多模态技术在学术界的研究现状以及在工业界的落地应用

付杰

jem.fu@jina.ai

AI Engineer@Jina AI

About Jina AI

Jina AI 成立于 2020 年 2 月，是一家业内领先的开源企业，致力于通过人工智能和深度学习技术，打造下一代云原生神经搜索框架，帮助开发者和企业简化非结构化数据的搜索难题。

Jina AI 总部位于德国柏林，在北京、深圳、巴塞罗那等地均设有办公室，海外员工人数比例超过三分之二。

2021 年，Jina AI 完成了由 Canaan Partners 领投的 3000 万美元 A 轮融资，截至目前融资总额达 3900 万美元。

Canaan GGVCAPITAL SAP.iO YUNQI PARTNERS MANGO CAPITAL

DocArray

将非结构化数据，统一成同一种数据结构

Jina

适用于所有数据类型的云原生神经搜索框架

Finetuner

微调深度神经网络，优化神经搜索任务中的向量

Hub

分享和发现神经搜索应用的可复用组件

CLIP-as-service

基于 CLIP 的图像和文本跨模态编码服务

JCloud

神经搜索系统的云端部署及管理平台

NOW

图像搜索的无代码解决方案

Creator of Neural Search

Contributor to Open Source

1 多模态数据vs单模态数据

2 学术界如何处理多模态数据

3 多模态技术在工业界的应用

4 **Jina**在多模态场景下的应用

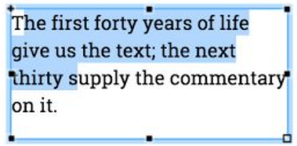
多模态数据VS单模态数据

- **什么是单模态 / 多模态数据？**

我们生活的世界是一个多模态的世界，包含丰富的文本、视觉、听觉、嗅觉信息，

通常我们所接触的物体都不只有一个模态，例如短视频、电商物品等





- **多模态数据处理起来有哪些难点？**

1. **表征**：利用多模态的互补性和冗余性的特质来表示和总结多模态数据
2. 翻译：从一种模式转换为另一种模式
3. **对齐**：确定来自两种或两种以上不同模式的(分)要素之间的直接关系
4. **融合**：将来自两个或多个模式的信息连接起来
5. 协同学习：从一种模式的知识学习可以帮助一个计算模型训练另一个不同的模态

多模态数据VS单模态数据

- **为什么要研究多模态数据？**

图表3：中国短视频行业产业链全景图

资料来源：前瞻产业研究院整理

@前瞻经济学人APP

哪些行业涉及到多模态数据？

1. 短视频数据：
   - 如何理解用户上传的视频
   - 如何实现内容的精准/多样性分发/推荐
2. 电商搜索：
   - 如何更好的理解商品
   - 针对用户搜索历史如何更好地推荐商品

