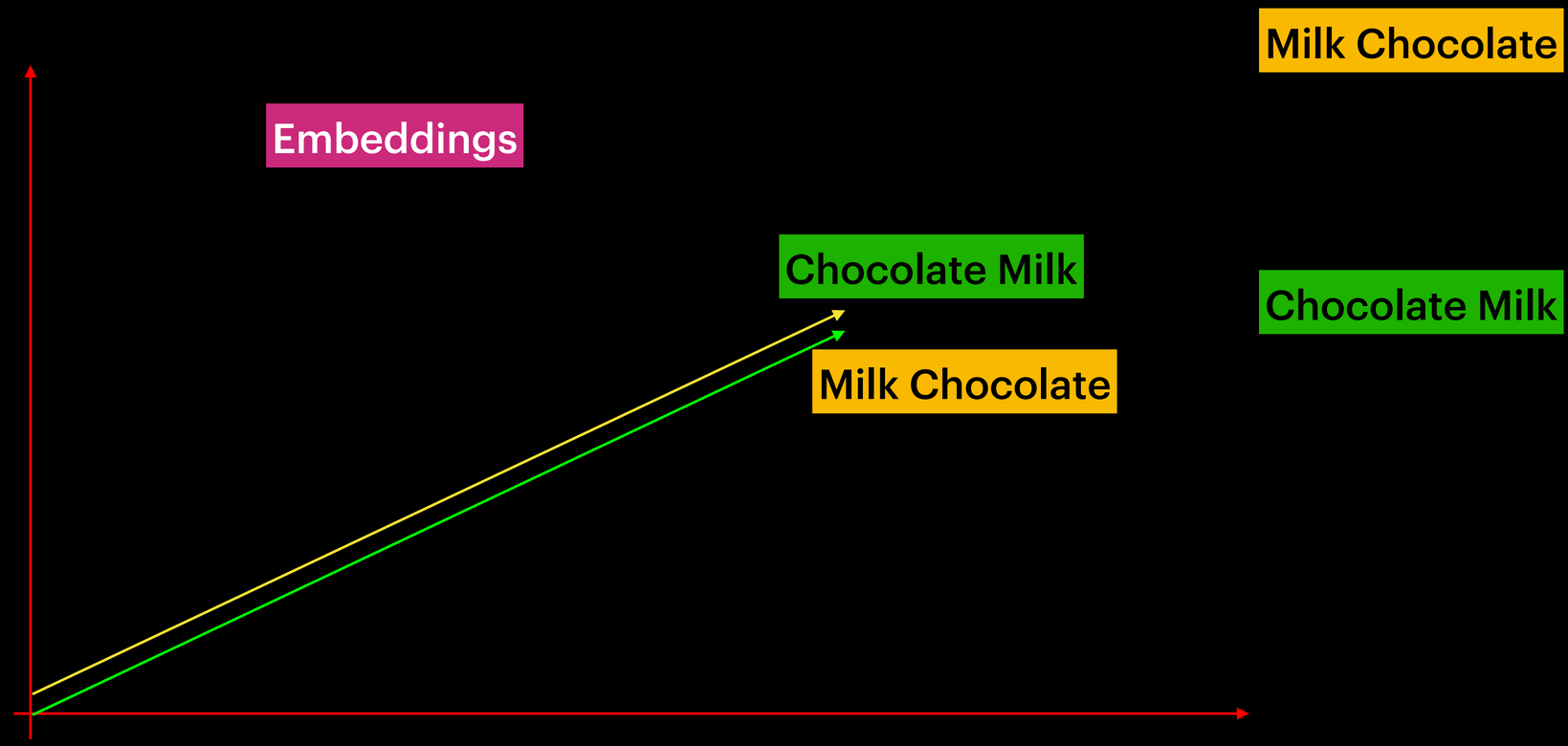
Consider Two sentences



Consider Two sentences



Cannot Differentiate Chocolate Milk From Milk Chocolate!



