EEC S6895 Adv. Big Data and AI

Lecture 2: Foundations

Prof. Ching-Yung Lin

January 26th, 2024

Milestone 1 Instruction

Milestone 1

- Each team will present on either February 2nd or February 9th
- Prepare about 20 mins of presentation (and expect 3-5 mins of Q&A)
- All teams will submit a written Milestone 1 report on February 9th.

- Key elements in Milestone 1 presentation:
  - Task Goal:
    - What do you want to achieve?
    - Why is the research and development important?
    - Is this a new topic that considers challenges of
      - Volume
      - Velocity
      - Variety
    - Does this topic try to incorporate multi-discipline knowledge?
    - (Optional) Is it related to A.I.?

Milestone 1

- Literature Survey:
  - What are the prior arts? What related works were done before?
  - Which research publications, tools and products may be utilized to build upon them to achieve the goal?

- Methodology:
  - What types of novel algorithms I shall try to invent and implement?
  - Where will you try to gather the data — Existing dataset? Self-collected dataset? Live dataset?

Milestone 1

- System:
  - What will your final system look like — potential backend components and interaction with front end?
  - How will you create visualization and user interface to help users consume your analysis outcome?

- Timeline:
  - What may you achieve in Milestones 2 and 3, and in the Final Project?

Note — it's good to think about what you want to achieve as complete as possible and study what other people have done. But, like all projects, everything can change when you make progress. Please do not afraid of making bold assumptions and attempt!!

Big Data Analytics Basic Foundation

Key Open Source Big Data Foundations

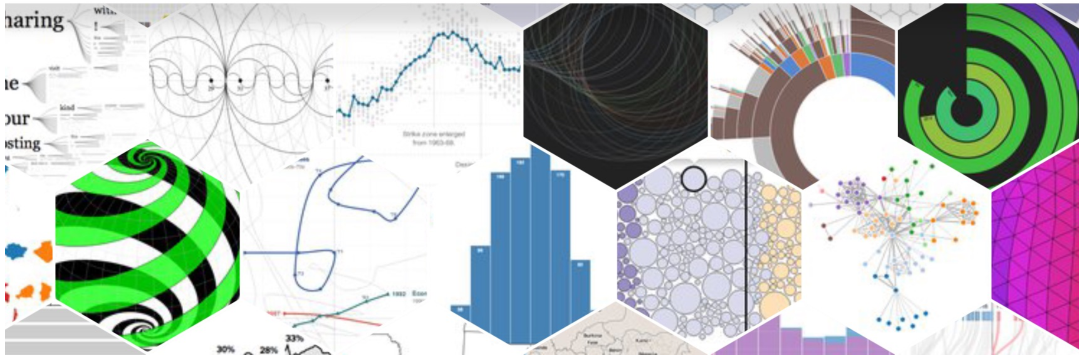
APACHE hadoop Apache Hadoop

APACHE Spark™ Lightning-fast unified analytics engine

APACHE kafka® A distributed streaming platform

Other Important Foundations (visualization and web servers)

Data-Driven Documents



Flask
web development, one drop at a time

APACHE
HTTP SERVER PROJECT


ENTITY
ENERGY
INFLUENCE
COMPLEX
INPUT
SYSTEMS
INFORMATION
PROCESS
MATERIAL
OVERALL
SCIENCE
THINKING
PROBLEM
APPROACH
UNDERSTANDING
SOLVING
COMPONENTS
SYSTEM
OUTPUT
123rf.com

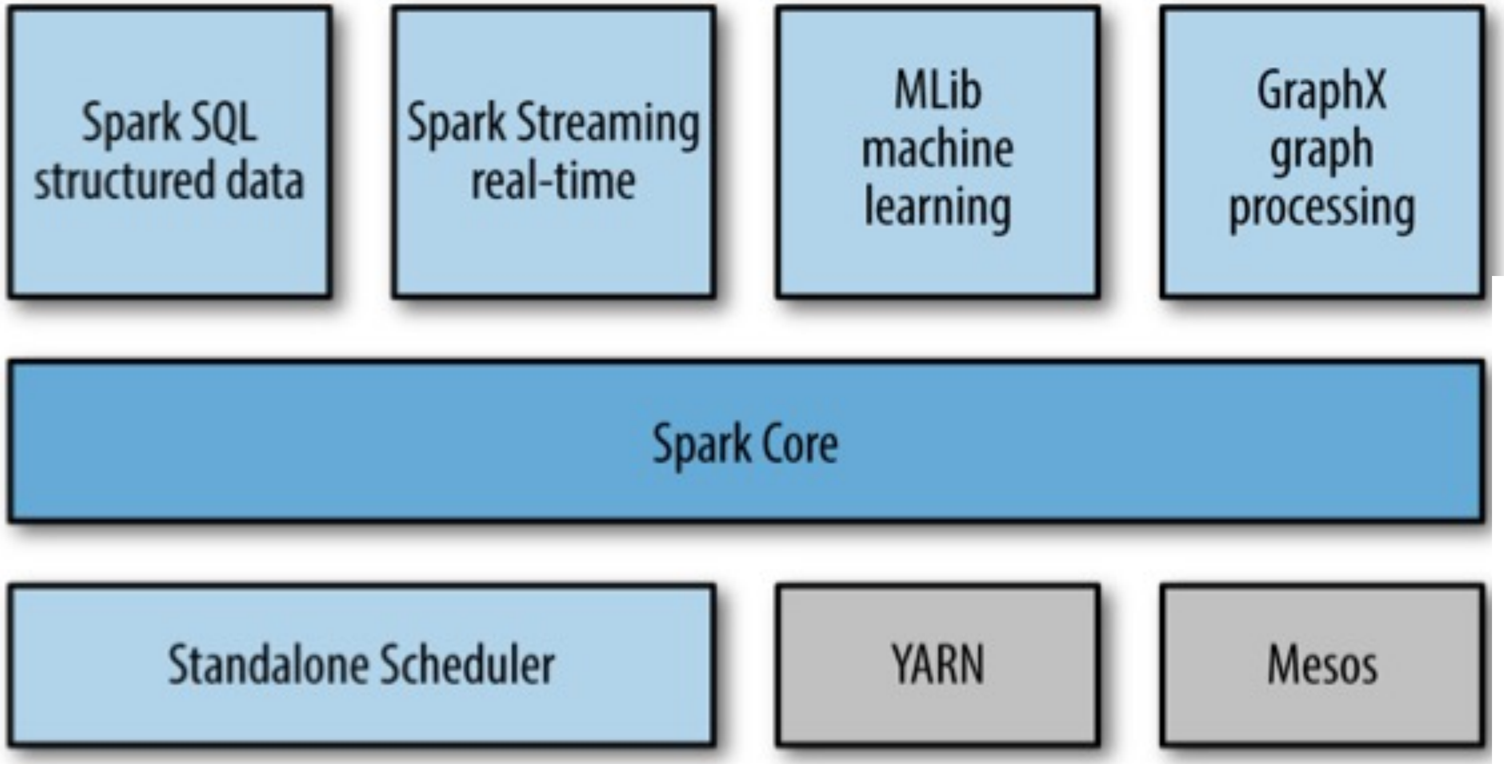
Where to store data?
How to get data in and out?
How to manage access of data?

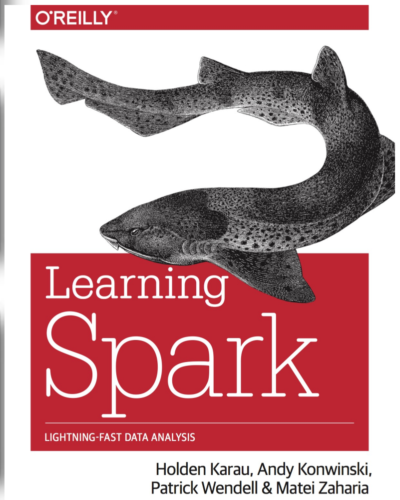
123rf.com

How do I process the data?
How do I execute machine learning from the data?
How do I tell people my analytics results?

Spark

Lightning-fast unified analytics engine





Spark MLlib

Includes:

- ML Algorithms: common learning algorithms such as classification, regression, clustering, and collaborative filtering
- Featurization: feature extraction, transformation, dimensionality reduction, and selection
- Pipelines: tools for constructing, evaluating, and tuning ML Pipelines
- Persistence: saving and load algorithms, models, and Pipelines
- Utilities: linear algebra, statistics, data handling, etc.

MLlib: Main Guide

- Basic statistics
- Data sources
- Pipelines
- Extracting, transforming and selecting features
- Classification and Regression
- Clustering
- Collaborative filtering
- Frequent Pattern Mining
- Model selection and tuning
- Advanced topics

Spark MLlib Basic Statistics

Includes:

- Correlation
- Hypothesis testing
- Summarizer

Example of Calculating Correlation of Time Sequences:

```python
from pyspark.ml.linalg import Vectors
from pyspark.ml.stat import Correlation

data = [(Vectors.sparse(4, [(0, 1.0), (3, -2.0)]),),
        (Vectors.dense([4.0, 5.0, 0.0, 3.0]),),
        (Vectors.dense([6.0, 7.0, 0.0, 8.0]),),
        (Vectors.sparse(4, [(0, 9.0), (3, 1.0)]),)]
df = spark.createDataFrame(data, ["features"])

r1 = Correlation.corr(df, "features").head()
print("Pearson correlation matrix:\n" + str(r1[0]))

r2 = Correlation.corr(df, "features", "spearman").head()
print("Spearman correlation matrix:\n" + str(r2[0]))
```

MLlib: Main Guide

- Basic statistics
- Data sources
- Pipelines
- Extracting, transforming and selecting features
- Classification and Regression
- Clustering
- Collaborative filtering
- Frequent Pattern Mining
- Model selection and tuning
- Advanced topics

Spark MLlib Features

Includes:

- Extraction: Extracting features from "raw" data
- Transformation: Scaling, converting, or modifying features
- Selection: Selecting a subset from a larger set of features
- Locality Sensitive Hashing (LSH): This class of algorithms combines aspects of feature transformation with other algorithms.

- Feature Extractors
  - TF-IDF
  - Word2Vec
  - CountVectorizer
  - FeatureHasher
- Feature Selectors
  - VectorSlicer
  - RFormula
  - ChiSqSelector
- Feature Transformers
  - Tokenizer
  - StopWordsRemover
  - n-gram
  - Binarizer
  - PCA
  - PolynomialExpansion
  - Discrete Cosine Transform (DCT)
  - StringIndexer
  - IndexToString
  - OneHotEncoder (Deprecated since 2.3.0)
  - OneHotEncoderEstimator
  - VectorIndexer
  - Interaction
  - Normalizer
  - StandardScaler
  - MinMaxScaler
  - MaxAbsScaler
  - Bucketizer
  - ElementwiseProduct
  - SQLTransformer
  - VectorAssembler
  - VectorSizeHint
  - QuantileDiscretizer
  - Imputer

Spark MLlib Supervised Machine Learning Algorithms

Classification

- Logistic regression
    - Binomial logistic regression
    - Multinomial logistic regression
- Decision tree classifier
- Random forest classifier
- Gradient-boosted tree classifier
- Multilayer perceptron classifier
- Linear Support Vector Machine
- One-vs-Rest classifier (a.k.a. One-vs-All)
- Naive Bayes

Regression

- Linear regression
- Generalized linear regression
    - Available families
- Decision tree regression
- Random forest regression
- Gradient-boosted tree regression
- Survival regression
- Isotonic regression

Spark MLlib UnSupervised Machine Learning & Recommendation Algorithms

Clustering:

- K-means
  - Input Columns
  - Output Columns
- Latent Dirichlet allocation (LDA)
- Bisecting k-means
- Gaussian Mixture Model (GMM)
  - Input Columns
  - Output Columns

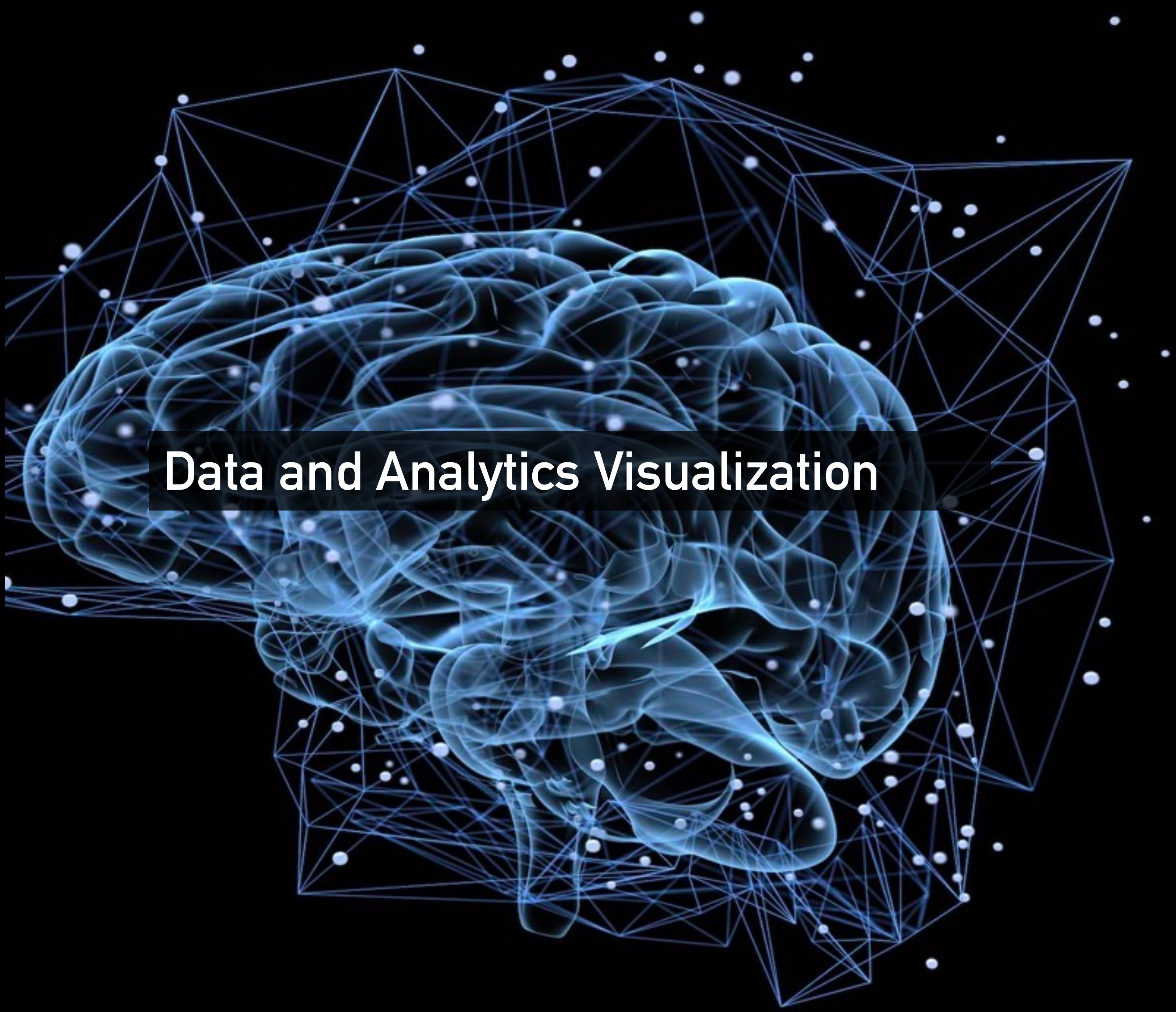
Collaborative Filtering:

- Explicit vs. implicit feedback
- Scaling of the regularization parameter
- Cold-start strategy

Spark MLlib Model Selection and Tuning

- Model selection (a.k.a. hyperparameter tuning)
- Cross-Validation
- Train-Validation Split

Data and Analytics Visualization



Webservers

- Apache (http server) — the oldest and most popular web server exists in every linux machine, including MacOS machines.