1 多模态数据vs单模态数据

2 学术界如何处理多模态数据

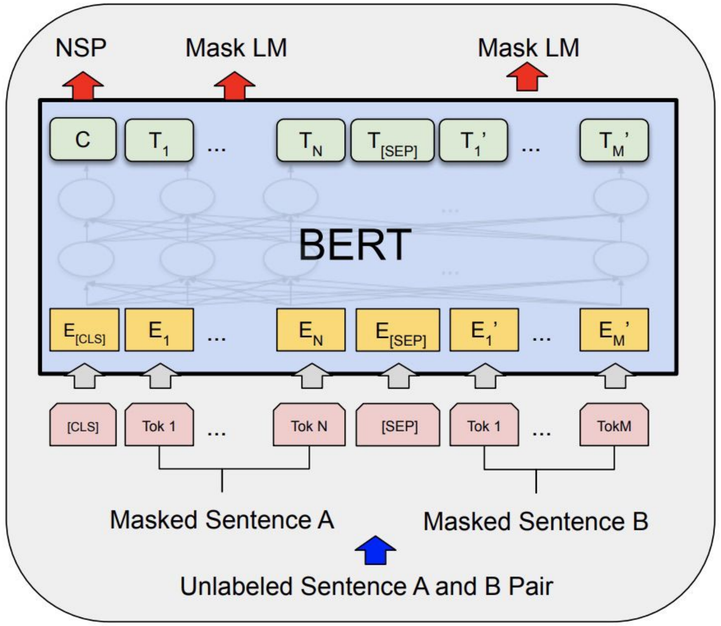
- 多模态表征
- 多模态预训练

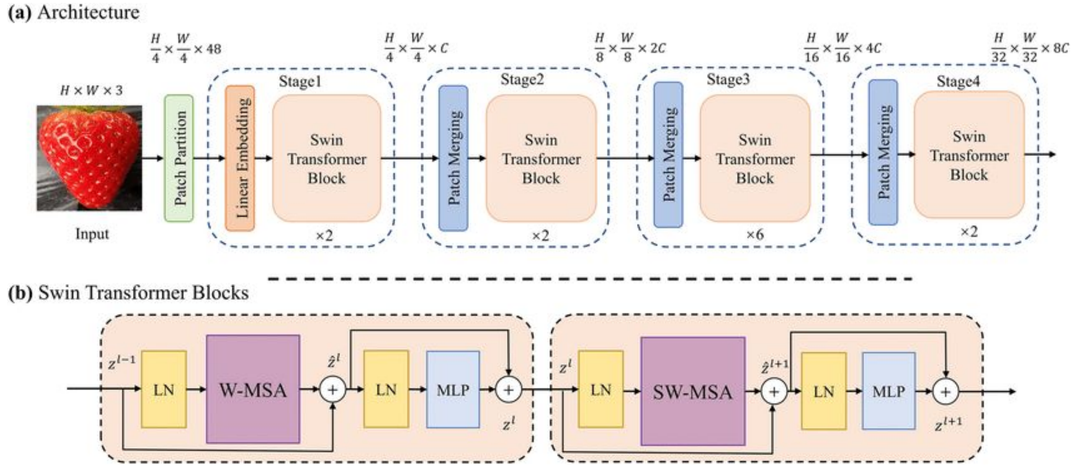
3 多模态技术在工业界的应用

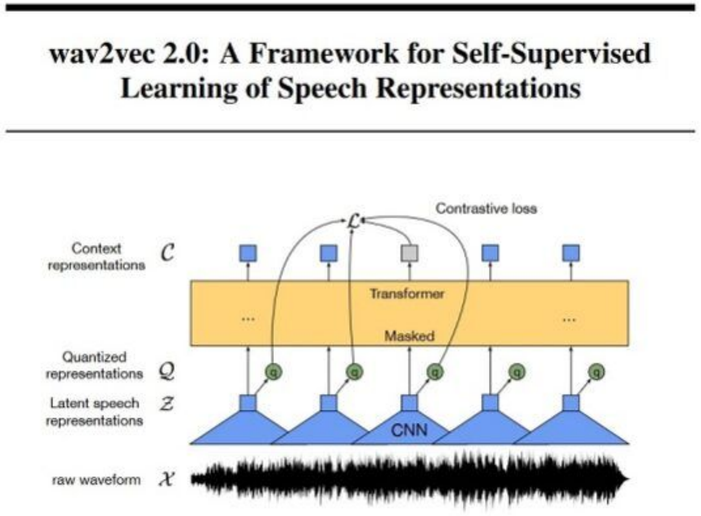
4 Jina在多模态场景下的应用

多模态表征

- **单模态表征**







BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding
Swin Transformer: Hierarchical Vision Transformer using Shifted Windows
wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations

多模态表征

- **多模态的特征对齐：引入传统统计学习方法进行模态对齐**

Canonical Correlation Analysis：将两组数据降维到一维后，其相关系数最大

$$\rho(X,Y)=\frac{cov(X,Y)}{\sqrt{D(X)}\sqrt{D(Y)}}$$

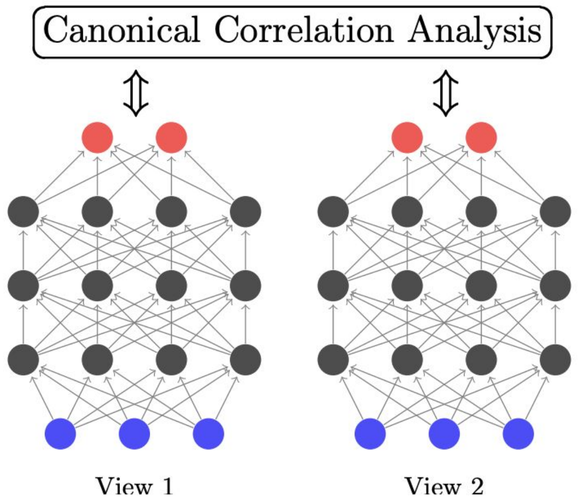
$$\underbrace{arg\,max}_{a,b}\frac{a^T S_{XY} b}{\sqrt{a^T S_{XX} a}\sqrt{b^T S_{YY} b}}$$

$$\underbrace{arg\,max}_{a,b}\ a^T S_{XY} b$$
$$s.t.\ a^T S_{XX} a = 1,\ b^T S_{YY} b = 1$$

$$X' = a^T X, Y' = b^T Y$$

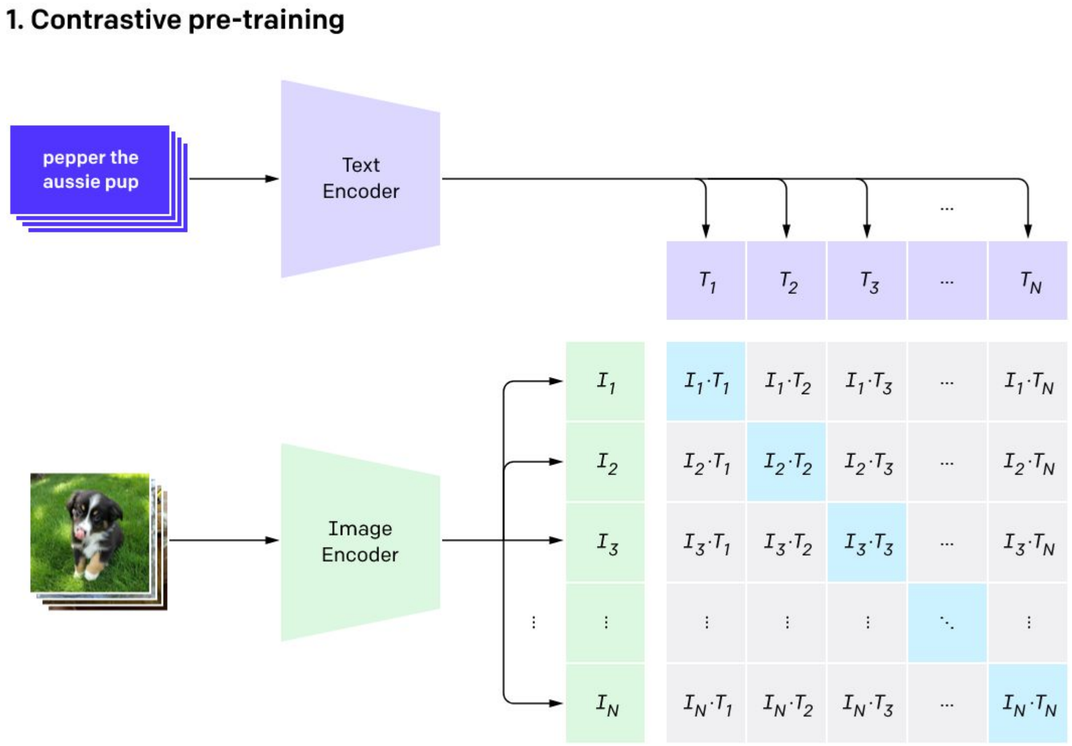
Deep Canonical Correlation Analysis：在深度网络后加入CCA，

