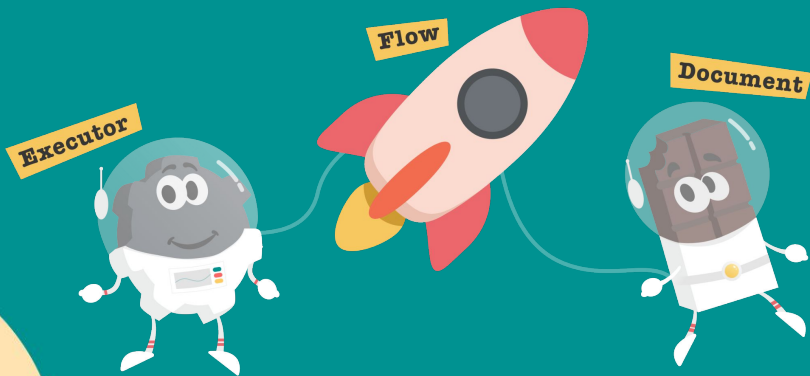


从语音转图像说起，生成式任务的前世今生

付杰

Jina AI 高级算法工程师



CONTENT

01 生成式任务简介

02 语音转图像任务

03 使用 Jina 快速搭建工作流

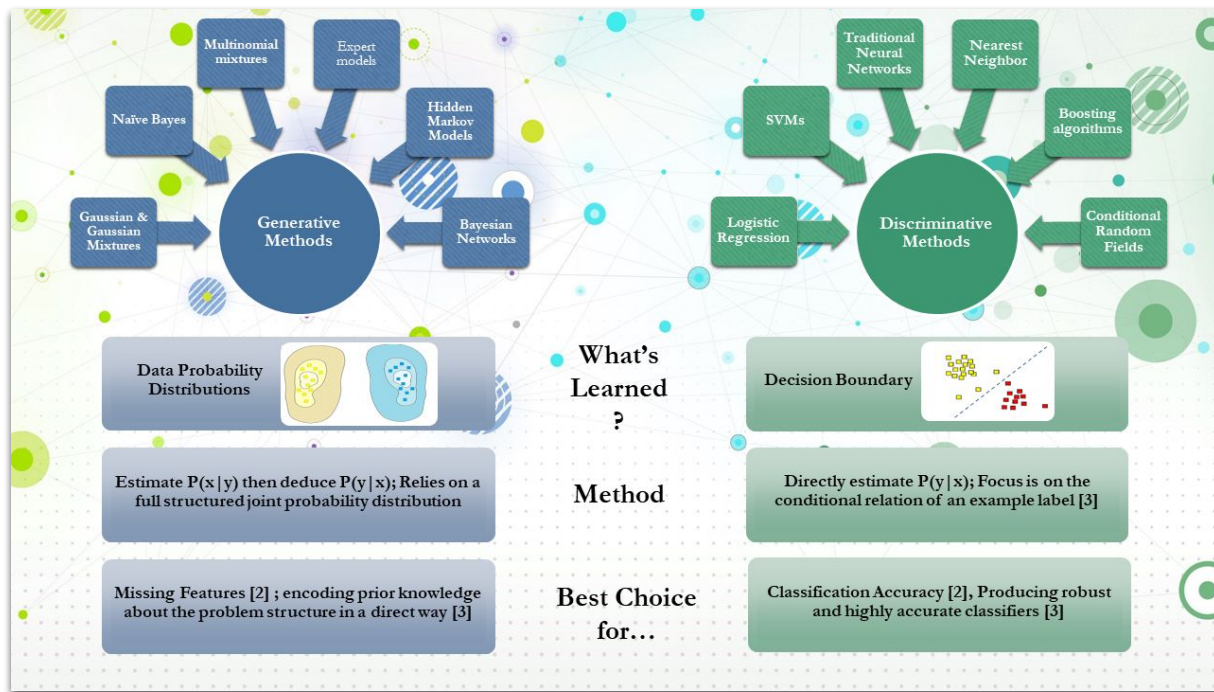
04 Demo 上手

生成式任务简介

01

生成式任务简介

- 判别式任务: 分类 / 回归, 从现有数据出发进行分析
- 生成式任务: 学习已有的数据分布, 生成全新的数据



生成式任务简介

文心一言



ChatGPT



MI Can you write a function that takes a string input and capitalizes the first letter of each word?