Embeddings for Sentences

Sentence 1: Milk Chocolate

Sentence 2: Chocolate Milk

Learning: Averaging word embeddings isn't as useful as embedding the sentence as a whole

Embeddings for Sentences

Sentence 1: Milk Chocolate

Sentence 2: Chocolate Milk

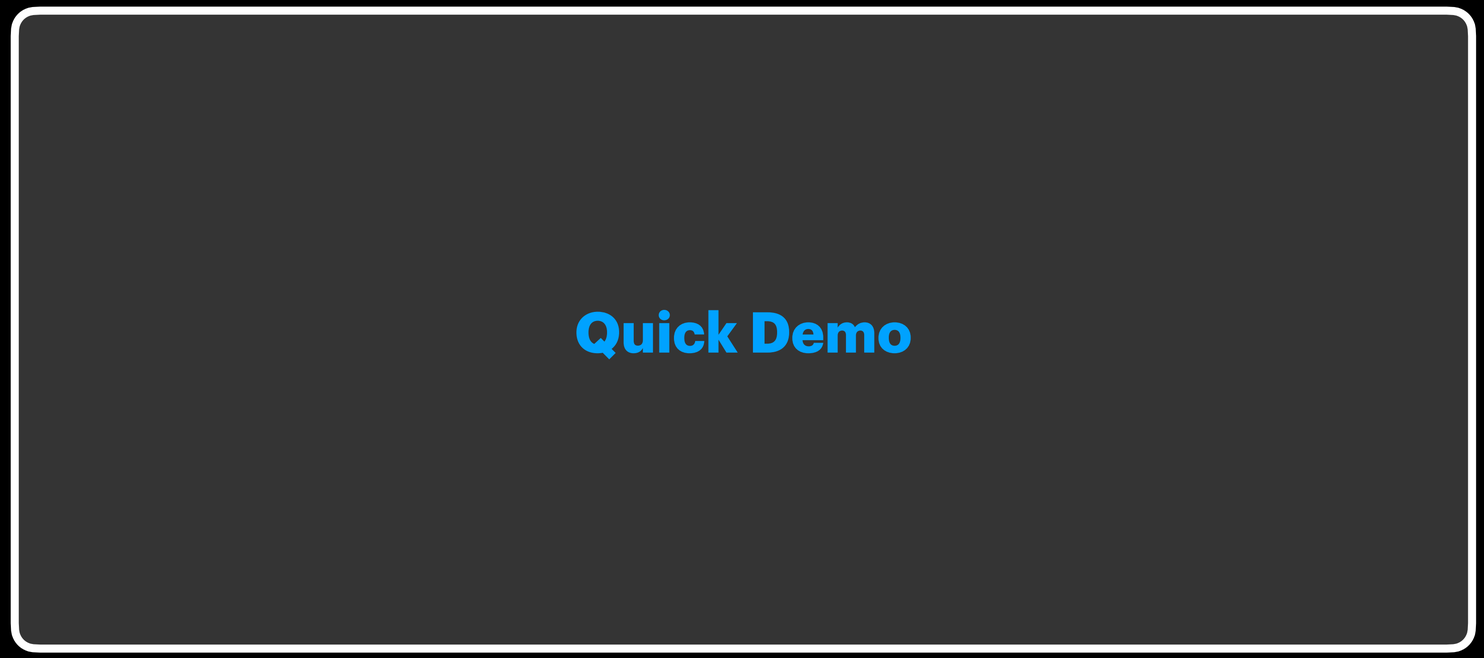
Learning: Averaging word embeddings isn't as useful as embedding the sentence as a whole

Learning: Sequence of words matters!!

Sentence BERT (sBERT) for sentence Embedings

Pre-Trained LLM to the rescue: Sentence BERT, a pre-trained LLM considers sequence structure of the sentence and is better embedding than averaging word embeddings.

Sentence BERT (sBERT) for sentence Embedings



In-class Coding Exercise (1 hour)

A. Pick another person to pair up for this
B. Starter Code for coding exercise provided on webpage
C. Exercise will involve the use of embeddings in different contexts.

Summary

Introduction and ChatGPT examples

History of LLMs and ChatGPT Engine

NLP Applications of ChatGPT

Notebook Demo

Summary

Introduction and ChatGPT examples

History of LLMs and ChatGPT Engine

NLP Applications of ChatGPT

Notebook Demo

Summary

Introduction and ChatGPT examples

History of LLMs and ChatGPT Engine

NLP Applications of ChatGPT

Notebook Demo

Summary

Introduction and ChatGPT examples

History of LLMs and ChatGPT Engine

NLP Applications of ChatGPT

Notebook Demo

Next Lecture (November 12 2023)

1. Prompt Engineering Demo and Principles

2. Fine-tuning LLMs

3. Sentiment Analysis

4. In-class Coding Exercise

Thank you!

References

Chip Huyen's blog: https://huyenchip.com/2023/05/02/rlhf.html

https://www.linkedin.com/pulse/meta-llama-vs-chatgpt-comprehensive-