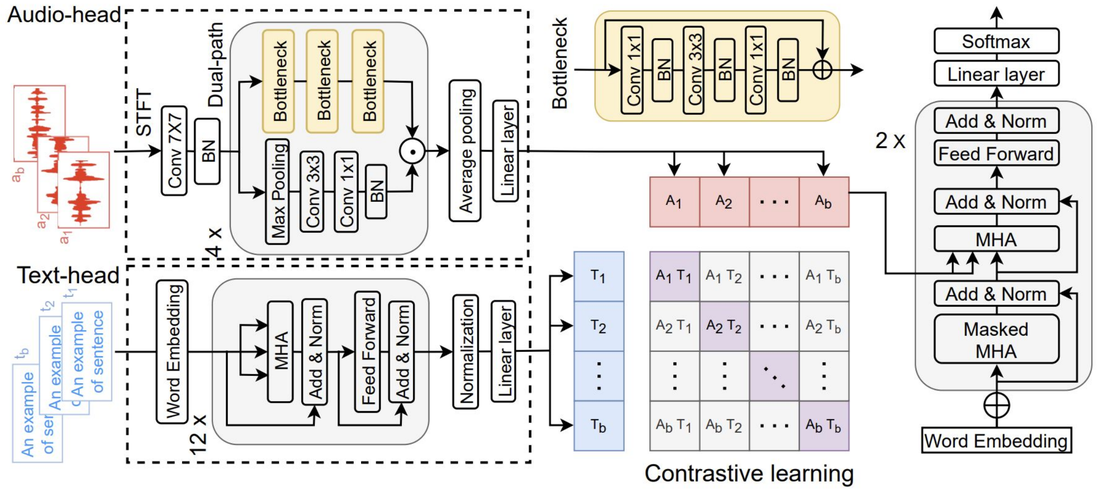
使得网络参数能够自动对相同类别但不同view的数据学习到对齐特征



多模态表征

- **多模态的特征对齐：基于对比学习进行模态对齐**



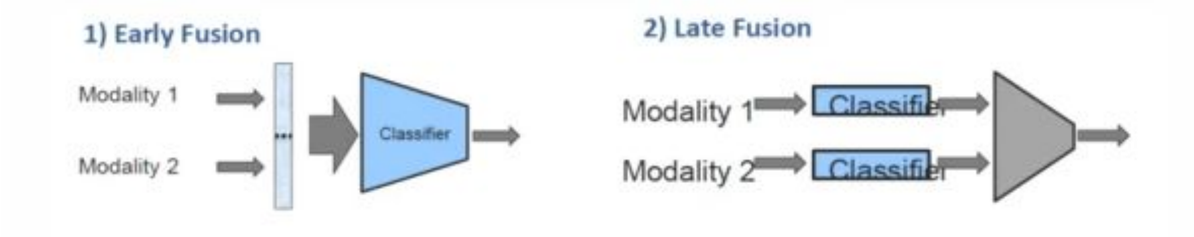


Learning Transferable Visual Models From Natural Language Supervision

Interactive Audio-text Representation for Automated Audio Captioning with Contrastive Learning

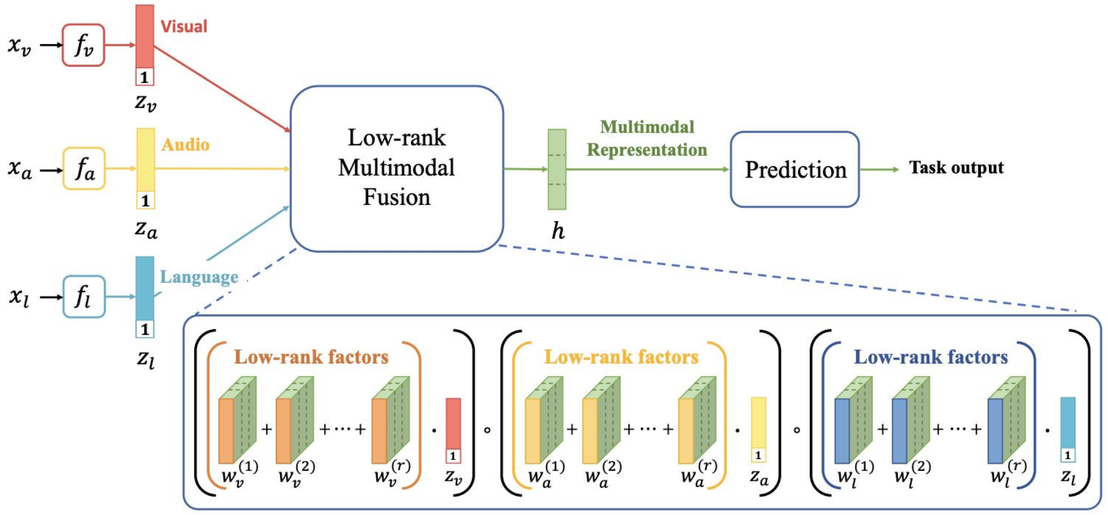
多模态表征

- **多模态的特征融合：基于矩阵/注意力的方法**



基于矩阵方法

基于注意力的方法



Efficient Low-rank Multimodal Fusion with Modality-Specific Factors

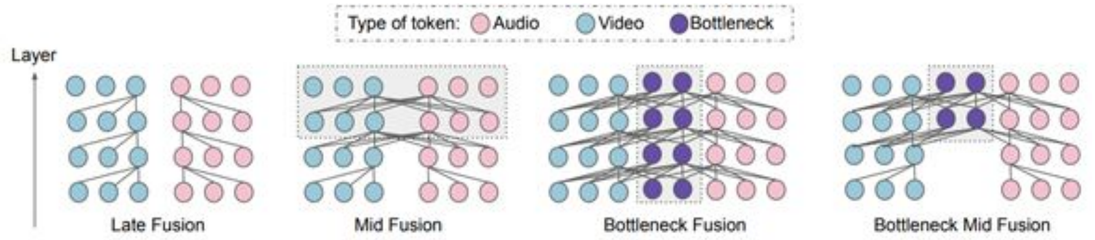


Figure 1: **Cross-modal Fusion**. Unlike late fusion (left), where no cross-modal information is exchanged in the model until after the classifier, we investigate two pathways for the exchange of cross-modal information. The first is via standard pairwise self attention across all hidden units in a layer, but applied only to later layers in the model – mid fusion (middle, left). We also propose the use of 'fusion bottlenecks' (middle, right) that restrict attention flow within a layer through tight latent units. Both forms of restriction can be applied in conjunction (Bottleneck Mid Fusion) for optimal performance (right). We show $B = 2$ bottleneck units and 3 hidden units per modality. Grey boxes indicate tokens that receive attention flow from both audio and video tokens.