Sure, here is a possible function that capitalizes the first letter of each word in a string input:

Copy code

```
def capitalize_first_letters(string):  
    words = string.split()  
    capitalized_words = []  
    for word in words:  
        capitalized_word = word[0].upper() + word[1:]  
        capitalized_words.append(capitalized_word)  
    return " ".join(capitalized_words)
```

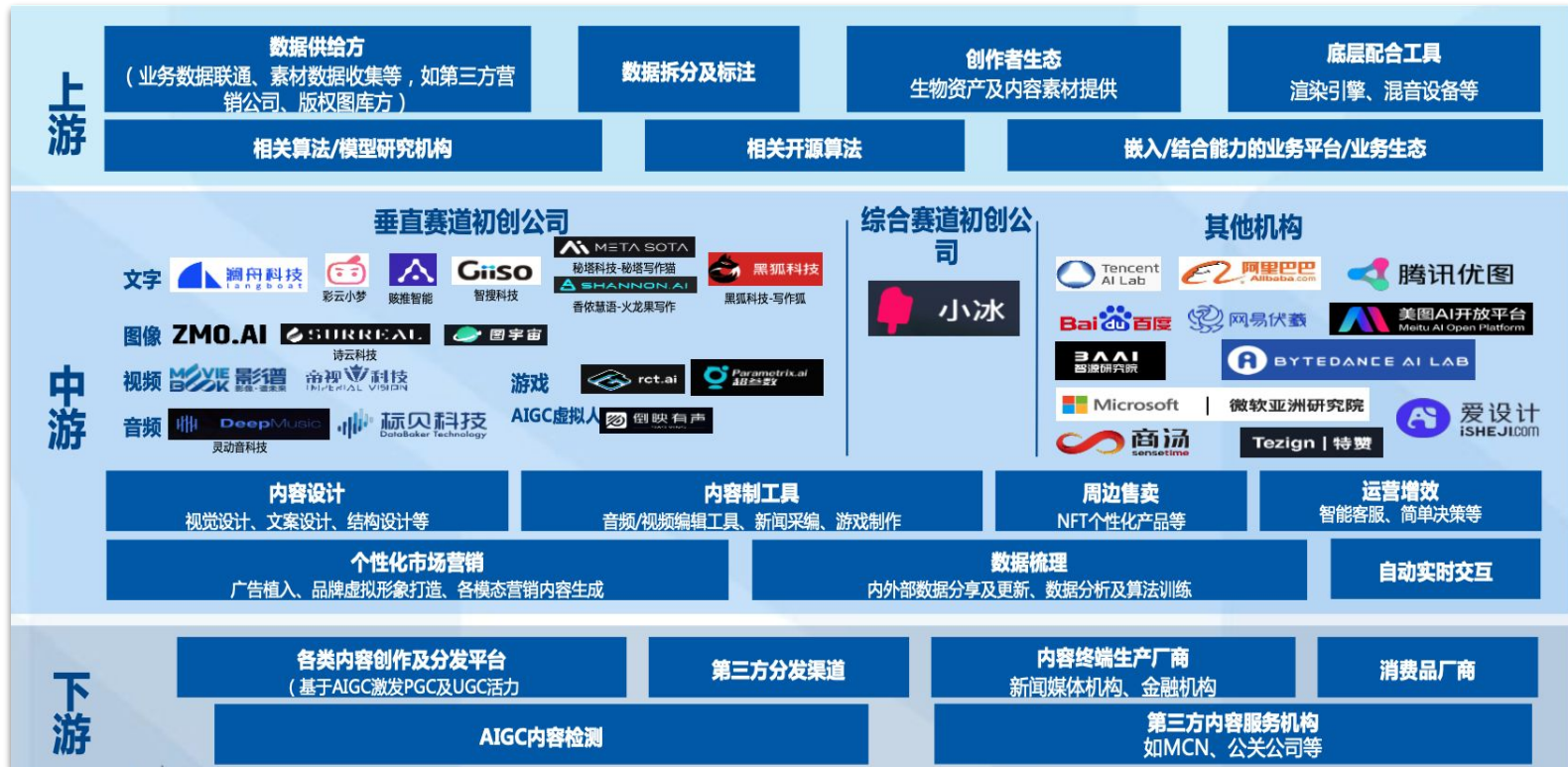
第一次去女友家，女友叫我带五箱茅台，20条软中华，同时带大红包，怎么办？

第一次去女友家，女友叫你带五箱茅台，20条软中华，同时带大红包，这是一种非常正式和亲近的礼物馈赠方式。以下是一些建议：

