

LARGE LANGUAGE MODELS

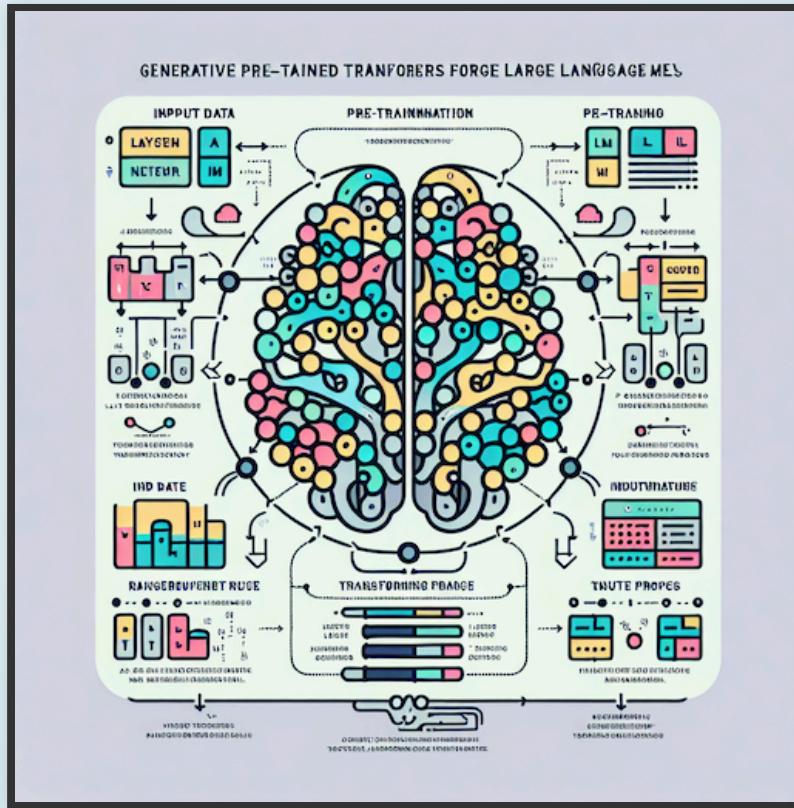
LARGE LANGUAGE MODELS



LARGE LANGUAGE MODELS

- Example LLMs: ChatGPT, Gemini, Llamma, Claude, capable of understanding and generating human-like language, images, music.
- Data scope: Trained on massive datasets, including PetaBytes of internet text, Wikipedia and Pubmed.
- Applications: LLMs are utilized in chatbots, text generation, reasoning and problem solving, creative output.

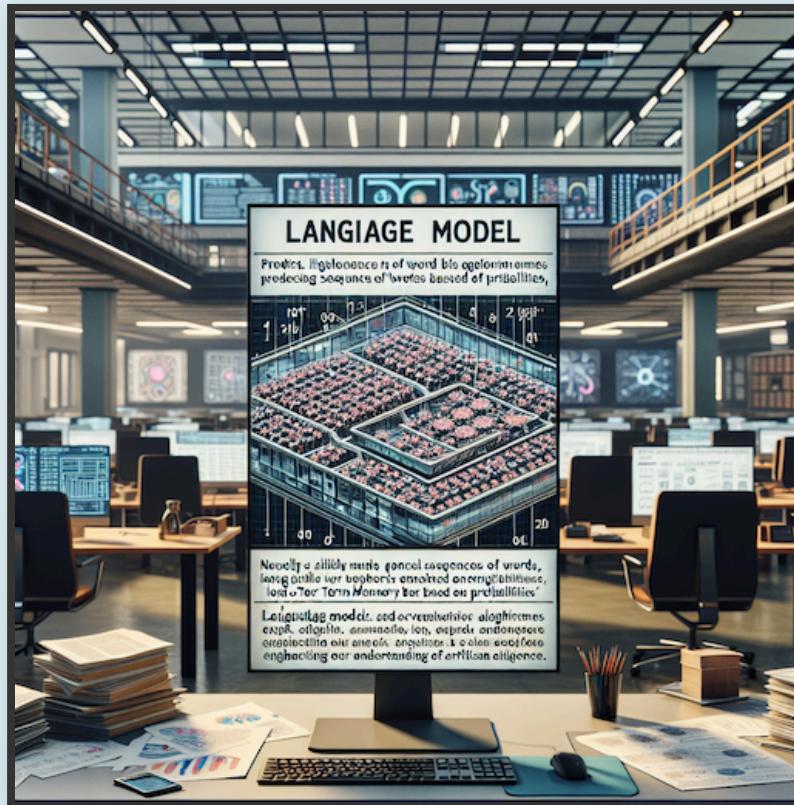
ARTIFICIAL NEURAL NETWORKS



ARTIFICIAL NEURAL NETWORKS

- Scalability: LLMs are based on Generative Pre-trained Transformers (GPTs), can be "prompt-engineered" for complex tasks.
- Definition: Artificial Neural Networks (ANNs), are fundamentally complex non-linear function estimators - pattern classifiers.
- Innovation: GPTs are ANNs that implement "multi-head attention", enabling capture of long-range patterns in training data, emergent "intelligence".