Attention Bottlenecks for Multimodal Fusion

1 多模态数据vs单模态数据

2 学术界如何处理多模态数据

- 多模态表征
- 多模态预训练

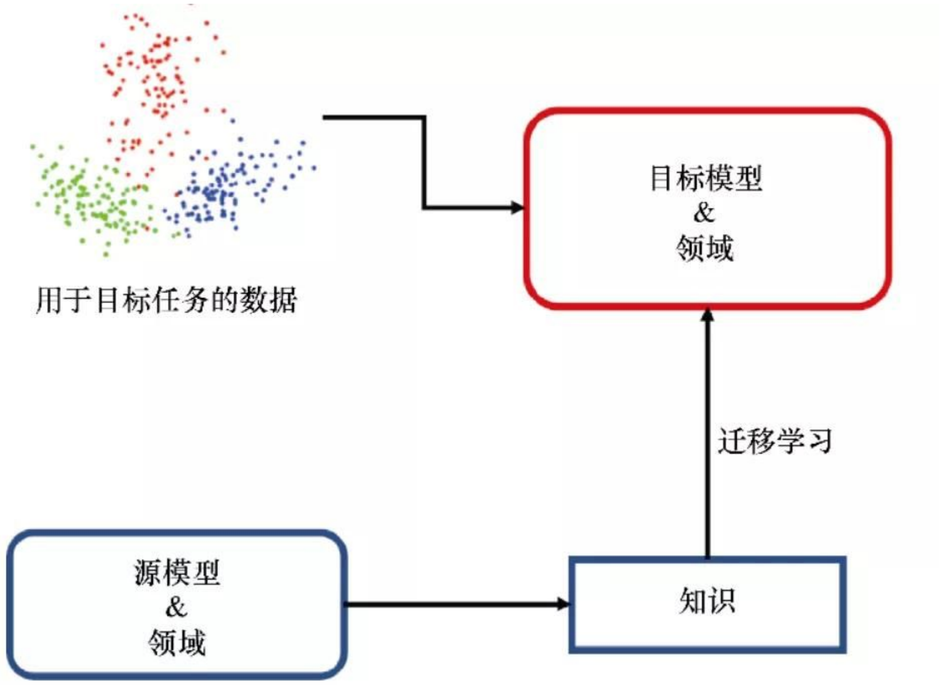
3 多模态技术在工业界的应用

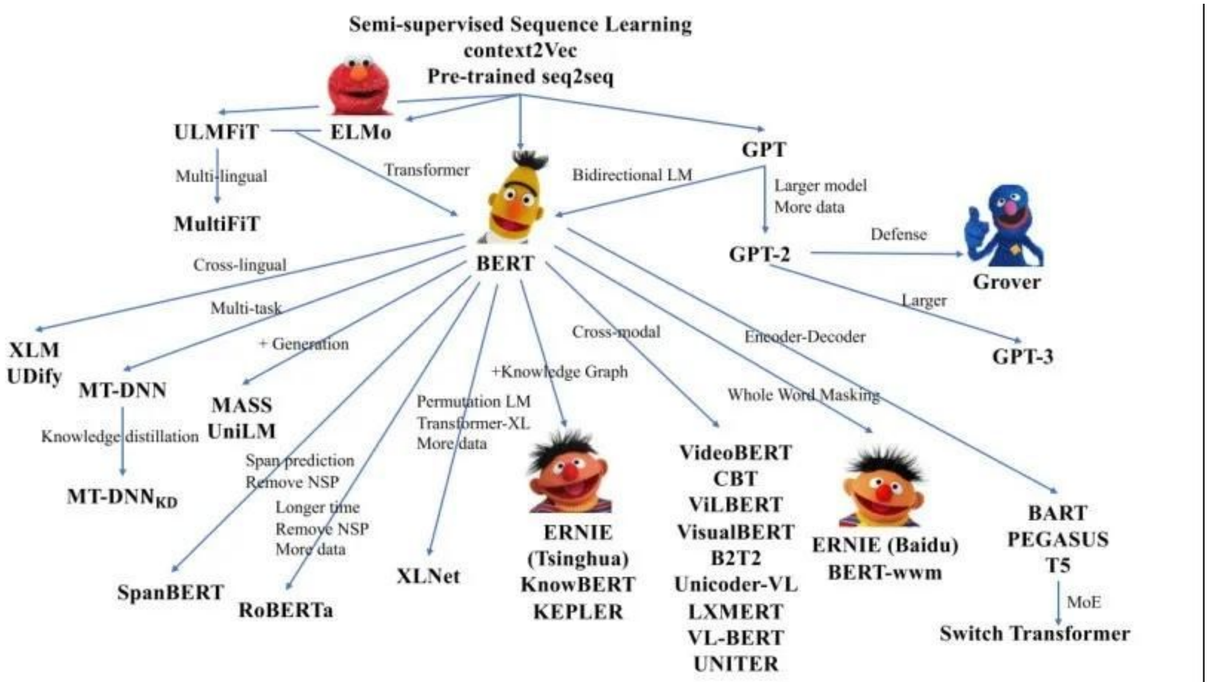
4 Jina在多模态场景下的应用

多模态预训练

- **什么是预训练**

把人类的语言知识，先学了一个东西，然后再代入到某个具体任务，就顺手了





多模态预训练

- **常用预训练任务**

Masked Language Modeling (MLM)

$$\mathcal{L}_{\mathrm{MLM}} = -\mathbb{E}_{(\mathcal{W},\mathcal{V})\in\mathcal{D}} \log P_\theta \left(w_m | w_{\setminus m}, \mathcal{V}\right), \quad (1)$$

where $w_m, w_{\setminus m}$ represent the masked tokens and unmasked tokens, respectively, and $(\mathcal{W},\mathcal{V}) \in \mathcal{D}$ represents a text $\mathcal{W}$ and an image $\mathcal{V}$ sampled from dataset $\mathcal{D}$.