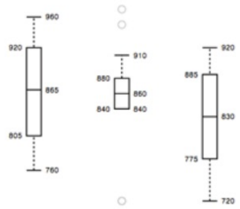
- — display webpages ctory

```python
from flask import Flask, escape, request

app = Flask(__name__)

@app.route('/')
def hello():
    name = request.args.get("name", "World")
    return f'Hello, {escape(name)}!'
```

D3 Visualization (via Javascript) Examples

Visual Index

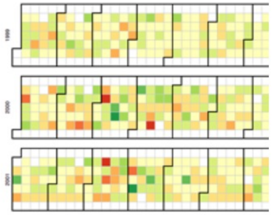
Box Plots



Bubble Chart

Bullet Charts

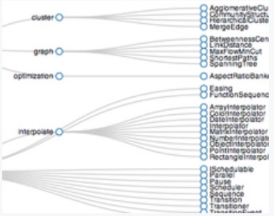
Calendar View



Non-contiguous Cartogram

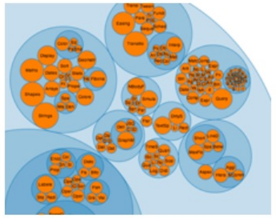
Chord Diagram

Dendrogram

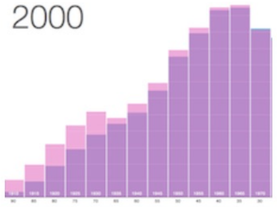


Force-Directed Graph

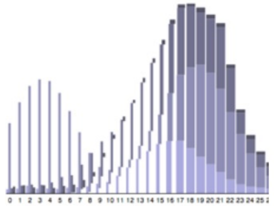
Circle Packing



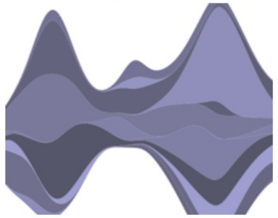
Population Pyramid



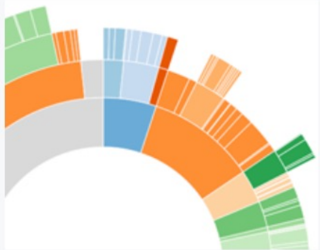
Stacked Bars



Streamgraph

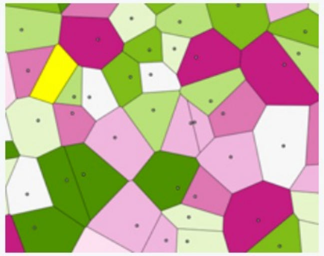


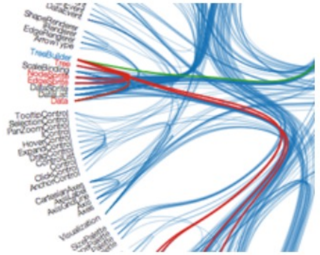
D3 Visualization (via Javascript) Examples

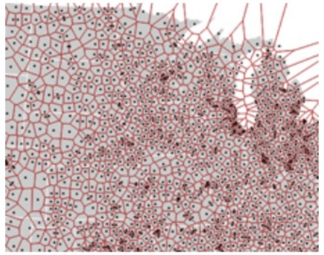
Sunburst


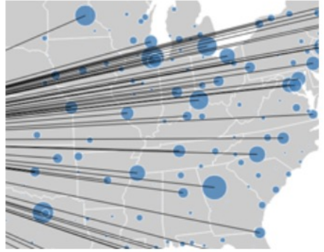
Node-Link Tree

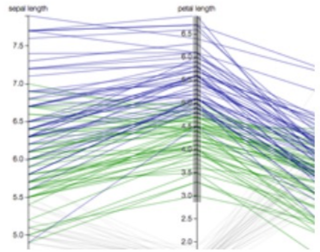

Treemap

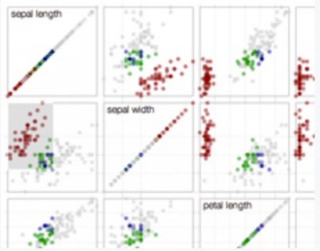

Voronoi Diagram


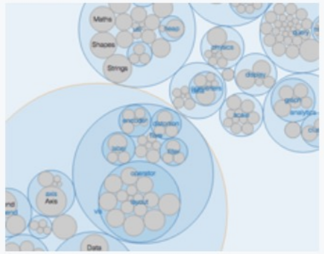
Hierarchical Edge Bundling


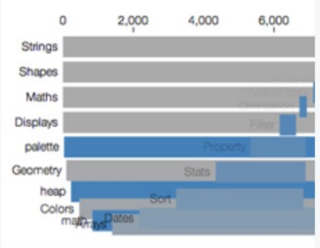
Voronoi Diagram


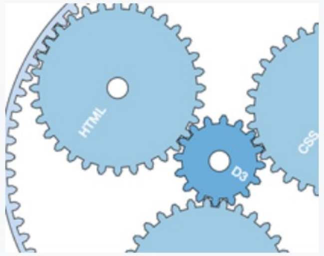
Bubble Map


Parallel Coordinates


Scatterplot Matrix


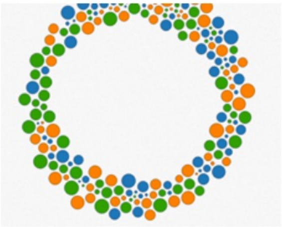
Zoomable Pack Layout


Hierarchical Bars


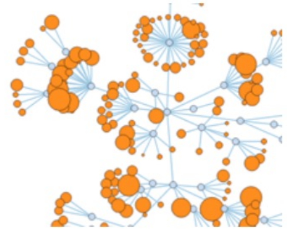
Epicyclical Gears


D3 Visualization (via Javascript) Examples

Collision Detection



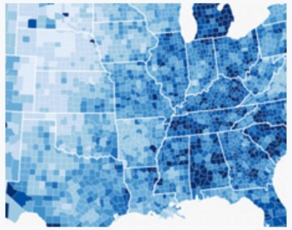
Collapsible Force Layout



Force-Directed States

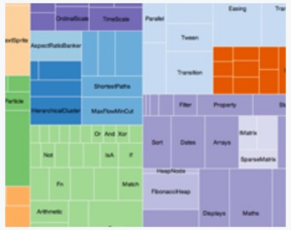
Versor Dragging

Choropleth



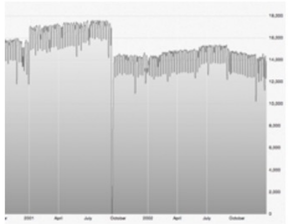
Collapsible Tree Layout

Zoomable Treemap

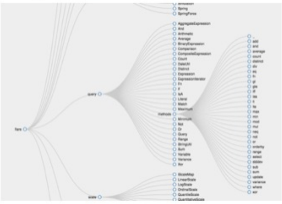


Zoomable Icicle

Zoomable Area Chart

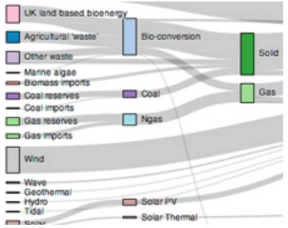


Drag and Drop Collapsible Tree Layout



Radial Cluster Layout

Sankey Diagram



D3 Installation

Installing

For NPM, `npm install d3`. For Yarn, `yarn add d3`. Otherwise, download the latest release. The released bundle supports AMD, CommonJS, and vanilla environments. Create a custom bundle using Rollup or your preferred bundler. You can also load directly from d3js.org:

```
<script src="https://d3js.org/d3.v5.js"></script>
```

For the minified version:

```
<script src="https://d3js.org/d3.v5.min.js"></script>
```

You can also use the standalone D3 microlibraries. For example, d3-selection:

```
<script src="https://d3js.org/d3-selection.v1.min.js"></script>
```

D3 Example Circles

https://bost.ocks.org/mike/circles/

```
<svg width="720" height="120">
  <circle cx="40" cy="60" r="10"></circle>
  <circle cx="80" cy="60" r="10"></circle>
  <circle cx="120" cy="60" r="10"></circle>
</svg>
```

```
var circle = d3.selectAll("circle");
```

```
circle.style("fill", "steelblue");
circle.attr("r", 30);
```

```
circle.attr("cx", function() { return Math.random() * 720; });
```

D3 Bar Chart Tutorial

https://bost.ocks.org/mike/bar/

```
var data = [4, 8, 15, 16, 23, 42];
```

Selecting an Element

Javascript:

```
var div = document.createElement("div");
div.innerHTML = "Hello, world!";
document.body.appendChild(div);
```

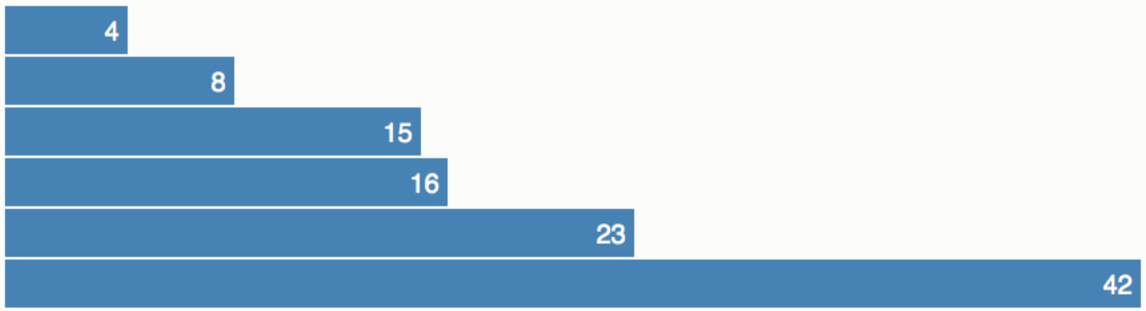
D3:

```
var body = d3.select("body");
var div = body.append("div");
div.html("Hello, world!");
```

D3 Bar Chart Tutorial

Coding a Chart, Manually

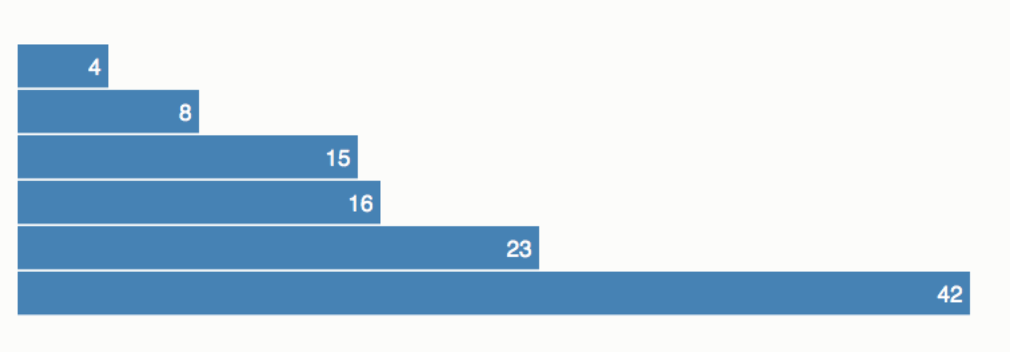
```
<!DOCTYPE html>
<style>

.chart div {
  font: 10px sans-serif;
  background-color: steelblue;
  text-align: right;
  padding: 3px;
  margin: 1px;
  color: white;
}

</style>
<div class="chart">
  <div style="width: 40px;">4</div>
  <div style="width: 80px;">8</div>
  <div style="width: 150px;">15</div>
  <div style="width: 160px;">16</div>
  <div style="width: 230px;">23</div>
  <div style="width: 420px;">42</div>
</div>
```



D3 Bar Chart Tutorial

Full code to do it manually

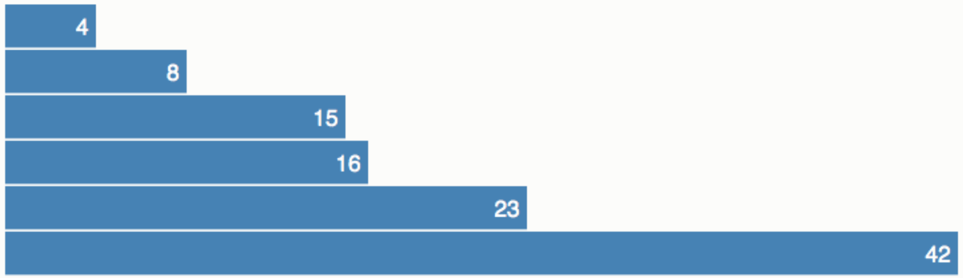
```
<!DOCTYPE html>
<style>

.chart rect {
  fill: steelblue;
}

.chart text {
  fill: white;
  font: 10px sans-serif;
  text-anchor: end;
}

</style>
<svg class="chart" width="420" height="120">
  <g transform="translate(0,0)">
    <rect width="40" height="19"></rect>
    <text x="37" y="9.5" dy=".35em">4</text>
  </g>
  <g transform="translate(0,20)">
    <rect width="80" height="19"></rect>
    <text x="77" y="9.5" dy=".35em">8</text>
  </g>
  <g transform="translate(0,40)">
    <rect width="150" height="19"></rect>
    <text x="147" y="9.5" dy=".35em">15</text>
  </g>
  <g transform="translate(0,60)">
    <rect width="160" height="19"></rect>
    <text x="157" y="9.5" dy=".35em">16</text>
  </g>
  <g transform="translate(0,80)">
    <rect width="230" height="19"></rect>
    <text x="227" y="9.5" dy=".35em">23</text>
  </g>
  <g transform="translate(0,100)">
    <rect width="420" height="19"></rect>
    <text x="417" y="9.5" dy=".35em">42</text>
  </g>
</svg>
```



D3 Bar Chart Tutorial

Full code to do it automatically

```
<!DOCTYPE html>
<meta charset="utf-8">
<style>

.chart rect {
  fill: steelblue;
}

.chart text {
  fill: white;
  font: 10px sans-serif;
  text-anchor: end;
}

</style>
<svg class="chart"></svg>
<script src="//d3js.org/d3.v3.min.js" charset="utf-8"></script>
```



D3 Bar Chart Tutorial

Full code to do it automatically

```
<script>

var data = [4, 8, 15, 16, 23, 42];

var width = 420,
    barHeight = 20;

var x = d3.scale.linear()
    .domain([0, d3.max(data)])
    .range([0, width]);

var chart = d3.select(".chart")
    .attr("width", width)
    .attr("height", barHeight * data.length);

var bar = chart.selectAll("g")
    .data(data)
  .enter().append("g")
    .attr("transform", function(d, i) { return "translate(0," + i * barHeight + ")"; });

bar.append("rect")
    .attr("width", x)
    .attr("height", barHeight - 1);

bar.append("text")
    .attr("x", function(d) { return x(d) - 3; })
    .attr("y", barHeight / 2)
    .attr("dy", ".35em")
    .text(function(d) { return d; });

</script>
```

D3 Bar Chart Tutorial

Load data

```
// 1. Code here runs first, before the download starts.

d3.tsv("data.tsv", function(error, data) {
  // 3. Code here runs last, after the download finishes.
});

// 2. Code here runs second, while the file is downloading.
```

```
name        value
Locke       4
Reyes       8
Ford        15
Jarrah      16
Shephard    23
Kwon        42
```