1. 尊重女友的意愿：虽然这是一种有创意的礼物选择，但在选择礼物之前，最好与女友进行充分的沟通，了解她真正想要的是什么样的礼物。如果女友明确表示不希望收到贵重礼物，那就不要强迫她。
2. 根据实际情况做出选择：如果你经济状况较为宽裕，可以考虑选择更加高档的礼物。但如果你的经济能力有限，建议考虑选择适合自己的礼物。例如，可以选择一些小巧、实用且适合女友家庭使用的礼品。
3. 注意礼物的包装和交付方式：为了让礼物看起来更加体面和尊重，应该选择适当的包装和交付方式。例如，可以选择礼盒或袋子将礼物包裹好，并在送礼时准备一份礼单，写明礼品名称、数量和价值等信息。同时，还应该及时向女友提供礼物，并感谢她对你的慷慨。
4. 感谢女友的关心：女友给你准备这么多礼物，其实也是对你关心和爱护的一种回报。在接受礼物后，可以向女友表达感激之情，并感谢她的关心。

总之，第一次去女友家，选择适当的礼物并保持诚信和礼貌，可以增进彼此间的感情和信任。

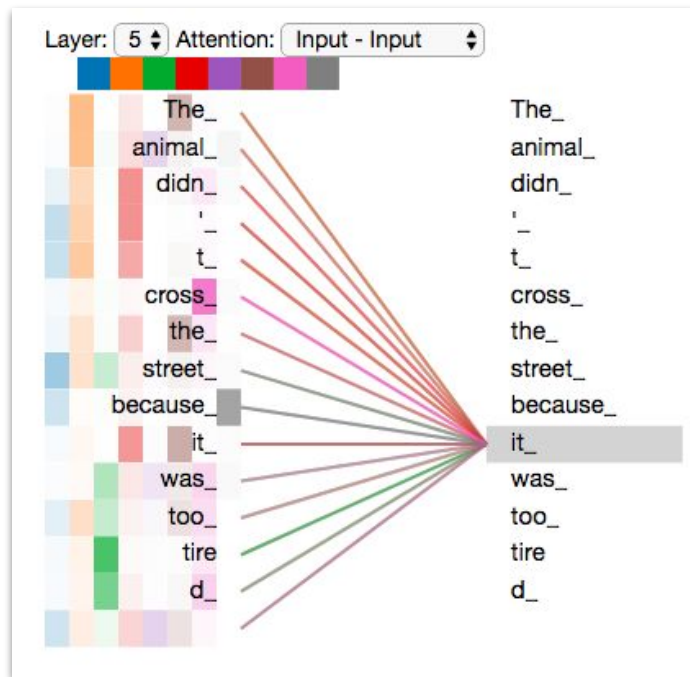
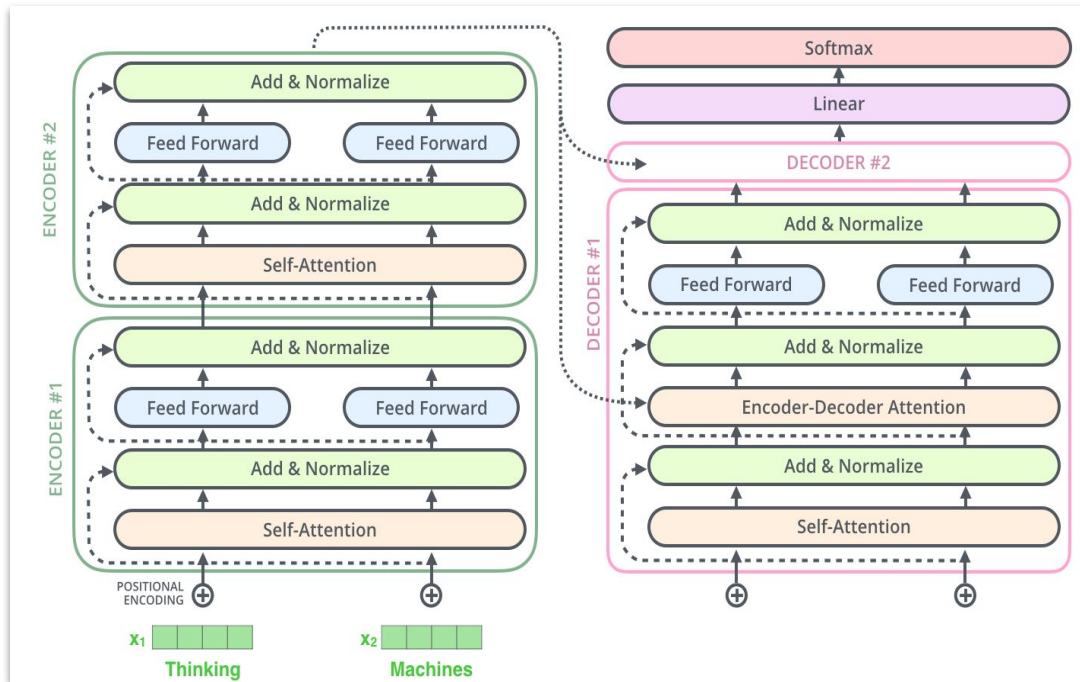
生成式任务简介