Masked Region Prediction (MRP)

- **Masked Region Classification (MRC)**
- **Masked Region Feature Regression (MRFR**

$$\mathcal{L}_{\mathrm{MRC}} = \mathbb{E}_{(\mathcal{W},\mathcal{V})\in\mathcal{D}} \sum_{i=1}^{m} \mathrm{CE}\left(\mathrm{softmax}(\mathrm{FC}(h_{v_i}), c(v_i))\right), \quad (2)$$

where m is the length of image features; $c\left(v_i\right)$ represents the true label of the masked region, such as the object detection output or the (pre-defined) visual tokens.

$$\mathcal{L}_{\mathrm{MRFR}} = \mathbb{E}_{(\mathcal{W},\mathcal{V})\in\mathcal{D}} \sum_{i=1}^{m} \|\mathrm{FC}(h_{v_i}) - \hat{E}_v(v_i)\|^2. \quad (3)$$

Image-Text Matching (ITM)

$$\begin{aligned}\mathcal{L}_{\mathrm{VLM}} = &- E_{(\mathcal{W},\mathcal{V})\in\mathcal{D}} \left[y \log s_\theta(\mathcal{W},\mathcal{V})\right. \\ &\left.+(1-y)\log\left(1-s_\theta(\mathcal{W},\mathcal{V})\right)\right].\end{aligned} \quad (4)$$

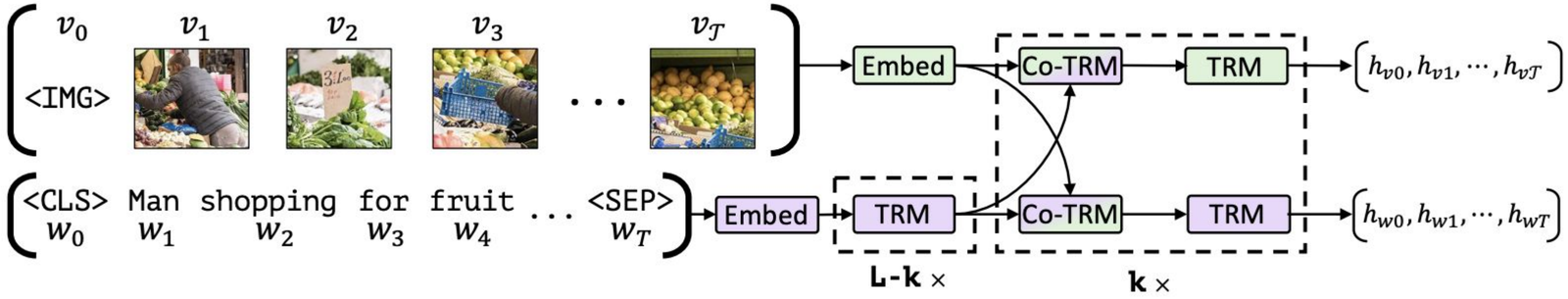
Cross-Modal Contrastive Learning (CMCL)

$$\mathcal{L}_{\mathrm{i2t}} = -\frac{1}{n}\sum_{i=1}^{N} \log \frac{\exp\left(x_i^\top y_i/\sigma\right)}{\sum_{j=1}^{N}\exp\left(x_i^\top y_j/\sigma\right)}, \quad (5)$$

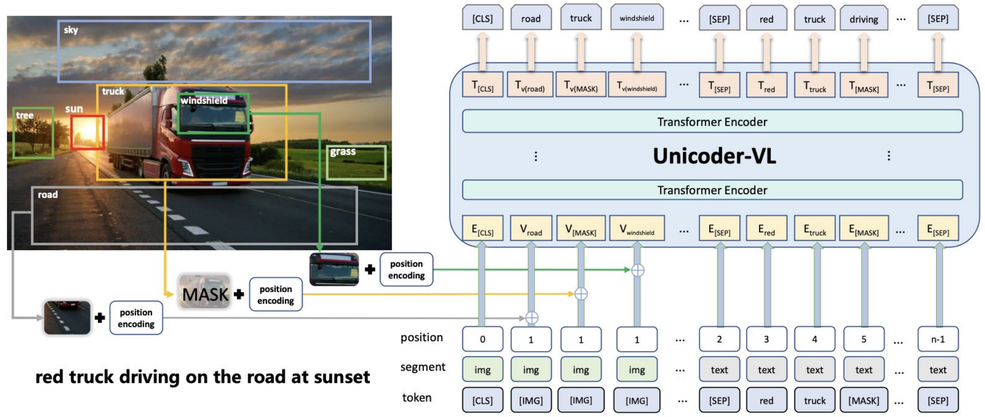
多模态预训练

- **经典的双塔/单流模型**

Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks



A Universal Encoder for Vision and Language by Cross-modal Pre-training



双塔模型无需计算cross-attention，速度更快，能满足线上需求，但性能相对较单流模型更低

多模态预训练

- **基于对比学习的预训练模型**

CommerceMM: Large-Scale Commerce MultiModal Representation Learning with Omni Retrieval