The equivalent of Javascript code:

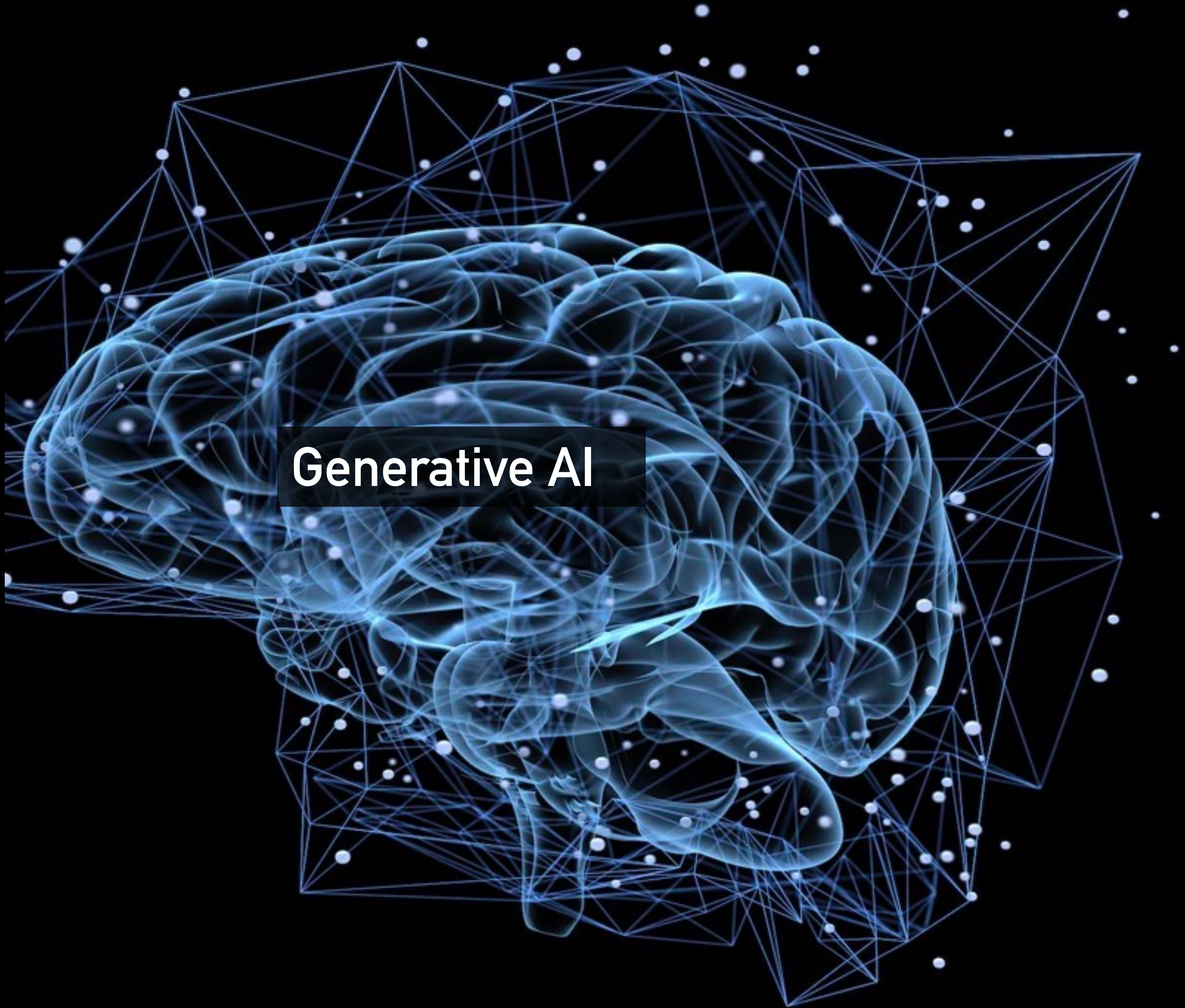
```
var data = [
  {name: "Locke",    value:  4},
  {name: "Reyes",    value:  8},
  {name: "Ford",     value: 15},
  {name: "Jarrah",   value: 16},
  {name: "Shephard", value: 23},
  {name: "Kwon",     value: 42}
];
```

D3 Bar Chart Tutorial

```
<!DOCTYPE html>
<meta charset="utf-8">
<style>

.chart rect {
  fill: steelblue;
}

.chart text {
  fill: white;
  font: 10px sans-serif;
  text-anchor: end;
}

</style>
<svg class="chart"></svg>
<script src="//d3js.org/d3.v3.min.js" charset="utf-8"></script>
```

D3 Bar Chart Tutorial

```
<script>

var width = 420,
    barHeight = 20;

var x = d3.scale.linear()
    .range([0, width]);

var chart = d3.select(".chart")
    .attr("width", width);

d3.tsv("data.tsv", type, function(error, data) {
  x.domain([0, d3.max(data, function(d) { return d.value; })]);

  chart.attr("height", barHeight * data.length);

  var bar = chart.selectAll("g")
      .data(data)
    .enter().append("g")
      .attr("transform", function(d, i) { return "translate(0," + i * barHeight + ")"; });

  bar.append("rect")
      .attr("width", function(d) { return x(d.value); })
      .attr("height", barHeight - 1);

  bar.append("text")
      .attr("x", function(d) { return x(d.value) - 3; })
      .attr("y", barHeight / 2)
      .attr("dy", ".35em")
      .text(function(d) { return d.value; });
});

function type(d) {
  d.value = +d.value; // coerce to number
  return d;
}

</script>
```


Generative AI

Levels of Artificial General Intelligence

Performance (rows) x **Generality** (columns)	**Narrow** *clearly scoped task or set of tasks*	**General** *wide range of non-physical tasks, including metacognitive abilities like learning new skills*
Level 0: No AI	**Narrow Non-AI** calculator software; compiler	**General Non-AI** human-in-the-loop computing, e.g., Amazon Mechanical Turk
Level 1: Emerging *equal to or somewhat better than an unskilled human*	**Emerging Narrow AI** GOFAI (Boden, 2014); simple rule-based systems, e.g., SHRDLU (Winograd, 1971)	**Emerging AGI** ChatGPT (OpenAI, 2023), Bard (Anil et al., 2023), Llama 2 (Touvron et al., 2023), Gemini (Pichai and Hassabis, 2023)
Level 2: Competent *at least 50th percentile of skilled adults*	**Competent Narrow AI** toxicity detectors such as Jigsaw (Das et al., 2022); Smart Speakers such as Siri (Apple), Alexa (Amazon), or Google Assistant (Google); VQA systems such as PaLI (Chen et al., 2023); Watson (IBM); SOTA LLMs for a subset of tasks (e.g., short essay writing, simple coding)	**Competent AGI** not yet achieved
Level 3: Expert *at least 90th percentile of skilled adults*	**Expert Narrow AI** spelling & grammar checkers such as Grammarly (Grammarly, 2023); generative image models such as Imagen (Saharia et al., 2022) or Dall-E 2 (Ramesh et al., 2022)	**Expert AGI** not yet achieved
Level 4: Virtuoso *at least 99th percentile of skilled adults*	**Virtuoso Narrow AI** Deep Blue (Campbell et al., 2002), AlphaGo (Silver et al., 2016, 2017)	**Virtuoso AGI** not yet achieved
Level 5: Superhuman *outperforms 100% of humans*	**Superhuman Narrow AI** AlphaFold (Jumper et al., 2021; Varadi et al., 2021), AlphaZero (Silver et al., 2018), StockFish (Stockfish, 2023)	**Artificial Superintelligence (ASI)** not yet achieved

Levels of AGI: https://arxiv.org/pdf/2311.02462.pdf

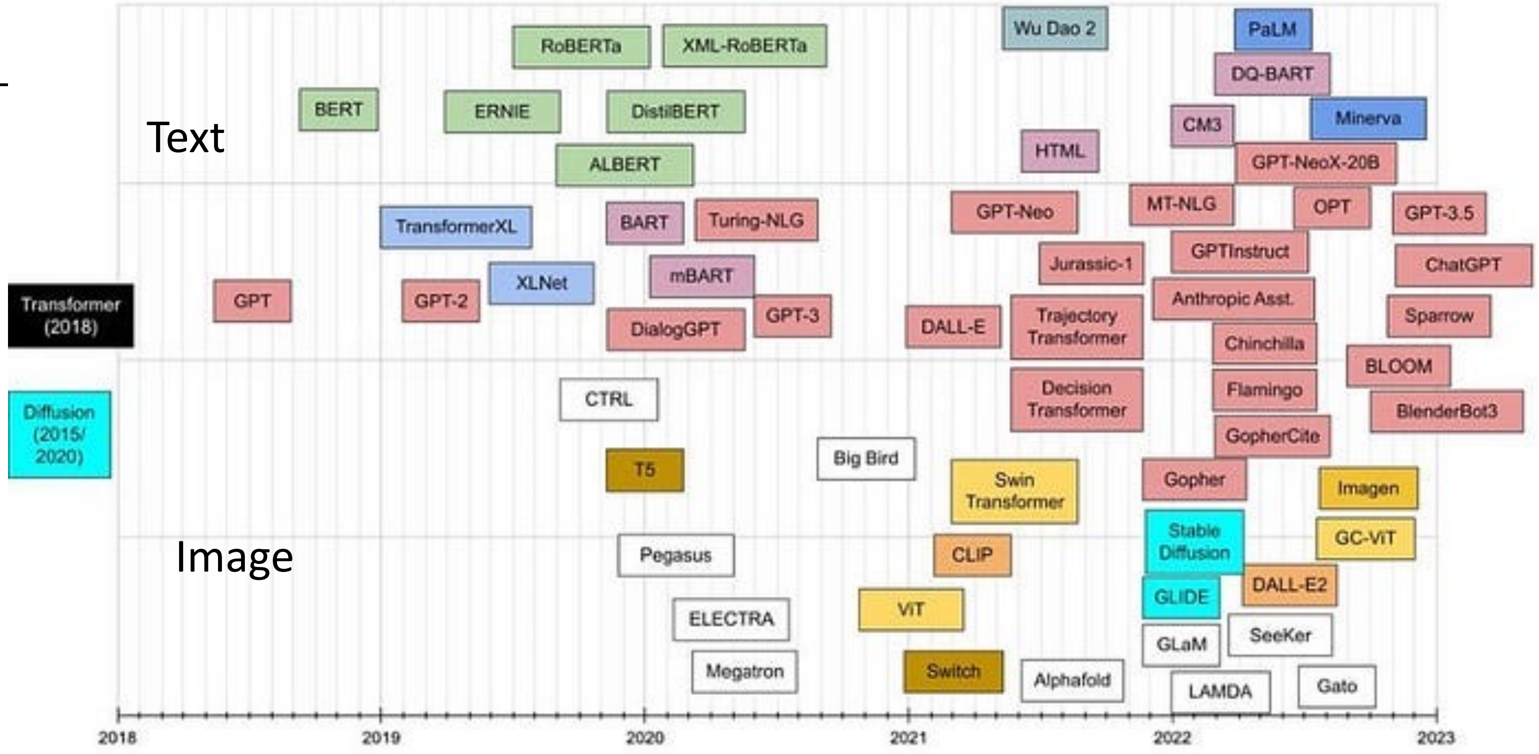
The Evolution of LLMs

1. In 2017, Google released the "Transformer Model", ~~which can be used in question-answering systems,~~ reading comprehension, sentiment analysis, instant translation of text or speech, and more

2. In 2018, OpenAI proposed "GPT" and Google proposed the "BERT" model, widely used in search engines, speech recognition, machine translation, question-answering systems, and more.

3. From 2018 to 2022, most of the research focused on BERT-related algorithms, when GPT performance was inferior to BERT

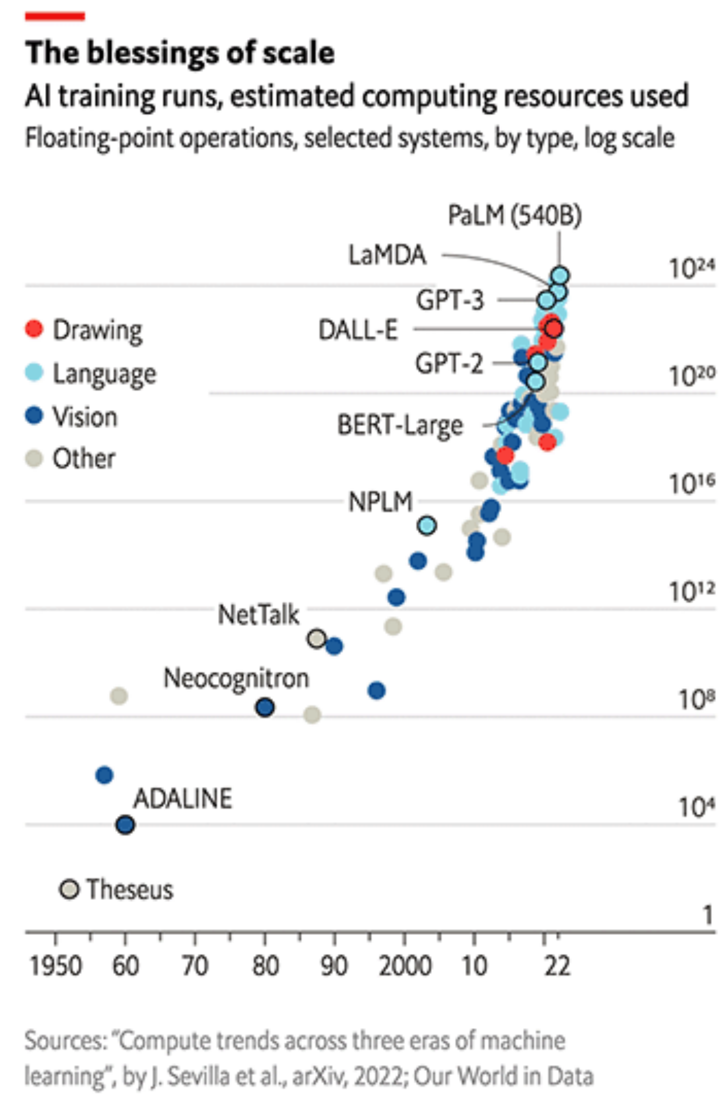
4. In 2023, ChatGPT (GPT3.5) was proposed by OpenAI, which significantly improves NLU's ability to understand most texts and surpasses humans in some area