GENERATION AND INFERENCE



GENERATION AND INFERENCE

- Text Generation: LLMs create responses by predicting likely sequences of words, based on billions of probabilities.
- Inference Techniques: The models use sophisticated algorithms to generate text that aligns with context and user input.
- Diversity: Can produce a wide range of responses, from factual information to hallucinations and "deepfakes".

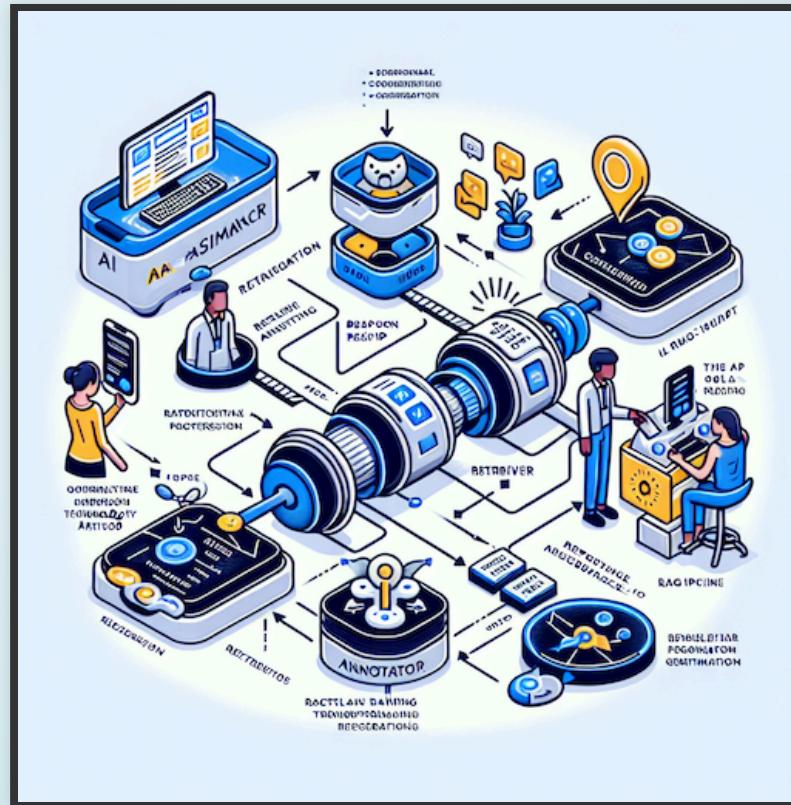
FINE-TUNING



FINE-TUNING

- Fine-Tuning: LLMs can be fine-tuned with data from specific domains, enhancing their relevance and performance.
- Task-Specific: Fine-tuning produces tailored AI models for specialized applications, i.e. bioinformatics / biomedical research.
- Alternative: Fine-tuning costs computing time, instead similar achievements via carefully designed prompt-engineering.

OPENAI ASSISTANTS



OPENAI ASSISTANTS

- Clarification: OpenAI (the company behind ChatGPT) offers rich functionality through their API
- Assistants: user file search and code Interpreter, external API function calling by the AI
- Functionality: build custom AI applications around user's data

GPT FOR BCO

The screenshot shows the OpenAI Playground interface for generating BioCompute Objects (BCOs). The left sidebar shows the 'Default Project' with sections like Playground, Chat, Assistants, Compare, Completions, Assistants, Fine-tuning, Storage, Usage, API keys, Settings, Docs, Help, and Ag Biotech. The main area is titled 'Playground' and shows a 'BCO_v2' thread. The 'Instructions' section contains the text: 'You specialize in generating a BioCompute Object (BCO), which is a standardized description of a bioinformatics data analysis pipeline.' The 'Model' section is set to 'gpt-3.5-turbo'. The 'TOOLS' section includes a 'File search' tool with a 'Vector store for BCO_v1' file (60 KB) and a 'Code interpreter' tool with files 'parametric_domain.json', 'execution_domain.json', and 'io_domain.json'. A message input field at the bottom says 'Enter your message...' with a 'Run' button.

Default Project
Personal

Playground Chat Assistants Compare Completions Assistants Fine-tuning Storage Usage

API keys Settings Docs

Help Ag Biotech

Playground Assistants

BCO_v2

THREAD

Instructions

You specialize in generating a BioCompute Object (BCO), which is a standardized description of a bioinformatics data analysis pipeline.