语音转图像任务

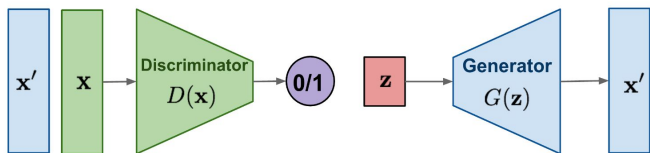
02

生成任务的基础模型——transformer

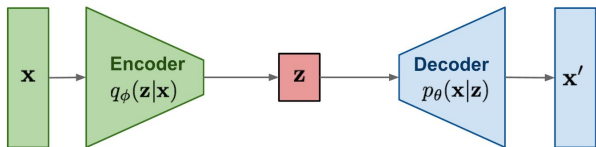


生成任务的基础模型——diffusion model

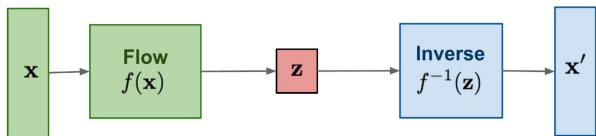
GAN: Adversarial training



VAE: maximize variational lower bound



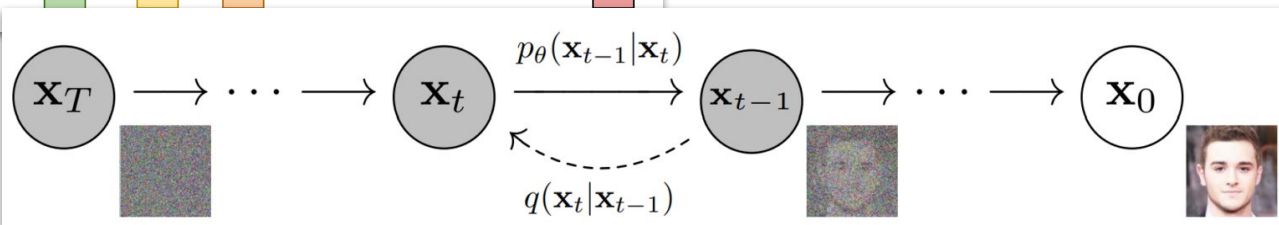
Flow-based models:
Invertible transform of distributions



Diffusion models:
Gradually add Gaussian noise and then reverse



	VAE	Flow	GAN	Diffusion
Pros	Fast Sampling rate. Diverse sample generation	Fast Sampling rate. Diverse sample generation	Fast Sampling rate. High sample generation quality.	High sample generation quality. Diverse sample generation
Cons	Low sample generation quality	Need specialized architecture, low sample generation quality	Unstable training, low sample generation diversity (Mode Collapse)	Low sampling rate