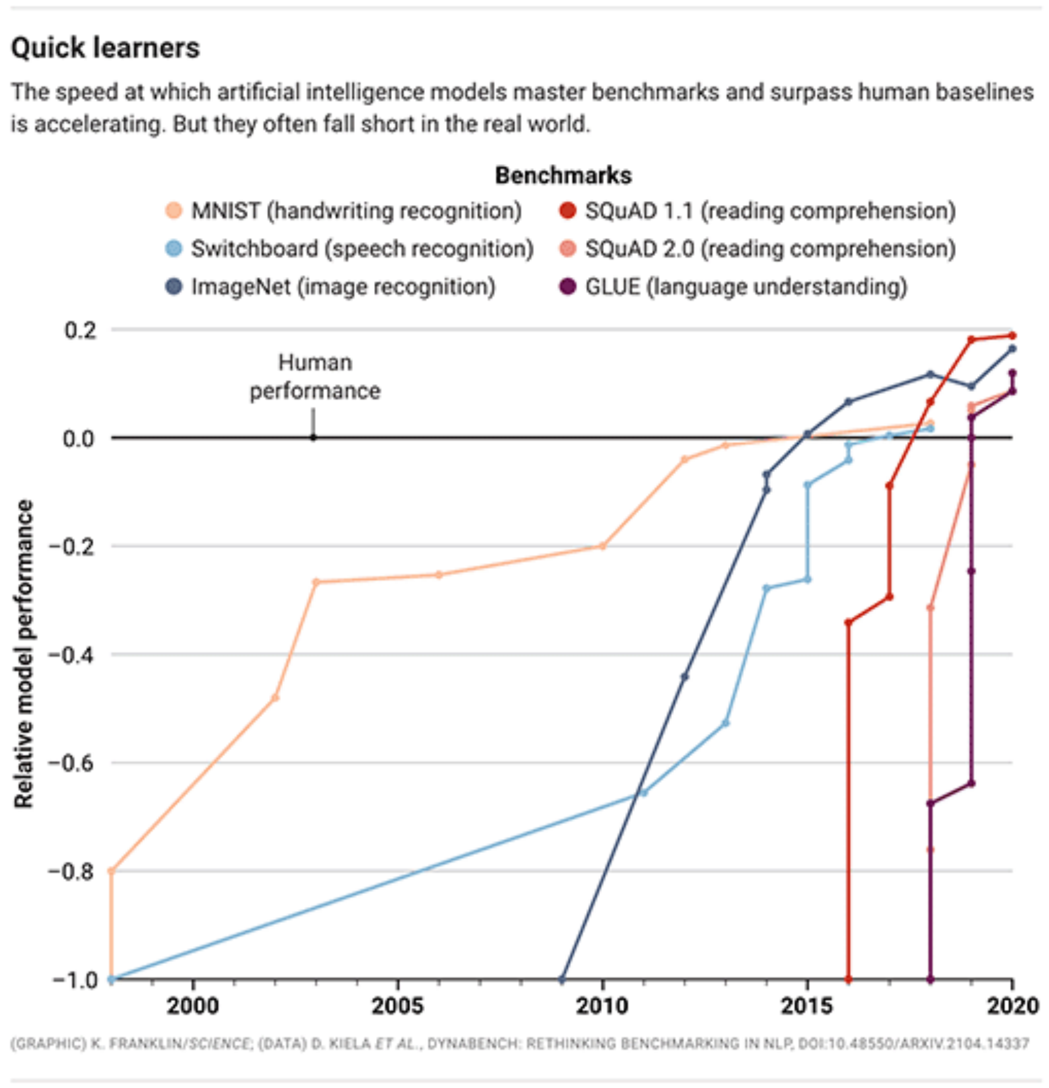


In NLU

CNN	RNN	Self-Attention
Local feature	Front and Back Dependency Issues	One to all attention, more flexible and trainable need large datasets

The speed of development of Generative AI





Generative AI Application

Multi-Model

Condition Model

Generative Model

Image Generator

Speech Generator

text Generator

Pose Generator

Conditional image Generator

Audio Generator

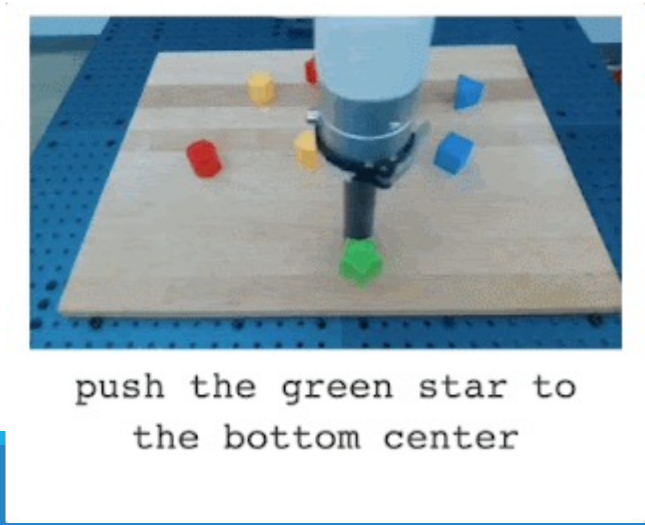
Chat Bot

Summarization and Translation

Robot

NLU + Image Generator

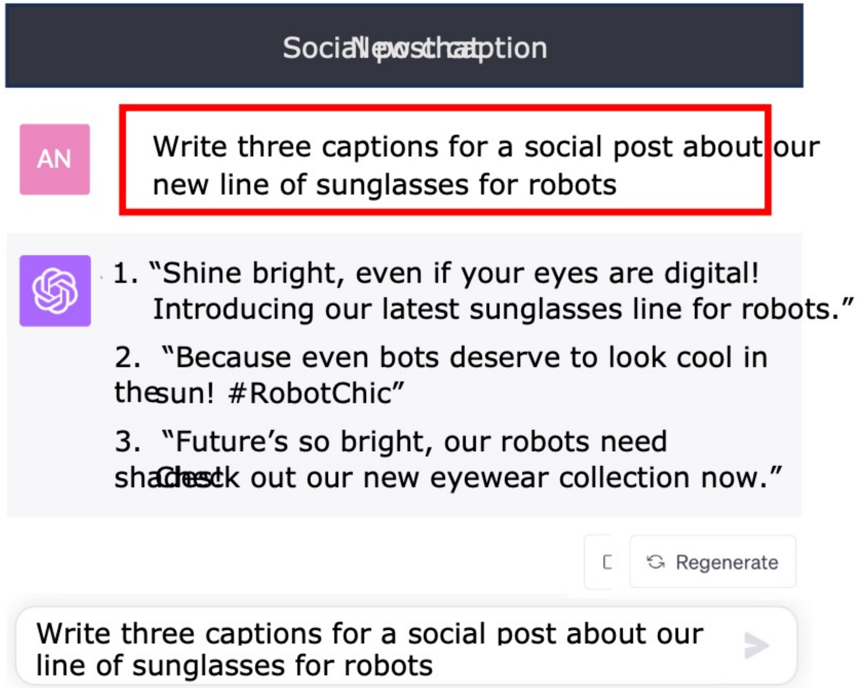
NLU + Robot



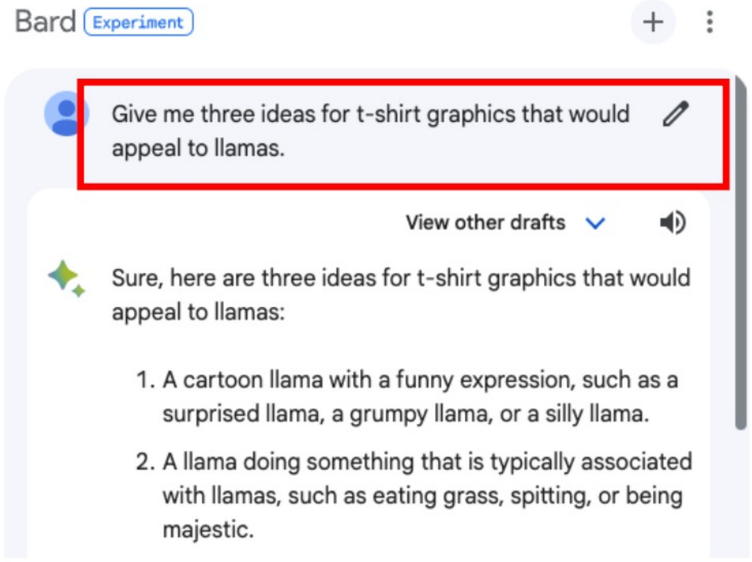
What is Generative AI

Artificial intelligence systems that can produce high quality content, specifically **text, images, and audio.**

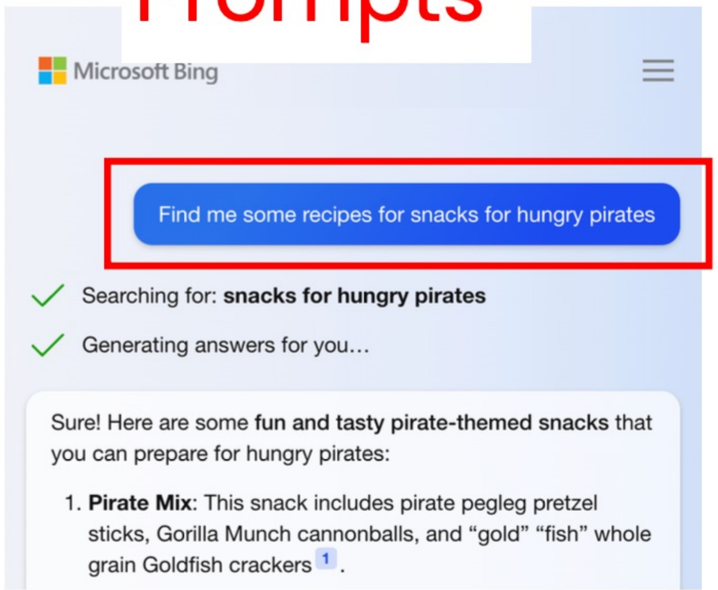
Prompts



ChatGPT/OpenAI



Bard/Google



Bing Chat/Microsoft

Multimedia Generation

A beautiful, pastoral mountain scene.
Landscape painting style (Midjourney)

Two cute kittens playing (DALL-E)

Generating Text using Large Language Models

Text generation process

I love eating ___________

prompt

bagels with cream cheese

my mother's meatloaf

out with friends

AI output

How LLM works

LLMs are built by using supervised learning (A→B) to repeatedly predict the next word.

My favorite food is a bagel with cream cheese

Input (A)	Output (B)
My favorite food is a	bagel
My favorite food is a bagel	with
My favorite food is a bagel with	cream
My favorite food is a bagel with cream	cheese

When we train a very large AI system on a lot of data (hundreds of billions of words), we get a Large Language Model like ChatGPT.

A New Way to Find Information

What is the capital of South Africa?
South Africa has three capital cities: 1. Pretoria (administrative capital) 2. Cape Town (legislative capital) 3. Bloemfontein (judicial capital)

What does LLM stand for?
LL.M. stands for "Legum Magister" in Latin, which translates to "Master of Laws" in English.
What about in the context of AI?
In the context of AI, LLM refers to "Large Language Model".

Assist Writing

Rewrite this for clarity:

Students all around the world are realizing learning has to happen not just in school but throughout life including after graduation

Students worldwide understand that learning extends beyond school and continues throughout life.

Write a 300 word story involving trucks for a 3 year old to encourage them to brush their teeth

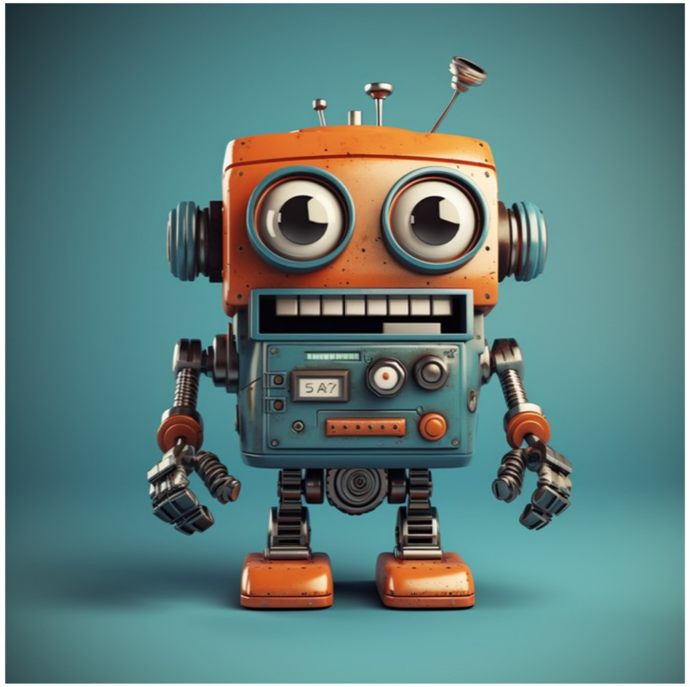
Once upon a time in a town called Toothville, there lived a happy tow truck named Timmy.

One day Timmy noticed that his shiny bumper had a dull spot. Worried, he went to the doctor [...]