Model

gpt-3.5-turbo

TOOLS

File search + Files
Vector store for BCO_v1 vs_x101UE91jbQJoqA3aF5sFeJM 60 KB

Code interpreter + Files
parametric_domain.json
execution_domain.json
io_domain.json

Enter your message... Run

Updated 4/19, 5:06 PM

Playground messages can be viewed by anyone at your organization using the API.

PROMT ENGINEERING

The screenshot shows the OpenAI Platform interface. On the left, a sidebar menu includes 'Default Project' (Personal), 'Playground', 'Assistants' (selected), 'Fine-tuning', 'Storage', 'Usage', 'API keys', 'Settings', and 'Docs'. The main area is titled 'Assistants' and shows a list of interactions:

- 5 days ago, Apr 19: BCO_v2 (asst_OU05P9PNeEKIBbFqjkfqaRL) - 4:30 PM
- 10 days ago, Apr 14: BCO_v1 (asst_6yICYnIQ1eAs9PD3L9rGfGVO) - 6:26 PM

The right side of the screen displays the details of the most recent interaction with 'BCO_v2':

ASSISTANT
asst_OU05P9PNeEKIBbFqjkfqaRL

[Playground ↗](#)

[Edit instructions](#)

You specialize in generating a BioCompute Object (BCO), which is a standardized description of a bioinformatics data analysis pipeline.

The definition of the various BCO domains with examples can be found in the .md files I have provided. For example the file provenance-domain.md, contains examples and the definition of the BCO provenance domain, and similarly for the other domains. There is an .md file for each domain.

The BCOs are written in JavaScript Object Notation (JSON) files, following a JSON standardized schema. The JSON schema definitions for the BCO domains can be found in the files that I gave you and have extension .json.

For example the file provenance-domain.json, contains the JSON schema definition for the provenance domain, and similarly for the other domains. There is an .json file for each domain.

Your task is to generate the JSON with the BCO domains, following strictly the JSON schema from the .json files.

EXAMPLE PUBLICATION

Astling *et al.* BMC Genomics (2017) 18:614
DOI 10.1186/s12864-017-3976-z

BMC Genomics

METHODOLOGY ARTICLE Open Access

 CrossMark

High resolution measurement of DUF1220 domain copy number from whole genome sequence data

David P. Astling¹, Illea E. Heft¹, Kenneth L. Jones² and James M. Sikela^{1*}

Abstract

Background: DUF1220 protein domains found primarily in Neuroblastoma BreakPoint Family (*NBPF*) genes show the greatest human lineage-specific increase in copy number of any coding region in the genome. There are 302 haploid copies of DUF1220 in hg38 (~160 of which are human-specific) and the majority of these can be divided into 6 different subtypes (referred to as clades). Copy number changes of specific DUF1220 clades have been associated in a dose-dependent manner with brain size variation (both evolutionarily and within the human population), cognitive aptitude, autism severity, and schizophrenia severity. However, no published methods can directly measure copies of DUF1220 with high accuracy and no method can distinguish between domains within a clade.

BCOS VIA GPT

Playground Assistants

New features launched in the Assistants API ↗

BCO_v2

THREAD thread_fP0Fv8aWDaqJxUian0UtIKer 22057 tokens 1 Logs

Name: BCO_v2
asst_0U05P9PNeEKIBbFqjkfqnaRL

Instructions:
You specialize in generating a BioCompute Object (BCO), which is a standardized description of a bioinformatics data analysis

Model: gpt-3.5-turbo

TOOLS

File search + Files
Vector store for BCO_v1 vs_x01UE91jbQJoqA3af5sFeJM 60 KB

Code interpreter + Files
parametric_domain.json
execution_domain.json
io_domain.json
usability_domain.json
provenance_domain.json
description_domain.json
error_domain.json

Functions + Functions

MODEL CONFIGURATION

Response format: JSON object

Temperature: 0.25

User
Search in the attached file for all bioinformatics data analysis operations. Write what you find in json format, following the json schema that can be found in the description_domain.json.

DUF1220.pdf

BCO_v2
Based on the information found in the provided document, here is a JSON representation of the bioinformatics data analysis operations, following the schema defined in the description_domain.json file:

```
1 {
2   "spec_version": "https://w3id.org/biocompute/1.3.0/",
3   "object_id": "https://example.com/bco/9487ae7e-c1aa-4a3c-b18f-3d3695b",
4   "type": "bioinformatics_data_analysis",
5   "etag": "584C7FE128717E1712426AB19CAAE8BC1E27365B54285BBEA1221284C7D",
6   "description_domain": {
7     "external_references": [
8       {
9         "description": "High resolution measurement of DUF1220 domain",
10        "url": "https://example.com/DUF1220_analysis"
11      }
12    ],
13    "pipeline_steps": [
14      {
15        "step_name": "Identification of DUF1220 domains",
16        "description": "Utilized HMMER and a published pipeline to identify DUF1220 domains in the genome",
17        "input_objects": ["hg38 reference genome"],
18        "output_objects": ["List of genomic coordinates for DUF1220 domains"]
19      },
20      {
21        "step_name": "Characterizing the read alignment ambiguity",
22        "description": "Performed detailed annotation of NBPF genes and their variants across the identified DUF1220 domains",
23        "input_objects": ["DUF1220 domains", "NBPF genes sequence"],
24        "output_objects": ["Detailed sequence analysis and clustering results for NBPF genes"]
25      }
26    ],
27    "io_relationships": [
28      ...
29    ]
30  }
31}
```

Enter your message...