语音转图像第一阶段——语音转文本

• OpenAI whisper模型

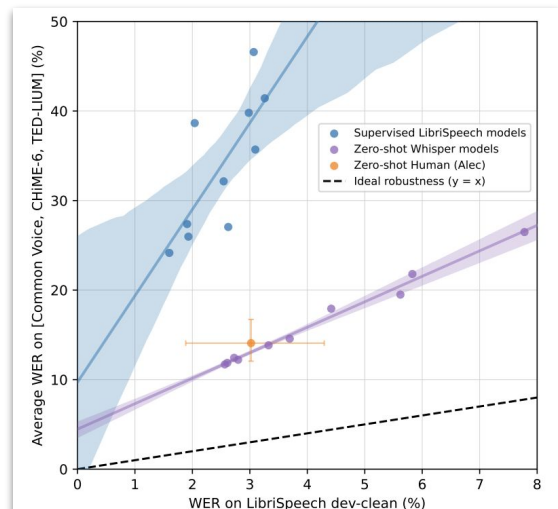
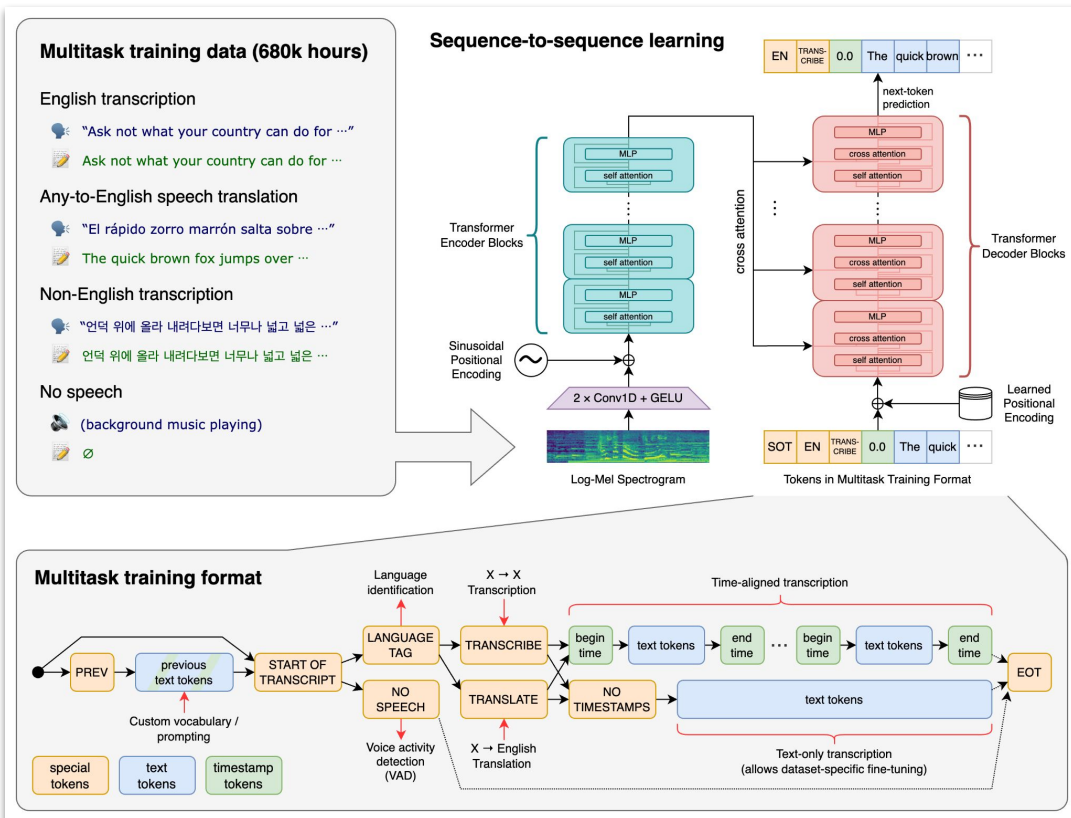
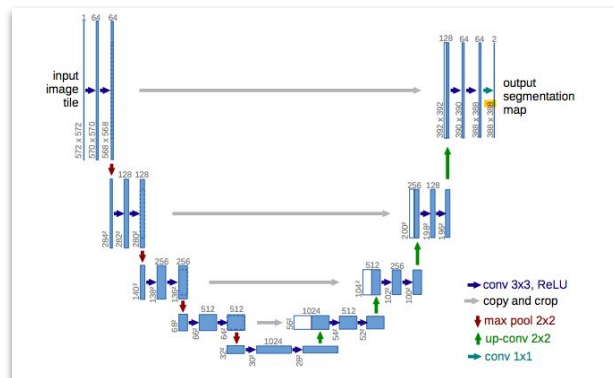
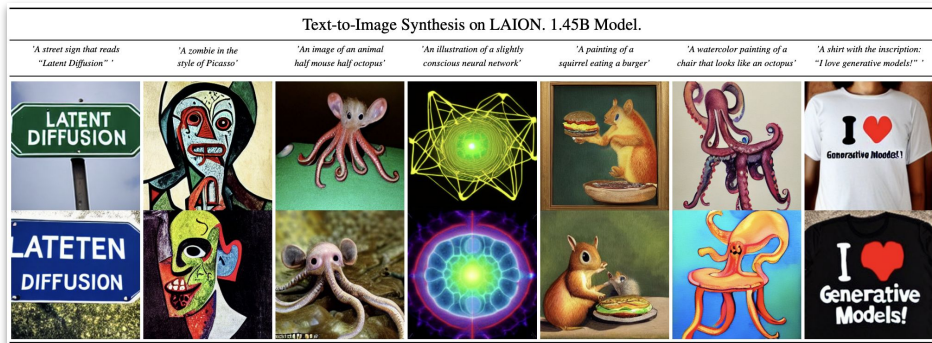
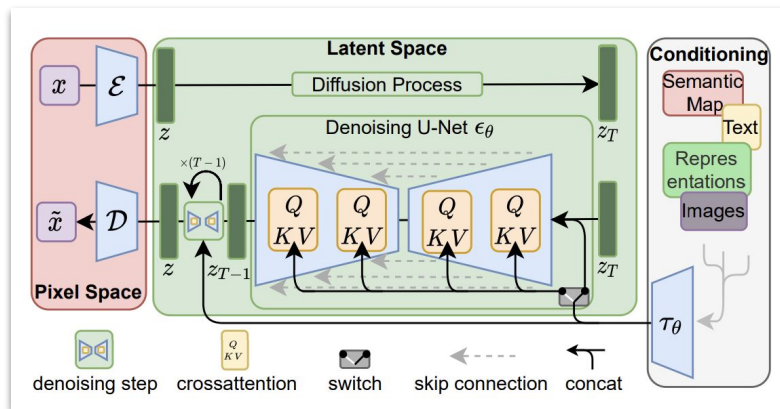