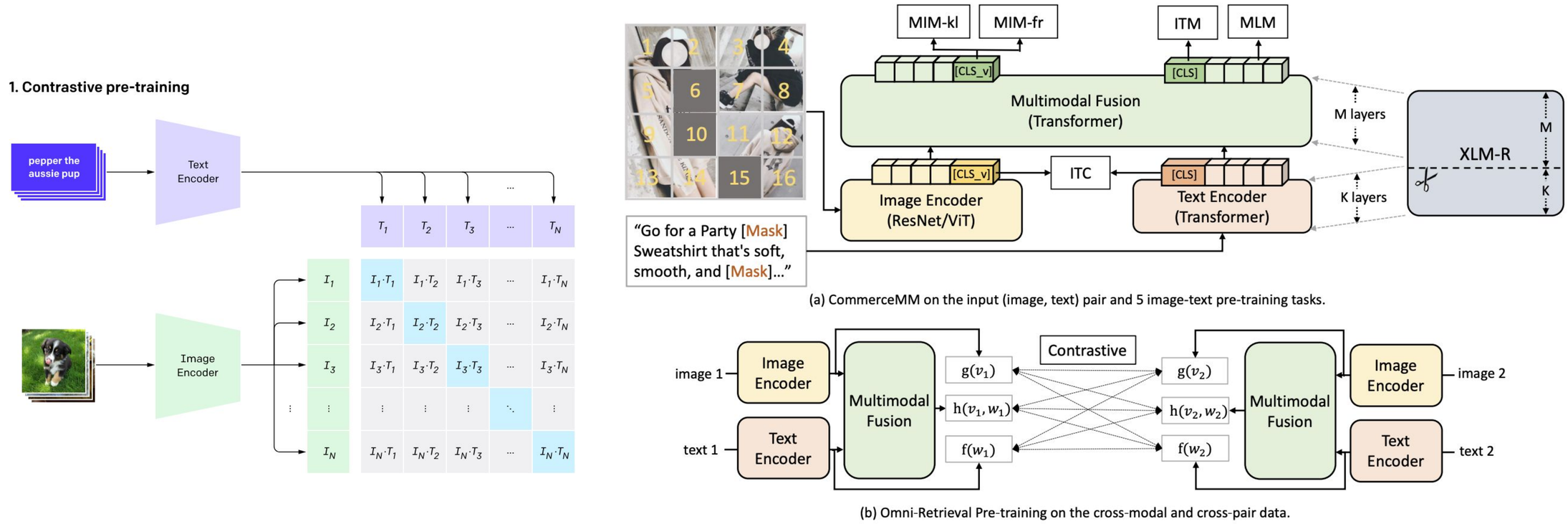
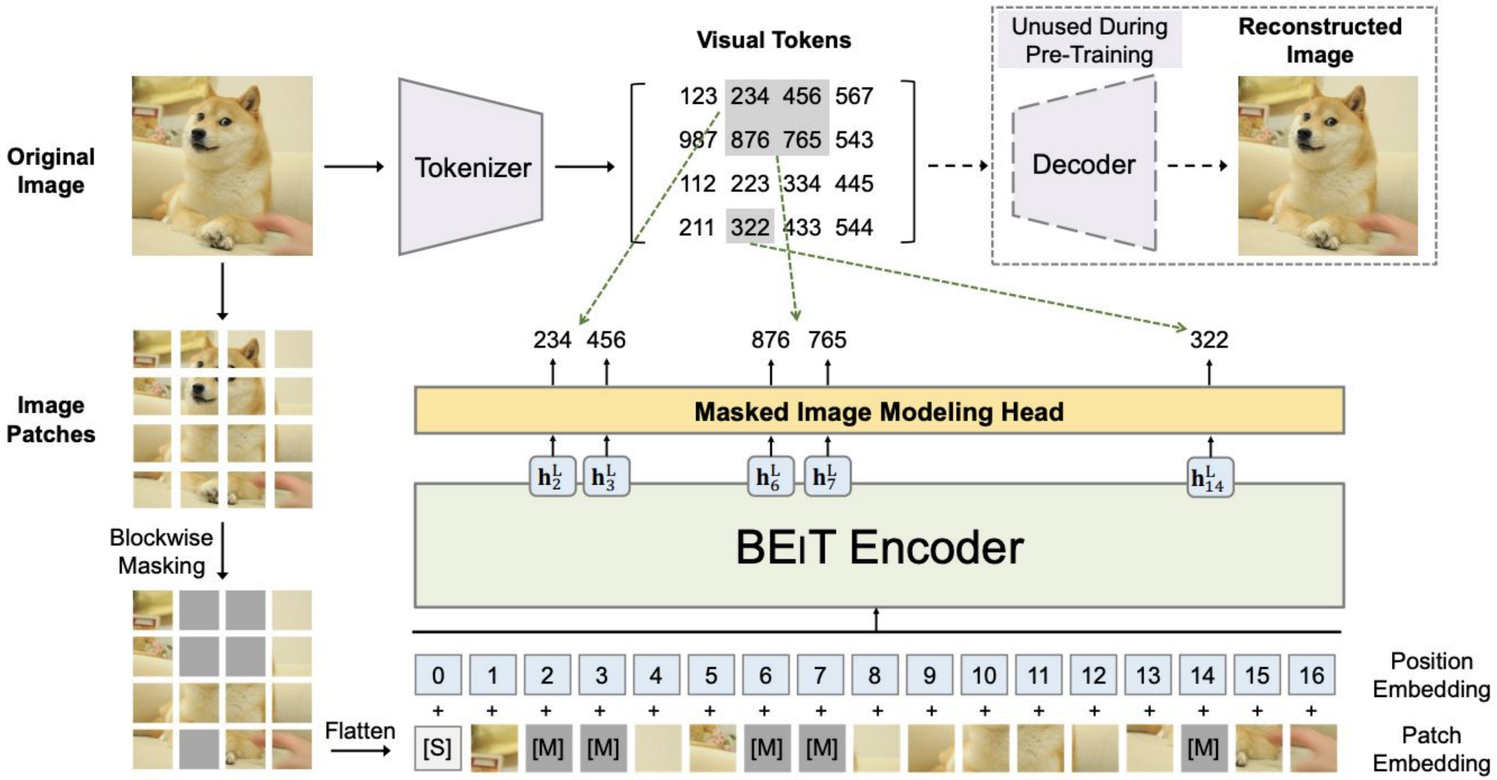


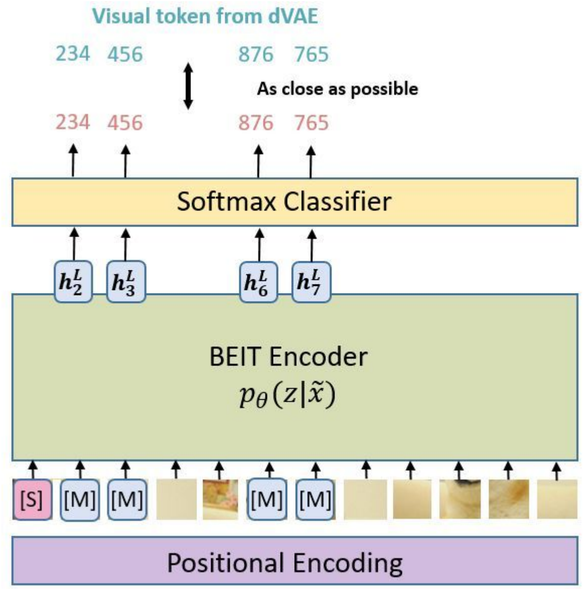
Figure 2: CommerceMM Model Architecture with the image-text pre-training and omni-retrieval tasks.