BCOS VIA GPT

The screenshot shows the AI Playground interface with the following details:

- Playground** tab selected.
- Assistants** dropdown.
- New features launched in the Assistants API** button.
- Learn more** button.
- THREAD** `thread_fPOFv8aWDaqJxU1anOUtIKeR`
- 87615 tokens**
- Logs** button.

BCO_v2 configuration:

- Name:** BCO_v2
asst_0U05P9PNeEKIBbFqjkfqnaRL
- Instructions:** You specialize in generating a BioCompute Object (BCO), which is a standardized description of a
- Model:** gpt-3.5-turbo
- TOOLS:**
 - File search**: Vector store for BCO_v1 vs_xi0iUE91jbQJoqA3aF5sFeJM (60 KB)
 - Code interpreter**: parametric_domain.json, execution_domain.json, io_domain.json, usability_domain.json, provenance_domain.json, description_domain.json, error_domain.json
- Functions**: + Functions

Text Input:

For the bioinformatics data analysis operations you identified, find the name for each software that performs the data analysis. It can be a .sh script, python or perl script, or other executable. Find the the reference in the text for where the software is available, for example it can be a github repository or other website. Print what you find in json format, following the json schema that can be found in the execution_domain.json.

BCO_v2 Response:

Based on the search results, I will now compile the information to provide you with the JSON representation of the software names and their references for the bioinformatics data analysis operations.

BCO_v2 JSON Output:

```
1 {  
2     "execution_domain": {  
3         "script": [  
4             {  
5                 "uri": {  
6                     "uri": "https://github.com/dpastling/plethora"  
7                 }  
8             },  
9             {  
10                 "uri": {  
11                     "uri": "https://github.com/IleaHeft/DUF1220"  
12                 }  
13             }  
14         ]  
15     }  
16 }
```

BCOS VIA GPT

The screenshot shows the Anthropic Playground interface. On the left, the sidebar displays the configuration for a project named "BCO_v2". The "Name" field contains "BCO_v2" and "asst_0U05P9PNeEKIBbFqjkfqnRL". The "Instructions" section contains a placeholder text: "You specialize in generating a BioCompute Object (BCO), which is a standardized description of a bioinformatics data analysis". The "Model" dropdown is set to "gpt-3.5-turbo". Under the "TOOLS" section, there are two active tabs: "File search" (with a file named "Vector store for BCO_v1") and "Code interpreter" (with files including "parametric_domain.json", "execution_domain.json", "io_domain.json", "usability_domain.json", "provenance_domain.json", "description_domain.json", and "error_domain.json"). Below these are sections for "Functions", "MODEL CONFIGURATION", "Response format" (set to "JSON object"), and "Temperature" (set to 0.25). On the right, the main workspace is titled "THREAD thread_fPOFv8aWDaqJxU1anOUtIKeR" with "22057 tokens" used. It shows a "User" instruction: "For the bioinformatics data analysis operations you identified, find the parameters for running each command line tool. Write the parameters in json format, following the json schema that can be found in the parametric_domain.json file." Below this, the "BCO_v2" section provides the JSON representation of the parameters. A large code block on the right shows the generated JSON code:

```
1 {
2     "parametric_domain": [
3         {
4             "param": "HMMER_seed",
5             "value": "14",
6             "step": "1"
7         },
8         {
9             "param": "HMMER_minimum_match_len",
10            "value": "66",
11            "step": "1"
12        },
13        {
14            "param": "Sequence_analysis_divergence_threshold_percent",
15            "value": 0.30,
16            "step": "1"
17        },
18        {
19            "param": "Annotation_minimum_coverage",
20            "value": "15",
21            "step": "2"
22        },
23        {
24            "param": "Clustering_freq_cutoff",
25            "value": 0.10,
26            "step": "2"
27        }
28    ]
}
```

At the bottom, there is a text input field labeled "Enter your message...".

SUMMARY & CONCLUSIONS

- Strong NLP capabilities of GPTs, good results.
- Iterative training, prompting with canonical BCO.
- Fine tuning with BCO json - text chunks dataset.

THANK YOU!