Figure 2. Zero-shot Whisper models close the gap to human robustness. Despite matching or outperforming a human on LibriSpeech dev-clean, supervised LibriSpeech models make roughly twice as many errors as a human on other datasets demonstrating their brittleness and lack of robustness. The estimated robustness frontier of zero-shot Whisper models, however, includes the 95% confidence interval for this particular human.



语音转图像第二阶段——文本转图像

- Stable Diffusion模型
- 图像感知压缩(Perceptual Compression)
 - 潜在扩散模型(Latent Diffusion Models)
 - 条件机制(Conditional Mechanisms)



使用Jina快速搭建工作流

03



如何用Jina搭建 workflow

- 为什么要使用 Jina



Cloud native

- Seamless Docker container integration: sharing, exploring, sandboxing, versioning and dependency control via Executor Hub.
- Full observability via OpenTelemetry, Prometheus and Grafana.
- Fast deployment to Kubernetes and Docker Compose.



Ecosystem

- Improved engineering efficiency thanks to the Jina AI ecosystem, so you can focus on innovating with the data applications you build.



Universal

- Build applications that deliver fresh insights from multiple data types such as text, image, audio, video, 3D mesh, PDF with LF's DocArray.

如何用Jina搭建 workflow

- Jina的几个核心概念



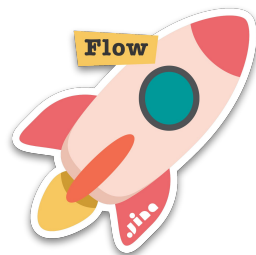
DocArray:

Basic data type to represent a piece of information(text, image, etc)



Executor:

How Jina process Documents.
(Segment, encode, index, etc)



Flow:

How Jina streamlines and distributes Executors

如何用Jina搭建 workflows

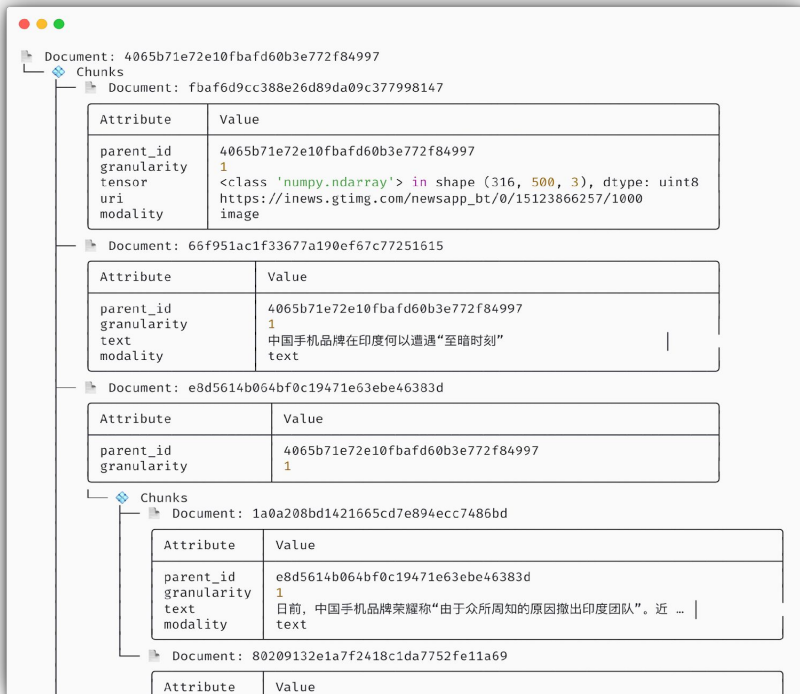
- 使用 DocArray 表示多模态数据

```
- from dataclasses import dataclass
from typing import List
+ from docarray import dataclass, Document
+ from docarray.typing import Image, JSON, Text
```

```
@dataclass
class NewsArticle:
-   content_picture: str
-   title: str
-   paragraphs: List[str]
-   meta: dict
+   content_picture: Image
+   title: Text
+   paragraphs: List[Text]
+   meta: JSON
```

```
article = NewsArticle(
    ...
)
```

```
+ doc = Document(article)
```