Examples of Generated Images

A picture of a woman smiling

A futuristic city scene



A cool, happy robot

Image Generation

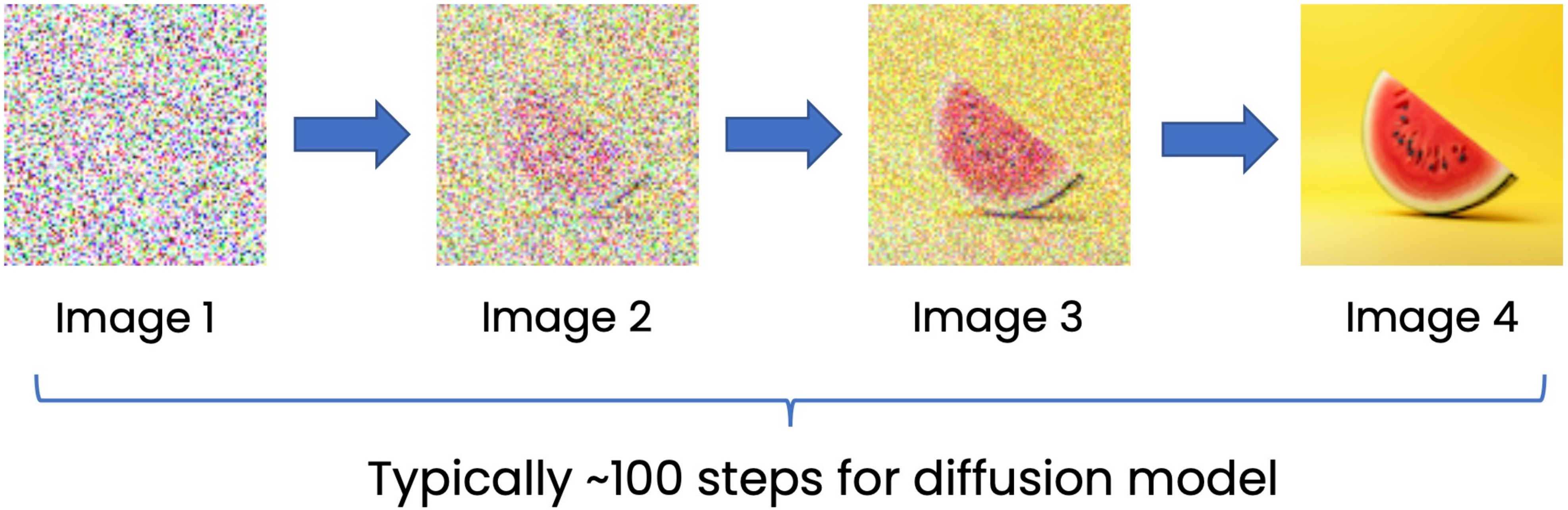
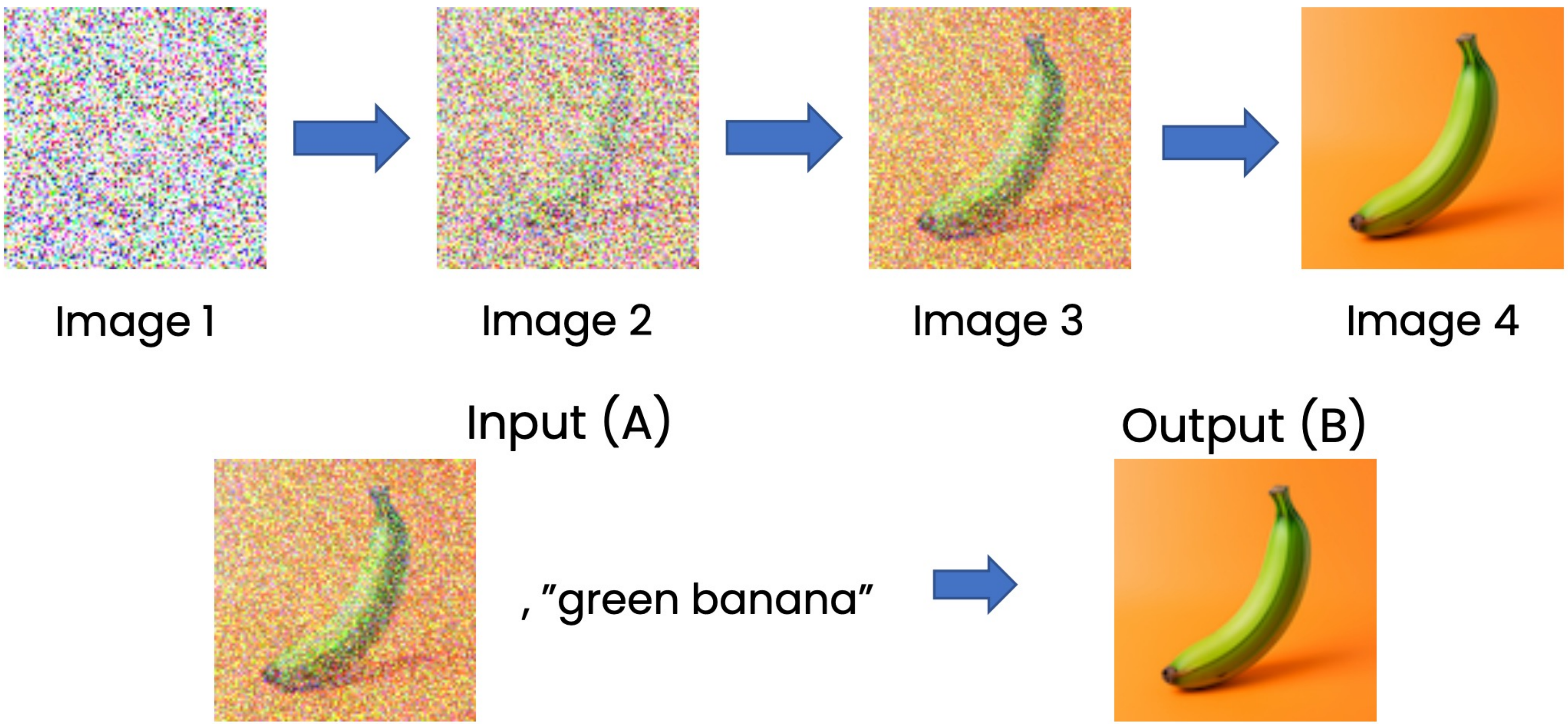
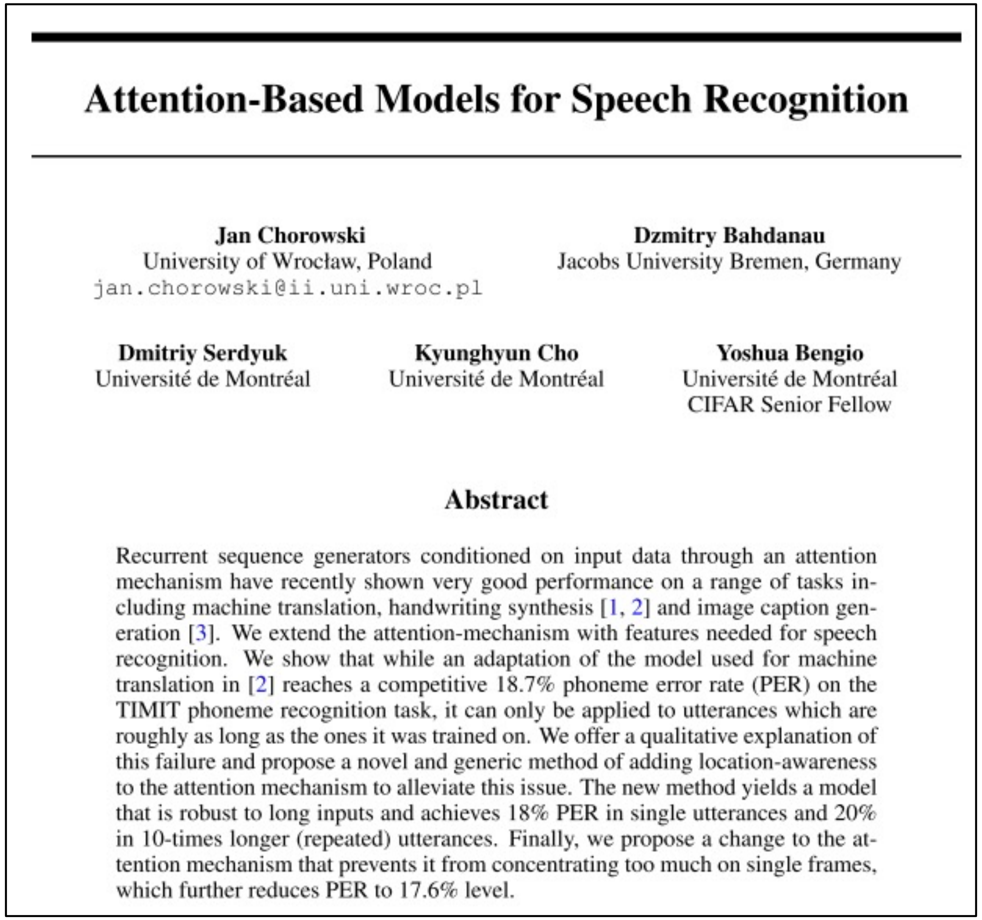


Image generation from Text



Attention Model
[Bengio_2015]



Attention-Based Models for Speech Recognition

Jan Chorowski
University of Wrocław, Poland
jan.chorowski@ii.uni.wroc.pl

Dzmitry Bahdanau
Jacobs University Bremen, Germany

Dmitriy Serdyuk
Université de Montréal

Kyunghyun Cho
Université de Montréal

Yoshua Bengio
Université de Montréal
CIFAR Senior Fellow

Abstract

Recurrent sequence generators conditioned on input data through an attention mechanism have recently shown very good performance on a range of tasks including machine translation, handwriting synthesis [1, 2] and image caption generation [3]. We extend the attention-mechanism with features needed for speech recognition. We show that while an adaptation of the model used for machine translation in [2] reaches a competitive 18.7% phoneme error rate (PER) on the TIMIT phoneme recognition task, it can only be applied to utterances which are roughly as long as the ones it was trained on. We offer a qualitative explanation of this failure and propose a novel and generic method of adding location-awareness to the attention mechanism to alleviate this issue. The new method yields a model that is robust to long inputs and achieves 18% PER in single utterances and 20% in 10-times longer (repeated) utterances. Finally, we propose a change to the attention mechanism that prevents it from concentrating too much on single frames, which further reduces PER to 17.6% level.

In 2015, Bengio 's Model focuses on every phenon's recognition as the combined weights.

$$\alpha_i = Attend(s_{i-1}, \alpha_{i-1}, h)$$

$$g_i = \sum_{j=1}^{L} \alpha_{i,j} h_j$$

$$y_i \sim Generate(s_{i-1}, g_i),$$

h : Input

α_i: Attention Weight

y_i: Output

Chorowski, Jan K., et al. "Attention-based models for speech recognition." *Advances in neural information processing systems* 28 (2015).

Transformer [Vaswani_2017]

In 2017, 8 Google researchers proposed Transformer Neuron Networks based on Attention, which was adopted by ChatGPT.

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com
Noam Shazeer* Google Brain noam@google.com
Niki Parmar* Google Research nikip@google.com
Jakob Uszkoreit* Google Research usz@google.com
Llion Jones* Google Research llion@google.com
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu
Łukasz Kaiser* Google Brain lukaszkaiser@google.com
Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Abstract

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

Cited 66157 (2023/2/21)

Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).

Jakob Uszkoreit proposed replacing RNNs with **self-attention** and started the effort to evaluate this idea.

Noam Shazeer proposed **scaled dot-product attention**, **multi-head attention** and the **parameter-free position representation**.

Transformer

- Transformer is a Deep Learning Model based on Self-Attention
- **Transformer** encodes and decodes data with different weights.
- Examples of **transformer language models include: GPT (GPT-1、GPT-2、GPT-3、ChatGPT) and BERT models (BERT、RoBERTa 、ERNIE).**

Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).

Attention to Transformer

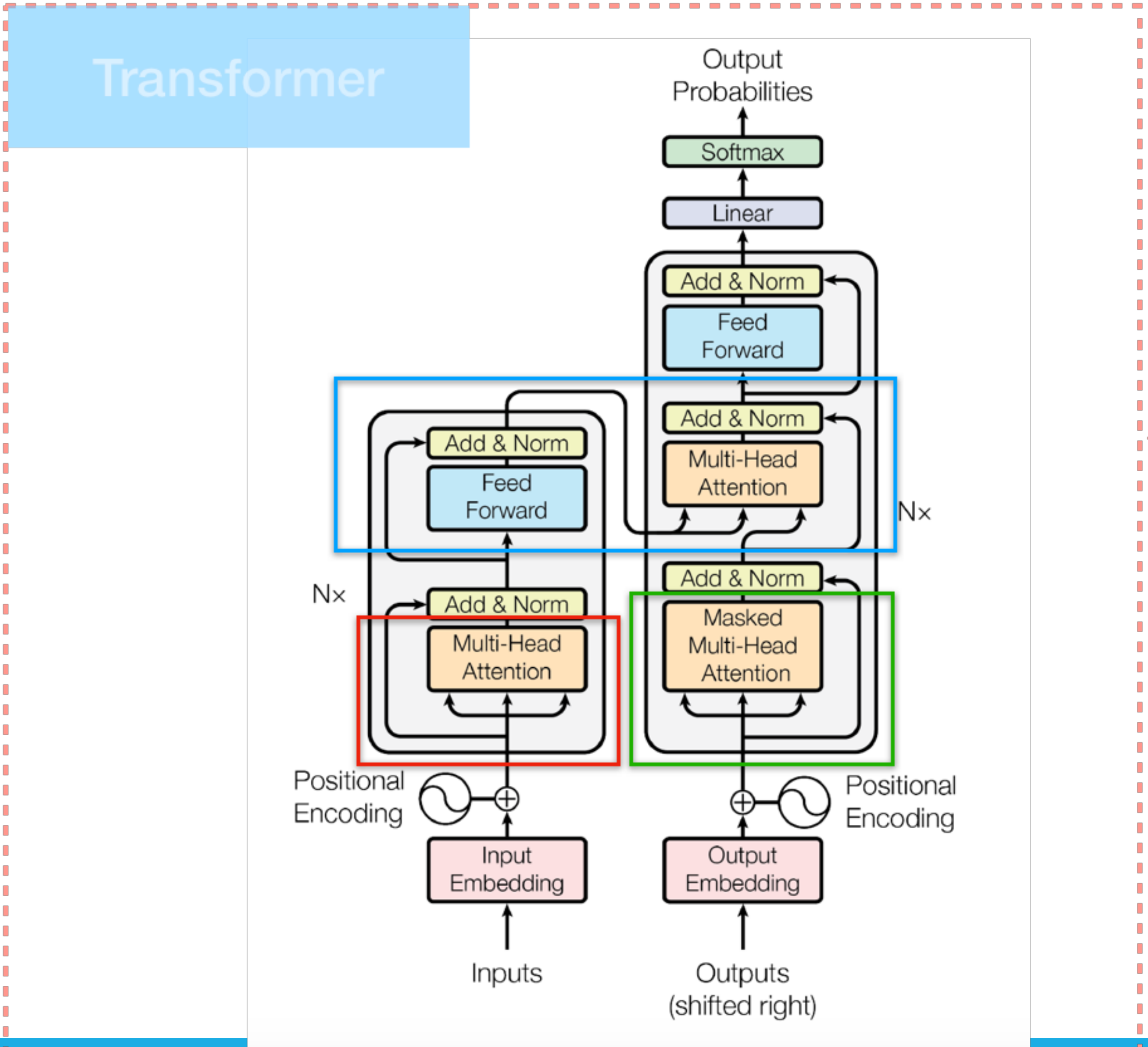


Figure 1: The Transformer - model architecture.

encoder self attention

1. Multi-head Attention
2. Query=Key=Value

decoder self attention

1. Masked Multi-head Attention
2. Query=Key=Value

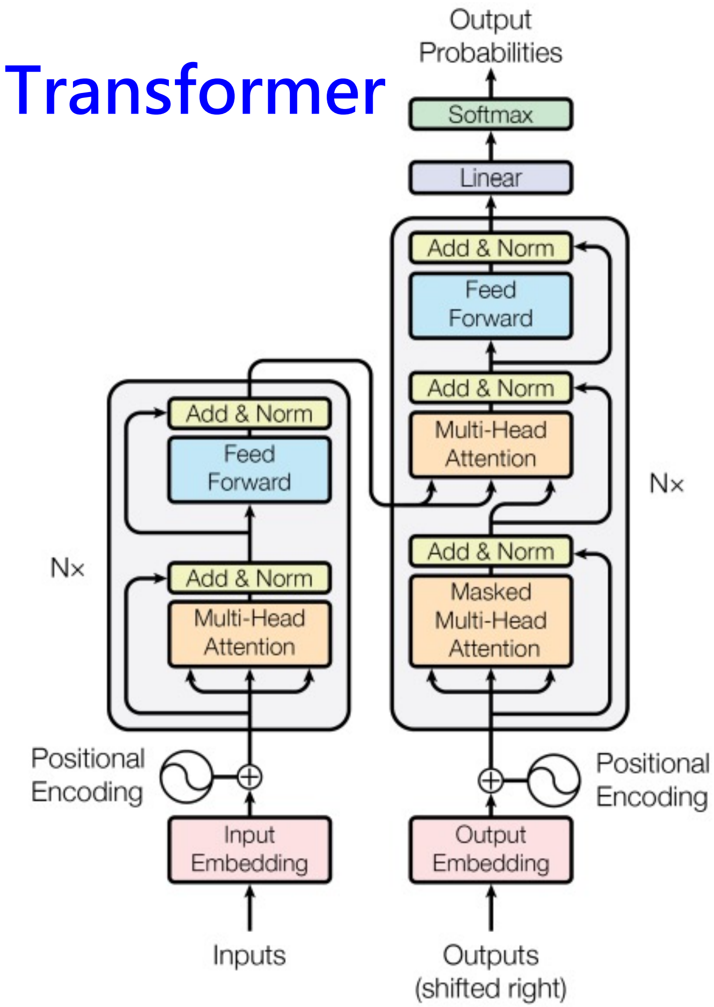
encoder-decoder attention

1. Multi-head Attention
2. Encoder Self attention=Key=Value
3. Decoder Self attention=Query

Transformer

Encoder

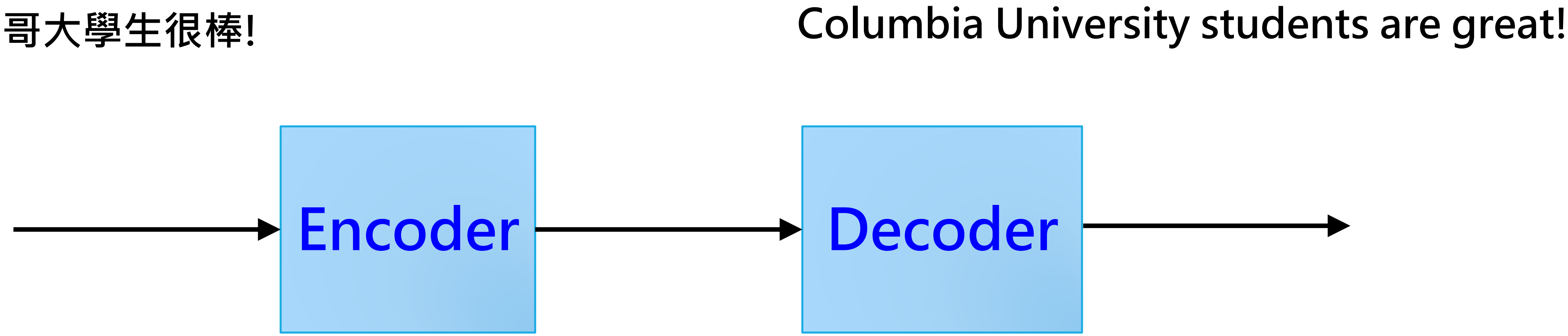
Transformer



Decoder

Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).