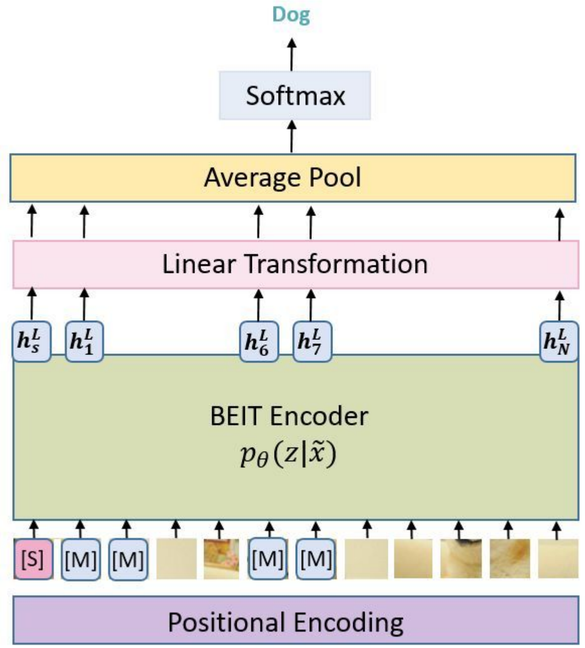
多模态预训练

- **基于predict masked patch的预训练模型**

BEIT: BERT Pre-Training of Image Transformers







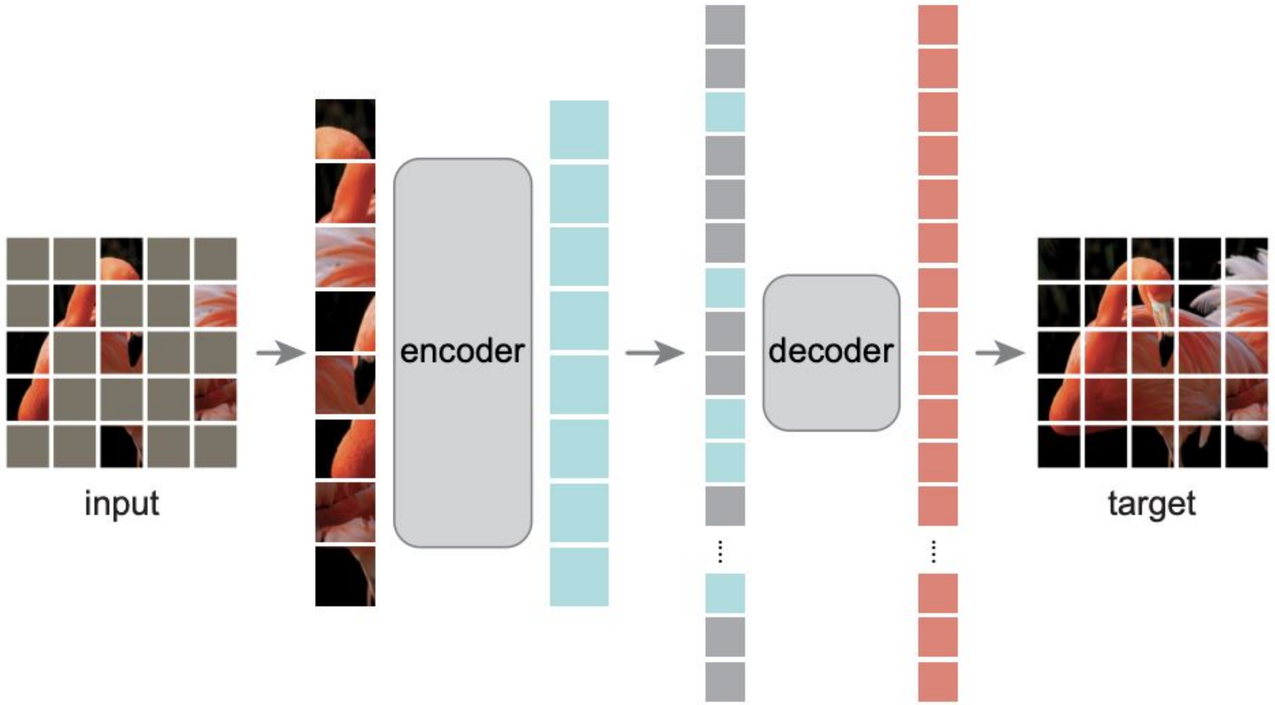
多模态预训练

- **基于predict masked patch的预训练模型**

Masked Autoencoders Are Scalable Vision Learners

掩码自编码在视觉和文本上有什么不同？

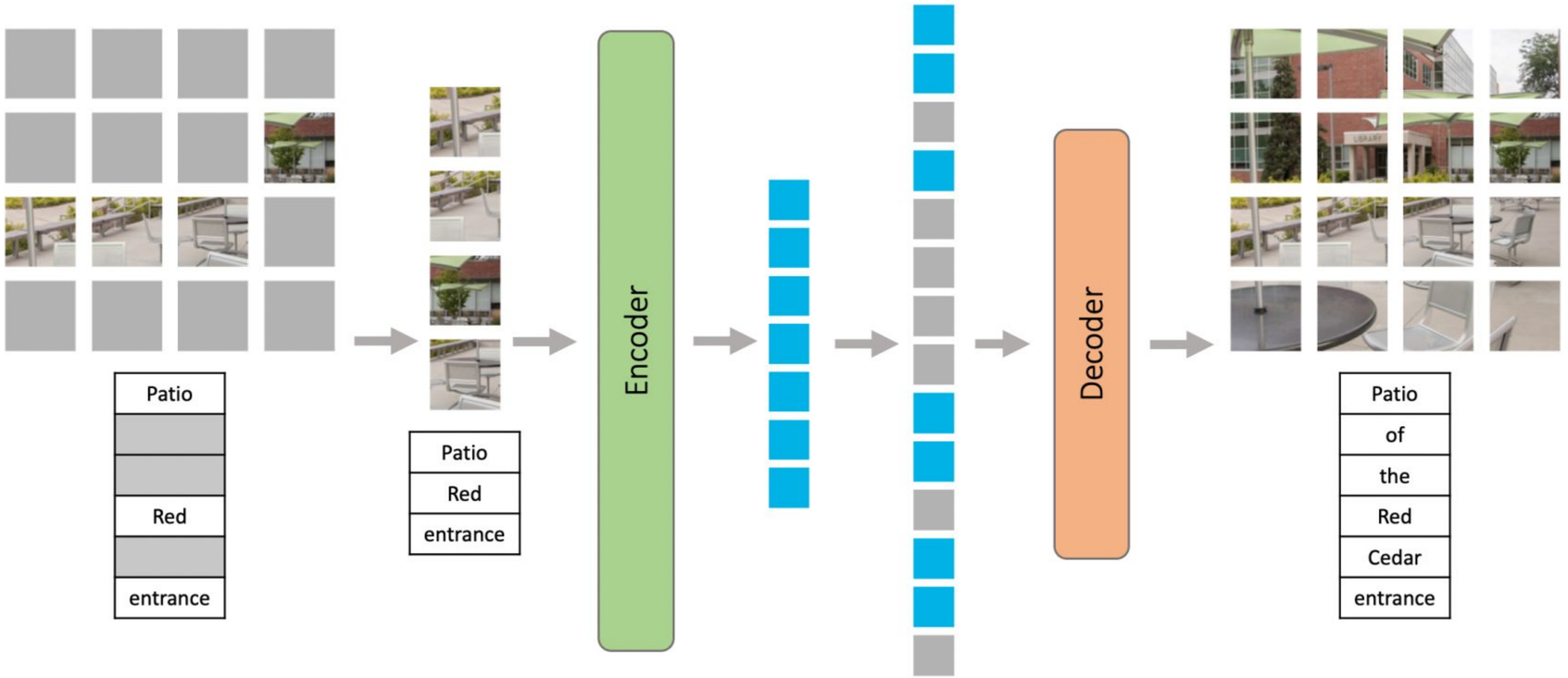
1. 模型结构有差异：transformer vs. conv
2. 信息密度差异较大：15% vs. 75%
3. 解码目标不一致：token vs. pixel