如何用Jina搭建 workflows

- 使用DocArray表示多模态数据
- 使用嵌套结构表示不同语义颗粒度的信息
- 支持不同模态的基本操作
- 支持colab和notebook中数据预览
- 支持条件查询
- 支持不同的深度学习框架
- 支持向量表示/预览及不同的向量存储方案

```
from docarray import DocumentArray

# in-memory
da = DocumentArray()

# sqlite
da = DocumentArray(storage='sqlite')

# weaviate
da = DocumentArray(storage='weaviate')

# qdrant
da = DocumentArray(storage='qdrant', config={'n_dim': 128})

# elasticsearch
da = DocumentArray(storage='elasticsearch', config={'n_dim': 128})

# annlite
da = DocumentArray(storage='annlite', config={'n_dim': 128})
```



```
from docarray import Document

d = Document(uri='toy.mp4')
d.load_uri_to_video_tensor()

print(d.tensor.shape)
```

(250, 176, 320, 3)

```
r = da.find({'modality': {'$eq': 'D'}})

pprint(r.to_dict(exclude_none=True)) # just for pretty print
```

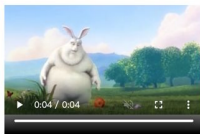
```
[{"id": "92aee5d665d0c4dd34db10d83642aded",
 "modality": "D",
 "tags": {"h": 8.5, "uom": "in", "w": 11.0},
 "text": "paper",
 "weight": 100.0},
 {"id": "1a9d2139b02b1c7842ecda94b347889",
 "modality": "D",
 "tags": {"h": 22.85, "uom": "cm", "w": 30.0},
 "text": "planner",
 "weight": 75.0}]
```

```
In [1]: from docarray import Document

d = Document(uri='docs/_static/hello.wav')
d.display()
```

0:00 / 0:00

```
In [2]: d = Document(uri='docs/_static/60fps.mp4')
d.display()
```



如何用Jina搭建 workflows

- Executor的内部实现

```
@requests
def transcribe(self, docs: DocumentArray, **kwargs):
    """
    Receives a DocumentArray of Documents containing audio data, either in `Document.tensor` or `Document.uri` field.
    It converts it to text using whisper model and inserts the resulting text in `Document.text` field.
    It is also possible to perform translation of other languages into english.
    The `language` parameter can be specified in `Document.tags` as a key-value pair, for instance: 'language: spanish'.
    List of supported languages: https://github.com/openai/whisper/blob/8cf36f354a3629181417435e556a2825b362d53c/whisper/decoding.py#L38
    :param model_name: the model name used to load whisper. Available model names: 'base', 'small', 'medium', 'large'
    """

    for (i, doc_) in enumerate(docs):
        model_output = self.model.transcribe(
            doc_.uri if doc_.tensor is None else doc_.tensor, task='translate',
        )
        doc_.text = model_output['text']
        doc_.tags['segments'] = model_output['segments']
        doc_.tags['language'] = model_output['language']

    return docs
```

```
@requests(on='/')
def generate(self, docs: DocumentArray, parameters: dict, **kwargs):
    """
    Generate images from text using the StableDiffusion model.
    This endpoint expects documents with text prompt stored in `Document.text`, will generate up to
    `num_images` images represented as Document objects. The Document objects contain the image data stored in the
    `Document.uri` field and are added to the query `Document.matches` field.
    The `num_images` parameter can be specified as a key-value pair in `parameters` of the request.
    """

    num_images = parameters.get('num_images', 1)

    for document in docs:
        self.generate_image_from_document(document, num_images)

    return docs

def generate_image_from_document(
    self, document: Document, num_images: int
) -> Document:

    with torch.autocast('cuda'):
        generated_imgs = self.diffusion([document.text] * int(num_images)).images

    for img in generated_imgs:
        _generated_document = Document(
            tags={
                'text': document.text,
                'generator': self.__class__.__name__,
            }
        ).load_pil_image_to_datauri(img)

        document.matches.append(_generated_document)
```