Transformer



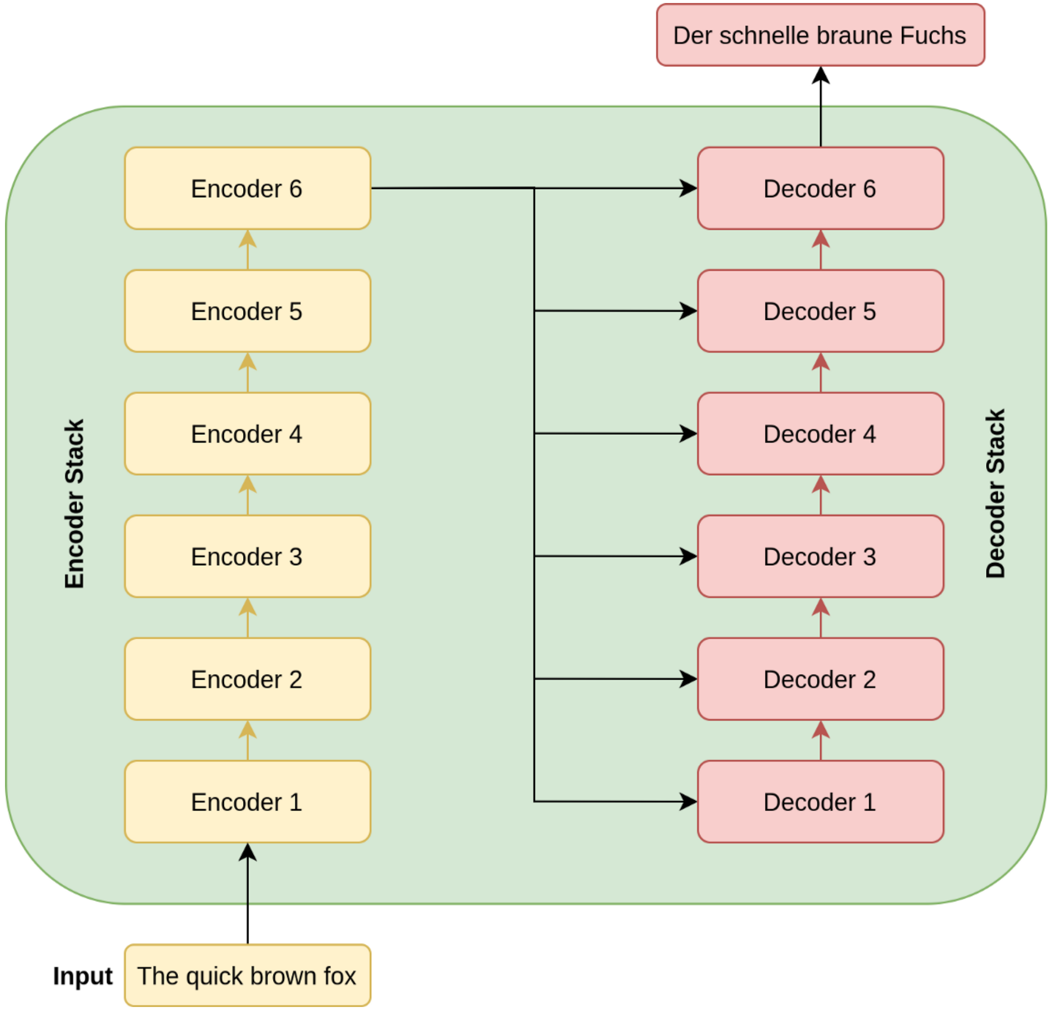
Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).

Transformer
Attention

K

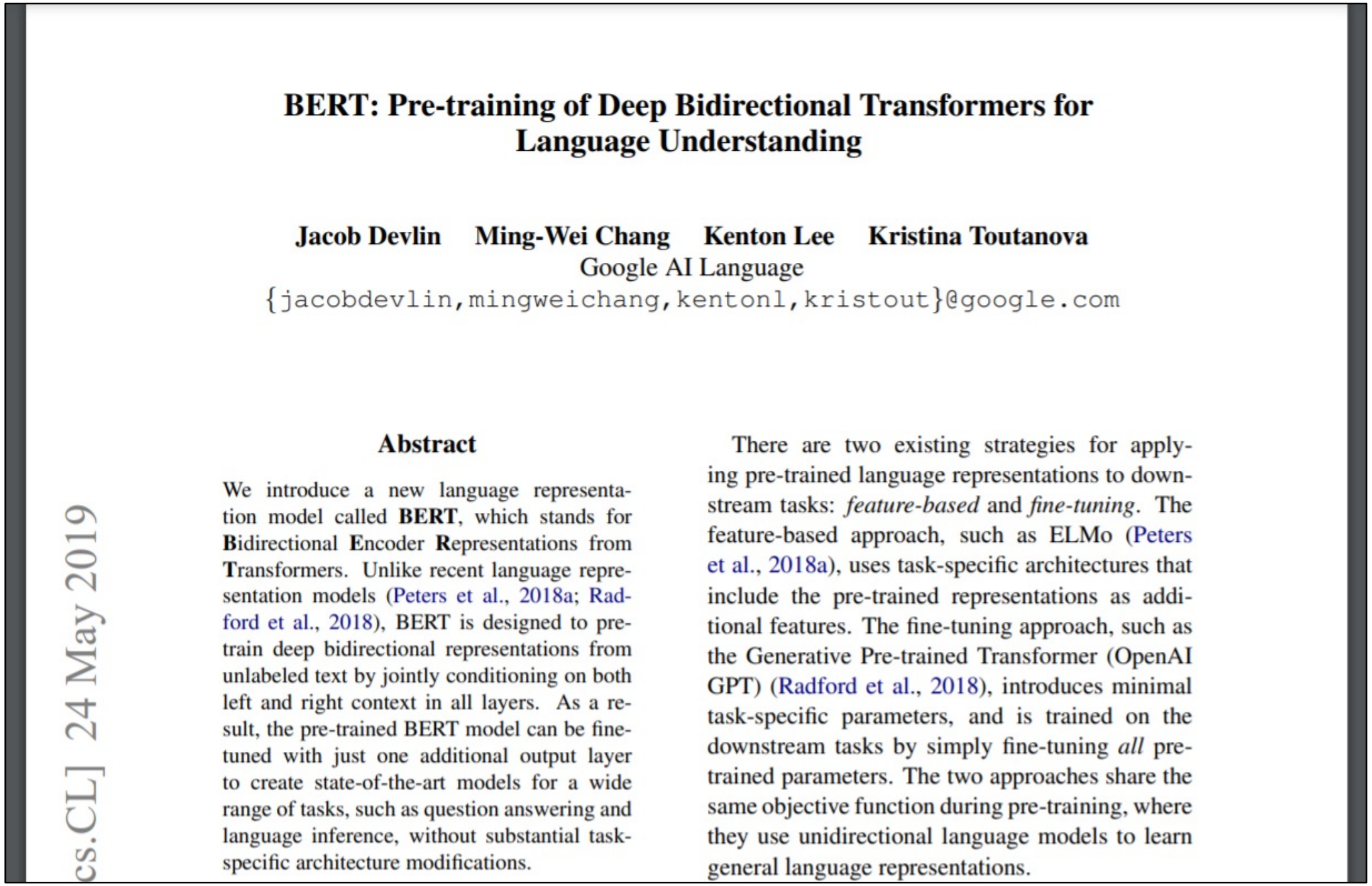
		k_1	k_2	k_3	k_4	k_5	k_6
	weights	Columbia	university	students	are	great	!
Q q_1	哥	1	0.5	0.2	0	0.3	0.2
q_2	大	0.5	1	0.2	0.1	0.3	0.1
q_3	學	0.2	0.2	1	0	0.5	0.2
q_4	生	0.3	0.3	0.8	0.5	0.5	0.6
q_5	很	0	0.1	0	1	0.5	0
q_6	棒	0.3	0.3	0.5	0.5	1	0.8
q_7	!	0.2	0.1	0.2	0	0.8	1

Transformer Translation



Transformer uses 6 layers of encoder and decoder to achieve the same quality of SOTA English-German and English-French translation.

BERT Introduction



BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

cs.CL] 24 May 2019

Abstract

We introduce a new language representation model called **BERT**, which stands for **B**idirectional **E**ncoder **R**epresentations from **T**ransformers. Unlike recent language representation models (Peters et al., 2018a; Radford et al., 2018), BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications.

There are two existing strategies for applying pre-trained language representations to downstream tasks: *feature-based* and *fine-tuning*. The feature-based approach, such as ELMo (Peters et al., 2018a), uses task-specific architectures that include the pre-trained representations as additional features. The fine-tuning approach, such as the Generative Pre-trained Transformer (OpenAI GPT) (Radford et al., 2018), introduces minimal task-specific parameters, and is trained on the downstream tasks by simply fine-tuning *all* pre-trained parameters. The two approaches share the same objective function during pre-training, where they use unidirectional language models to learn general language representations.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

BERT Introduction

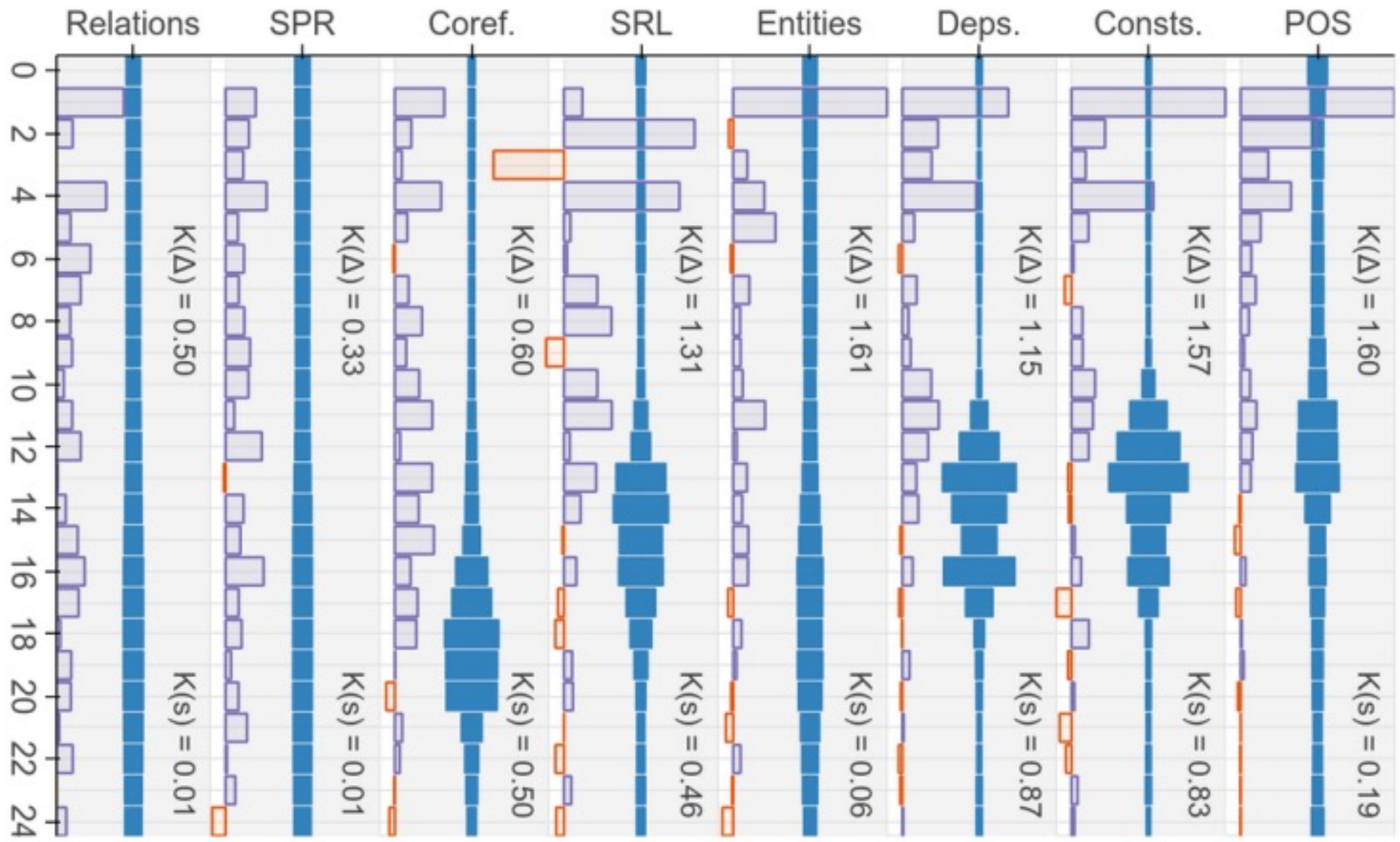
- 2018 Google'BERT has 24 layers of Transformer Encoder
- BERT's original model is based on Wikipedia and booksorpus, using unsupervised training to create BERT.
- At Stanford's Machine Reasoning Test SQuAD1.1 beats human performance.
- Google NLU English was replaced from seq2seq to BERT

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

BERT understands language's meaning

High-Level NLP ⟷ Low-Level NLP

Semantic-Role | Core | Semantic-Role Level | Dependents | Constitutions



Tenney, I., Das, D., & Pavlick, E. (2019). BERT rediscovers the classical NLP pipeline. *arXiv preprint arXiv:1905.05950*.

In 2018, BERT Comprehension test outperformed human

SQuAD1.1 Leaderboard

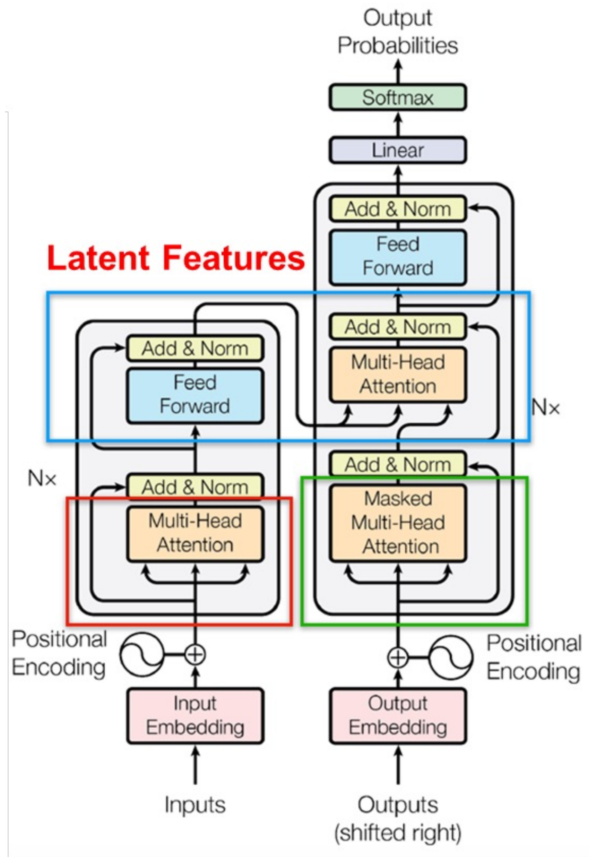
Since the release of SQuAD1.0, the community has made rapid progress, with the best models now rivaling human performance on the task. Here are the ExactMatch (EM) and F1 scores evaluated on the test set of v1.1.

Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar et al. '16)	82.304	91.221
1 Oct 05, 2018	BERT (ensemble) *Google A.I.*	**87.433**	**93.160**
2 Oct 05, 2018	BERT (single model) *Google A.I.*	85.083	91.835
2 Sep 09, 2018	nlnet (ensemble) *Microsoft Research Asia*	85.356	91.202
2 Sep 26, 2018	nlnet (ensemble) *Microsoft Research Asia*	85.954	91.677

Transformer to GPT

Transformer

Input -> **Encoder** -> Latent Feature + Masked Output -> **Decoder** -> Output



An [redacted] a day keeps the doctor away

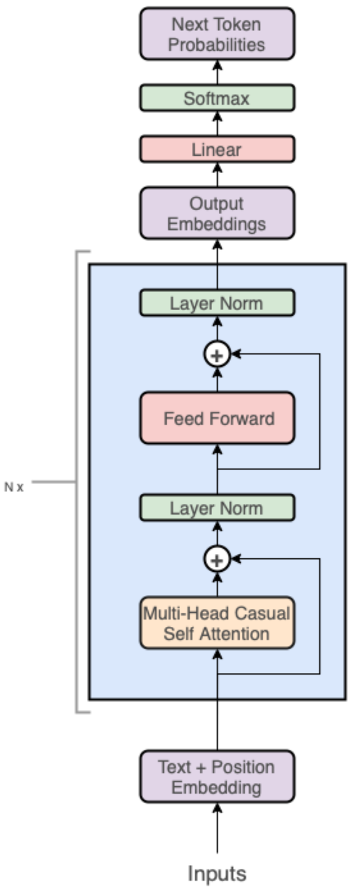
apple 95%
banana 5%

Masked Language Learning

An apple a day keeps the doctor away

GPT

Input -> **Decoder(with Casual mask)** -> shift Output



An

apple 99%
almond 1%

An apple

a 99%
watch 1%

An apple a

Autoregressive Learning

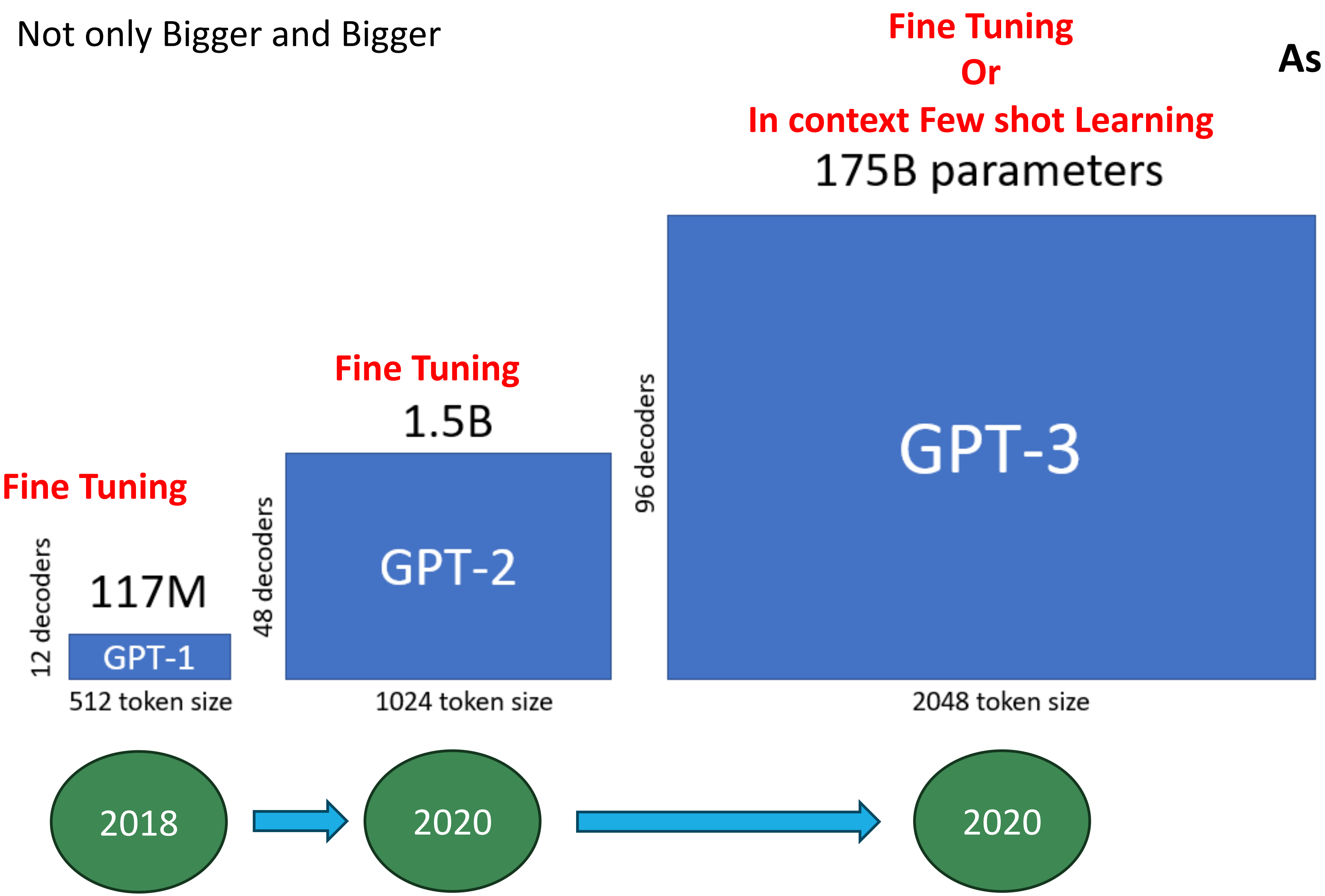
ChatGPT

Software dev job	**ChatGPT would be hired as L3 Software Developer at Google: the role pays $183,000/year.**
Politics	**ChatGPT writes several Bills (USA).**
MBA	**ChatGPT would pass an MBA degree exam at Wharton (UPenn).**
Accounting	**GPT-3.5 would pass the US CPA exam.**
Legal	**GPT-3.5 would pass the bar in the US.**
Medical	**ChatGPT would pass the United States Medical Licensing Exam (USMLE).**
AWS certificate	**ChatGPT would pass the AWS Certified Cloud Practitioner exam.**
IQ (verbal only)	**ChatGPT scores IQ=147, 99.9th %ile.**
SAT exam	**ChatGPT scores 1020/1600 on SAT exam.**

https://lifearchitect.ai/chatgpt/

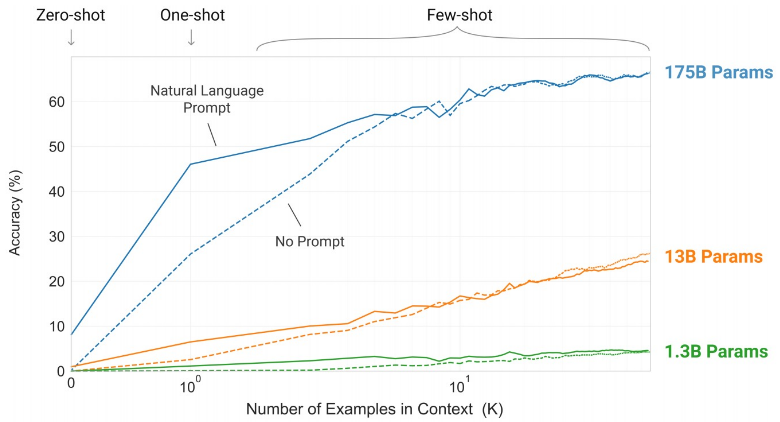
GPT Evolution

Not only Bigger and Bigger



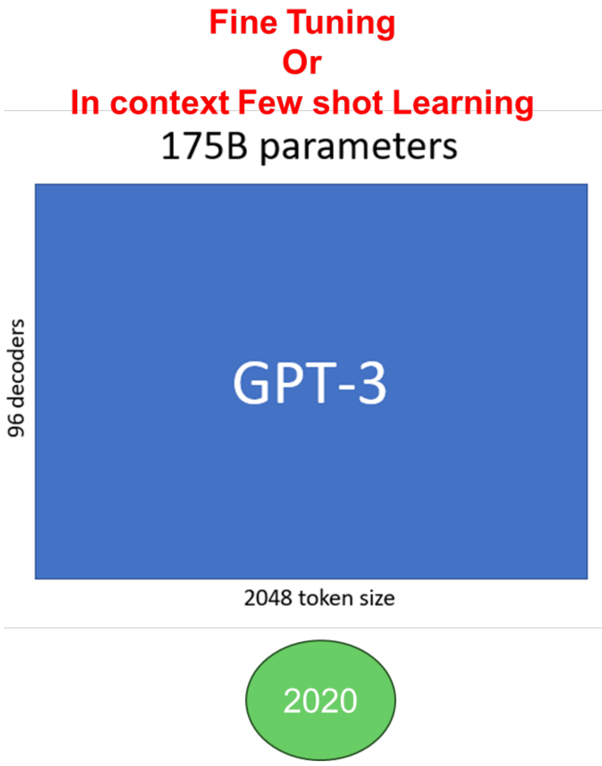
As the model and dataset get larger, it will know more and more

"GPT-3 is applied without any gradient updates or fine-tuning, with tasks and few-shot demonstrations specified purely via text interaction with the model."
From **Language Models are Few-Shot Learners (2020)**



GPT Evolution

Not only Bigger and Bigger

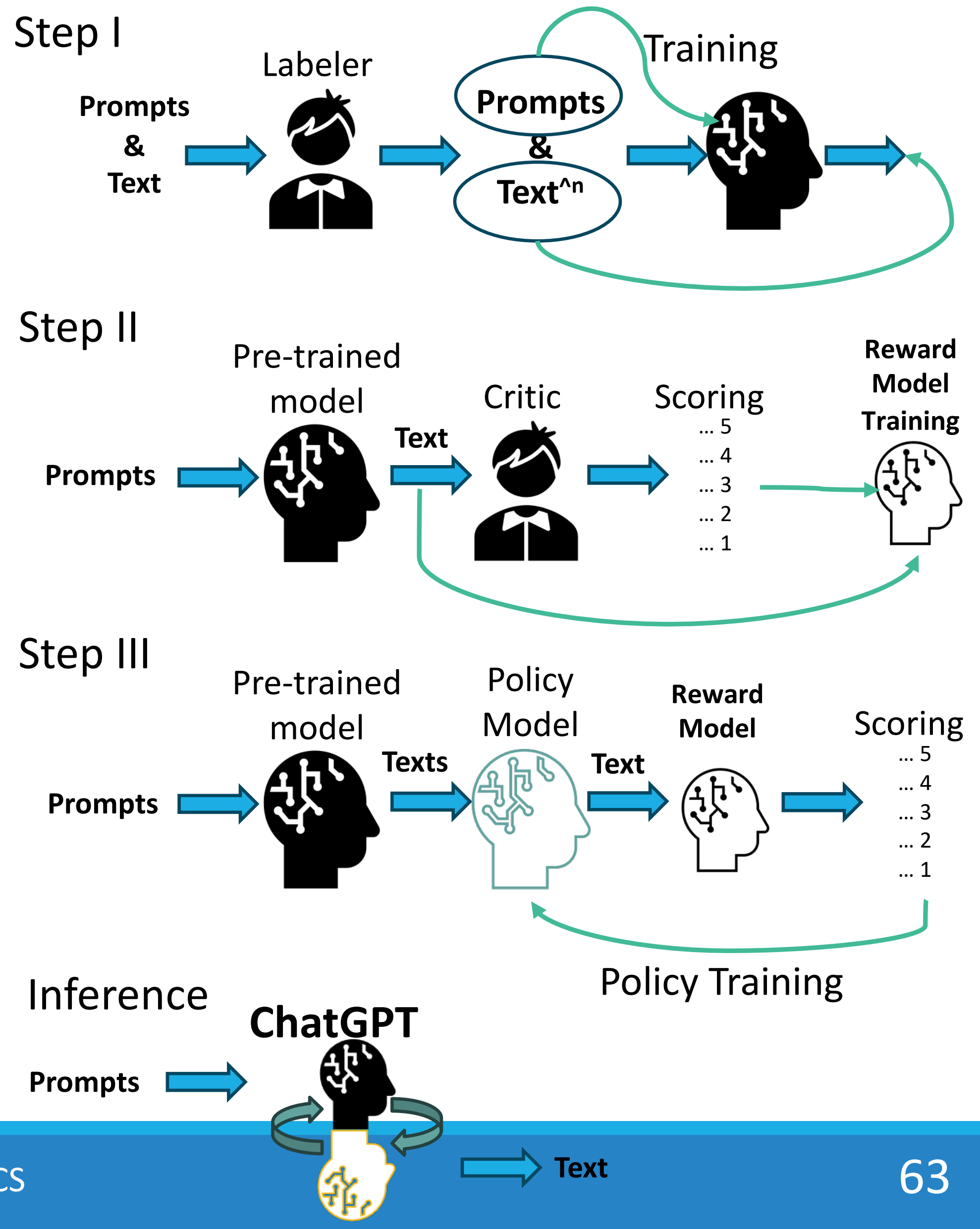


?

How does the Model Answer smartly or more like an Adult human

Thinking and Answering policy optimization

Reinforcement Learning from Human Feedback (RLHF)



What is Next ?