如何用Jina搭建 workflows

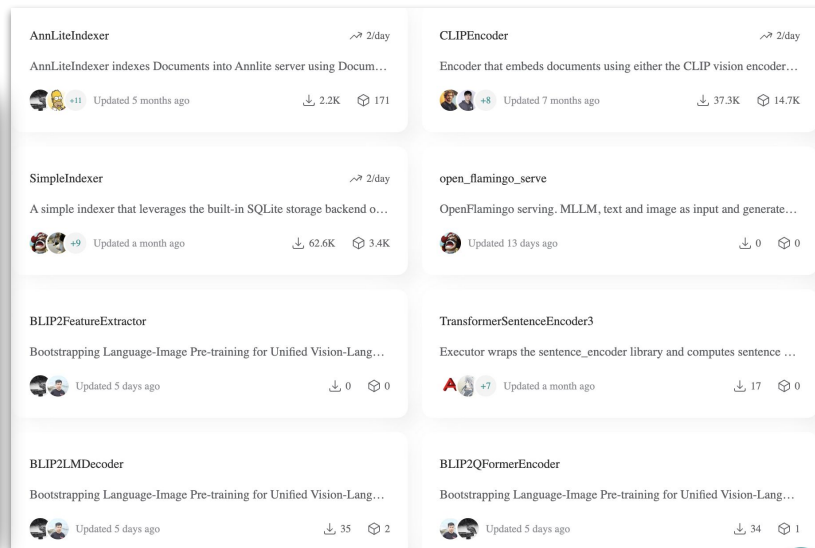
- Flow的构成



- Flow yaml

```

jtype: Flow
executors:
- name: whisper
  uses: jinahub+docker://WhisperExecutor
  uses_with:
    model_name: medium
  gpus: all
  jcloud:
    resources:
      gpu: 1
      memory: 2G
- name: diffuson
  uses: jinahub+docker://StableDiffusionExecutor
  uses_with:
    auth_token: ${ ENV.HF_TOKEN }
    timeout_ready: -1 # slow download speed often leads to timeout
    replicas: 2
  jcloud:
    resources:
      gpu: 1
      memory: 16G
  
```



Executor Name	Description	Updated	Downloads	Stars
AnnLiteIndexer	AnnLiteIndexer indexes Documents into Annlite server using Docum...	Updated 5 months ago	2.2K	171
CLIPEncoder	Encoder that embeds documents using either the CLIP vision encoder...	Updated 7 months ago	37.3K	14.7K
SimpleIndexer	A simple indexer that leverages the built-in SQLite storage backend o...	Updated a month ago	62.6K	3.4K
open_flamingo_serve	OpenFlamingo serving. MLLM, text and image as input and generate...	Updated 13 days ago	0	0
BLIP2FeatureExtractor	Bootstrapping Language-Image Pre-training for Unified Vision-Lang...	Updated 5 days ago	0	0
TransformerSentenceEncoder3	Executor wraps the sentence_encoder library and computes sentence ...	Updated a month ago	17	0
BLIP2LMDDecoder	Bootstrapping Language-Image Pre-training for Unified Vision-Lang...	Updated 5 days ago	35	2
BLIP2QFormerEncoder	Bootstrapping Language-Image Pre-training for Unified Vision-Lang...	Updated 5 days ago	34	1

如何用 Jina 搭建 workflow

- 启动 Flow 并发送请求

```
import os
from jina import Flow
from executors.whisper import WhisperExecutor
from executors.stablediffusion import StableDiffusionExecutor

hf_token = os.getenv('HF_TOKEN')

f = (
    Flow(port=54322)
    .add(uses=WhisperExecutor, timeout_ready=-1, uses_with={'model_name': 'large'})
    .add(uses=StableDiffusionExecutor, uses_with={'auth_token': hf_token})
)

if __name__ == '__main__':
    with f:
        f.block()
```

```
from jina import Client
from docarray import Document
client = Client(host='localhost:54322')
docs = client.post('/', inputs=[Document(uri='audio.wav') for _ in range(1)])
for img in docs[0].matches:
    img.load_uri_to_image_tensor()

docs[0].matches.plot_image_sprites()
```

Demo

04



Berlin · San Jose · Beijing · Shenzhen

大家:問題那麼多有沒有禮物啊



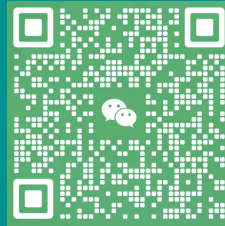
只有兩道題



而且有周邊



領周邊指路 



扫一扫上面的二维码图案，加我为朋友。