AutoGPT

What goal do you want

Connect to Internet to Find the way or answer

Write code and debug

Do research and try & error

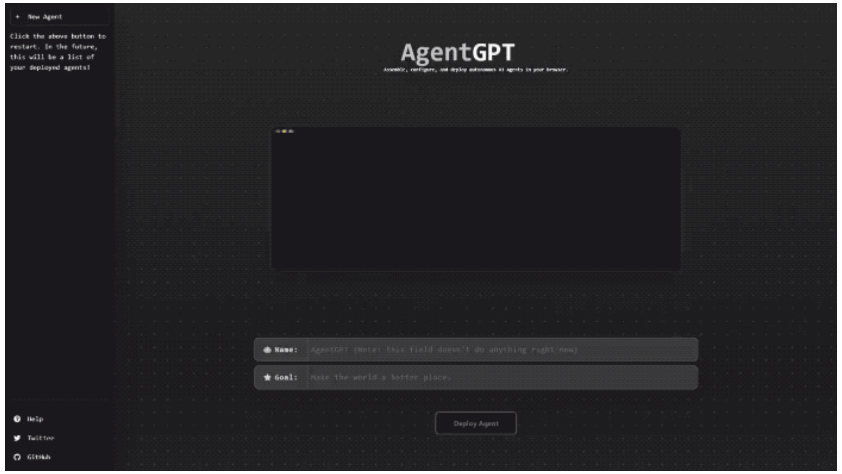
Make a Hypothesis and provide it

Summarization and Organization

Like JARVIS

In Iron Man

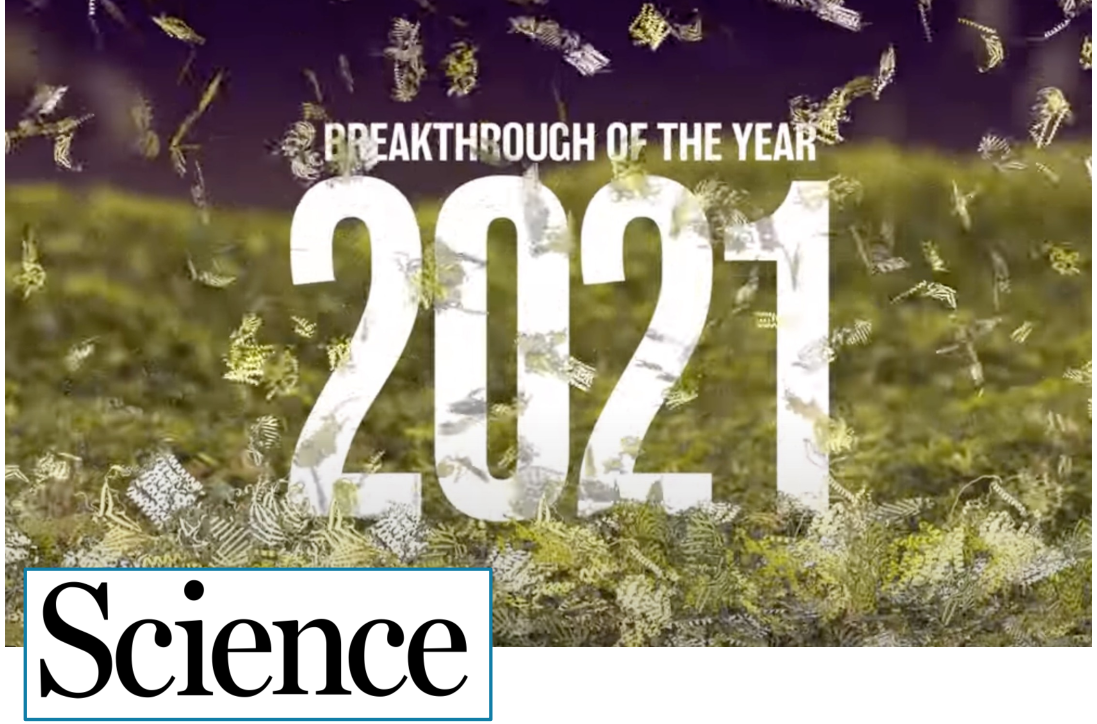
The no longer future will come true

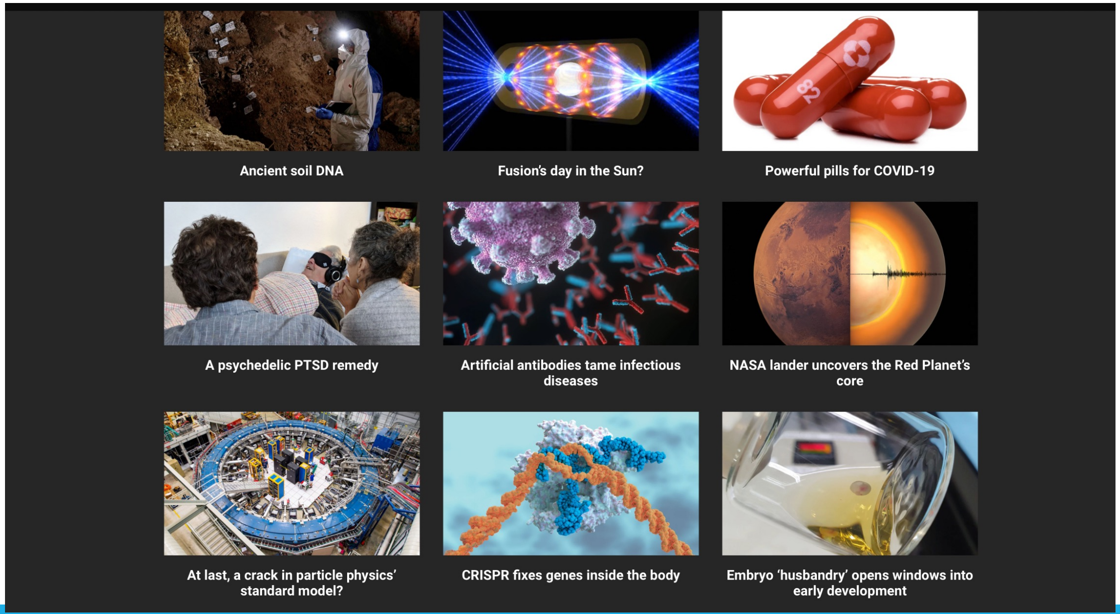


https://agentgpt.reworkd.ai/zh

A.I. for Drug Design

-- New Era is Now. Significant Scientific and Business Breakthrough





Protein == Biological Machine

https://youtu.be/iUMpm3tYsVE?t=218

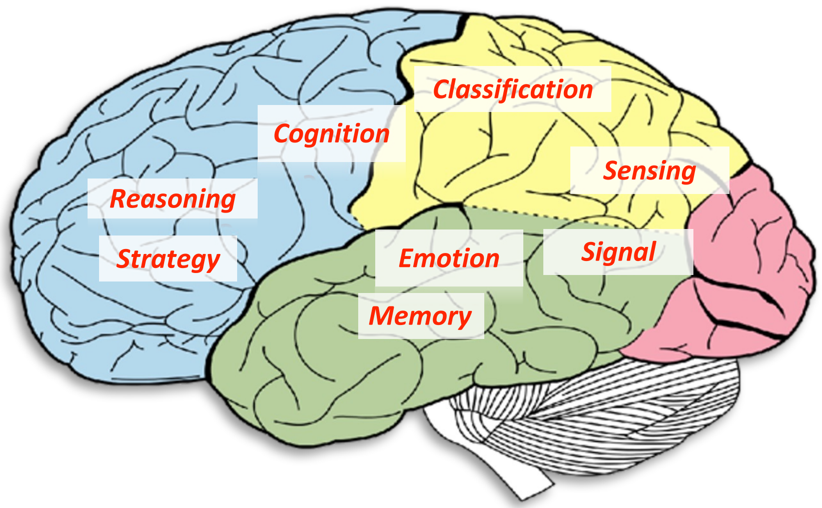
When people in 50 years later look back history, they may recognize 'Today' is a scientific breakthrough moment in Biology as an equivalent of Newton's moment in Physics

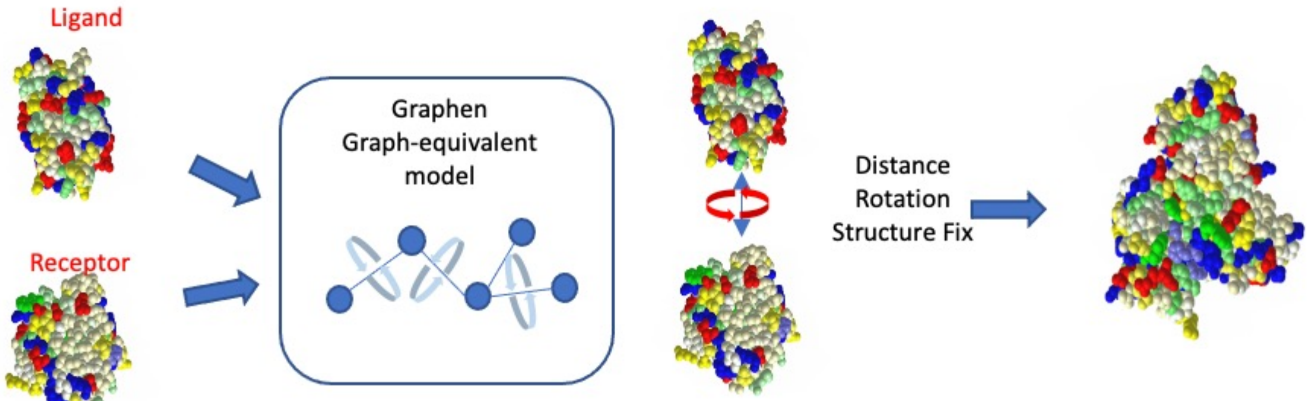
-- Graphen 2023

ABIresearch
for visionaries

*"Tools from established companies like **Google** DeepMind, startups like **Graphen**, and AI chipsets from vendors like **NVIDIA** and **Intel** will help accelerate the speed of drug discovery, development, and testing, allowing pharmaceutical companies and healthcare authorities to combat the pandemic." – ABI research, May 2020*

Graphen Ardi Full-Brain Platform's Graph Models enable Graphen Atom Tools that better simulate biological functions



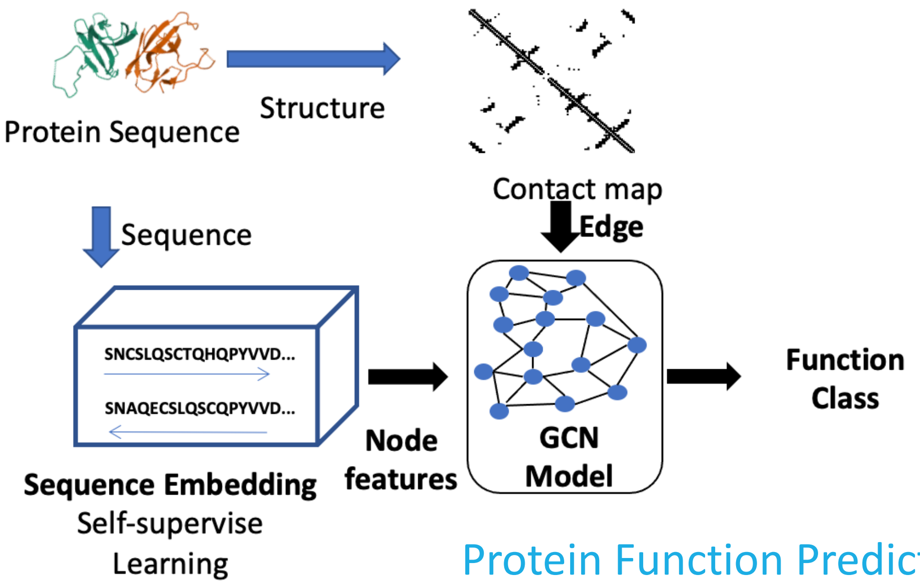


Atom Network Modeling

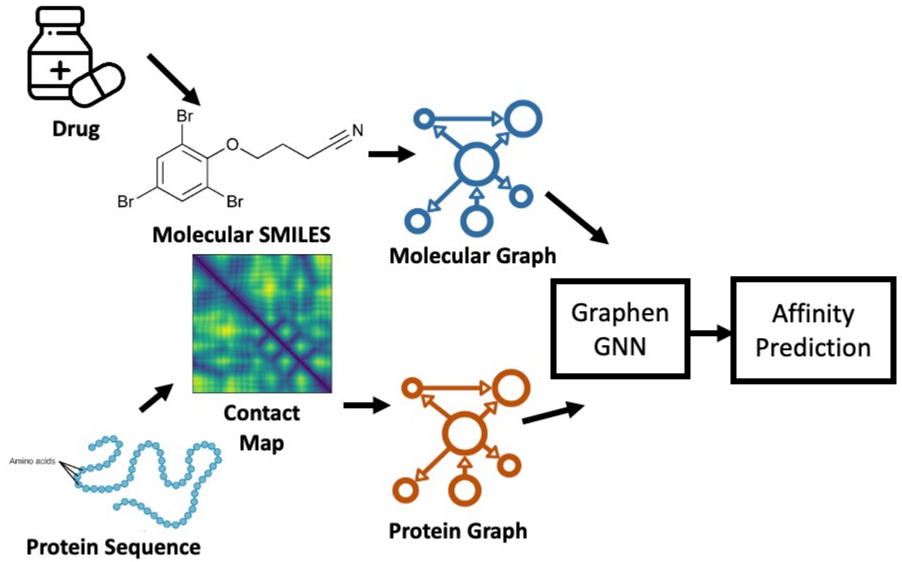
https://www.graphen.ai/products/ardi.htm

Ardi Graph Analytics and Database of zillions of nodes and edges were deployed in one of the world's largest institutes in 2018.

Graphen's Differentiator:
Graph Modeling + Deep Learning + Generative AI



Protein Function Prediction

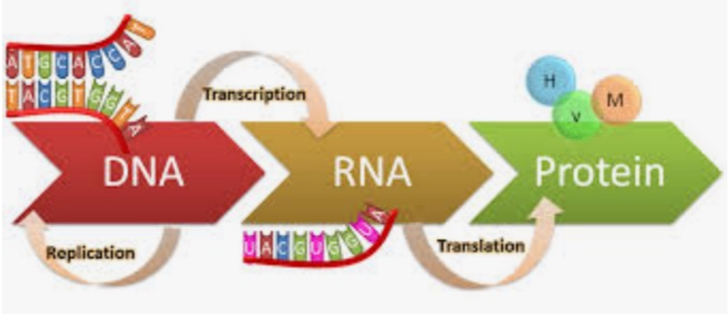


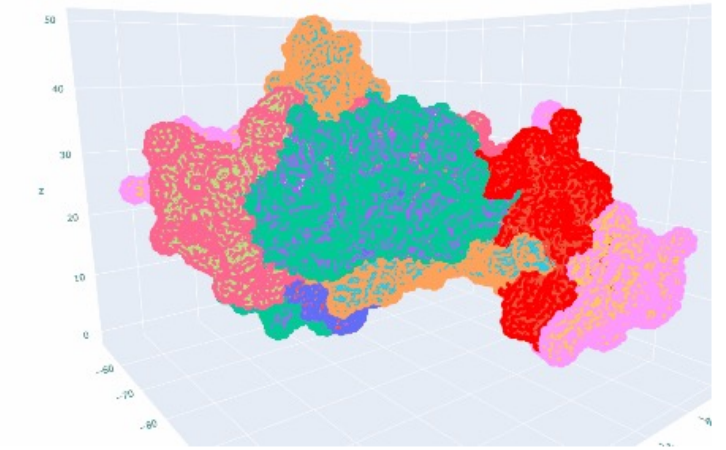
Affinity Prediction

Graphen Small Mole Drug Dev ➔ 1/27 of the Time; 1/9000 of the Cost, comparing to traditional methods

1. Generate Synthesis and low side effects Drug
2. Filtering drug by ADMET and Solubility
3. Filtering Drug by Graphen QF Energy model and Kinase model
4. Disease-Clinical/Cell Target mapping (Spectrum Mutant Prediction)
5. Synthesis and Wet Lab Evaluation

	Samples	Required Time
	1.5M	
Generate de novo new compound	30K	7 Days
Manufacturable and Safety Screening	2.5K	3 Days
Dynamic Target-Lead interaction simulation	0.1K	5 Days
Target Cell and Disease Finding	30	5 Days
Chemical Synthesis Kinase Assay/Cell Assay	10	30+30 Days





AI is making a paradigm shift to reduce the risk of drug development via:

- **Enrich pipeline**
- **Increase Probability of Success**
- **Cost Reduction**
- **More Precise, Fewer Side Effects**
- **Rare Disease & Personalized Drugs**