多模态预训练

- **基于predict masked patch的预训练模型**

Multimodal Masked Autoencoders Learn Transferable Representations



1 多模态数据vs单模态数据

2 学术界如何处理多模态数据

3 多模态技术在工业界的应用

4 Jina在多模态场景下的应用

多模态技术在工业界的应用

- **爱奇艺/优酷——视频搜索，视频内容理解**

基于视频内容召回

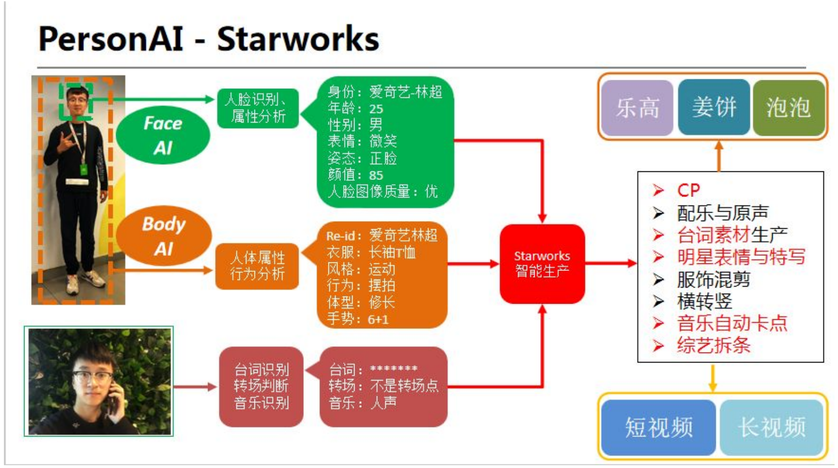
基于标题以出现召回不足

基于视频图像以下信息召回

1、类目标签
2、事件/场景标签
3、物体/人物标签
4、字幕/OCR/ASR形成NLP标签

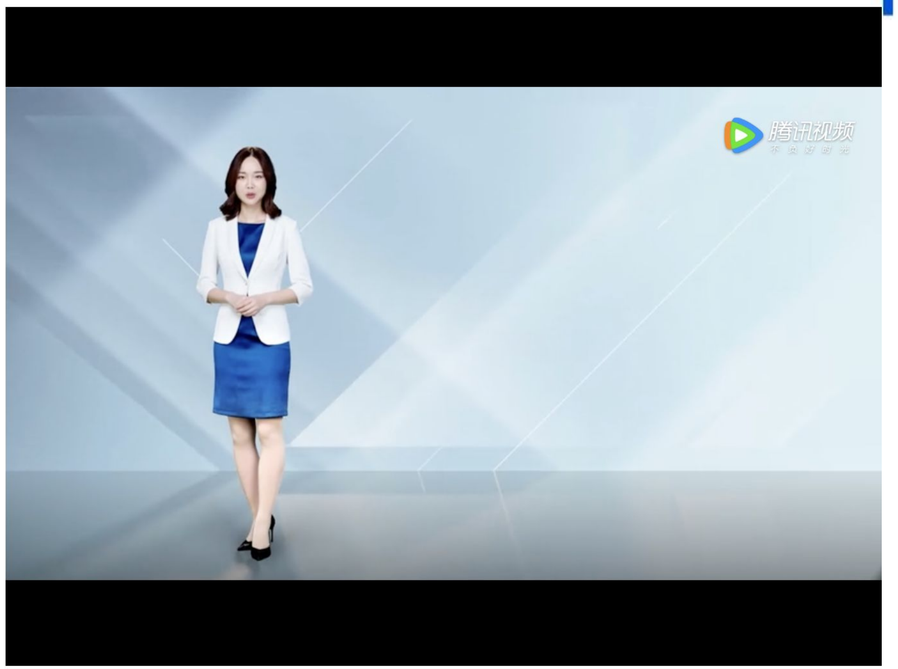
- 多模态召回辅助文本召回
- 只看TA
- 明星视频自动化生成

…

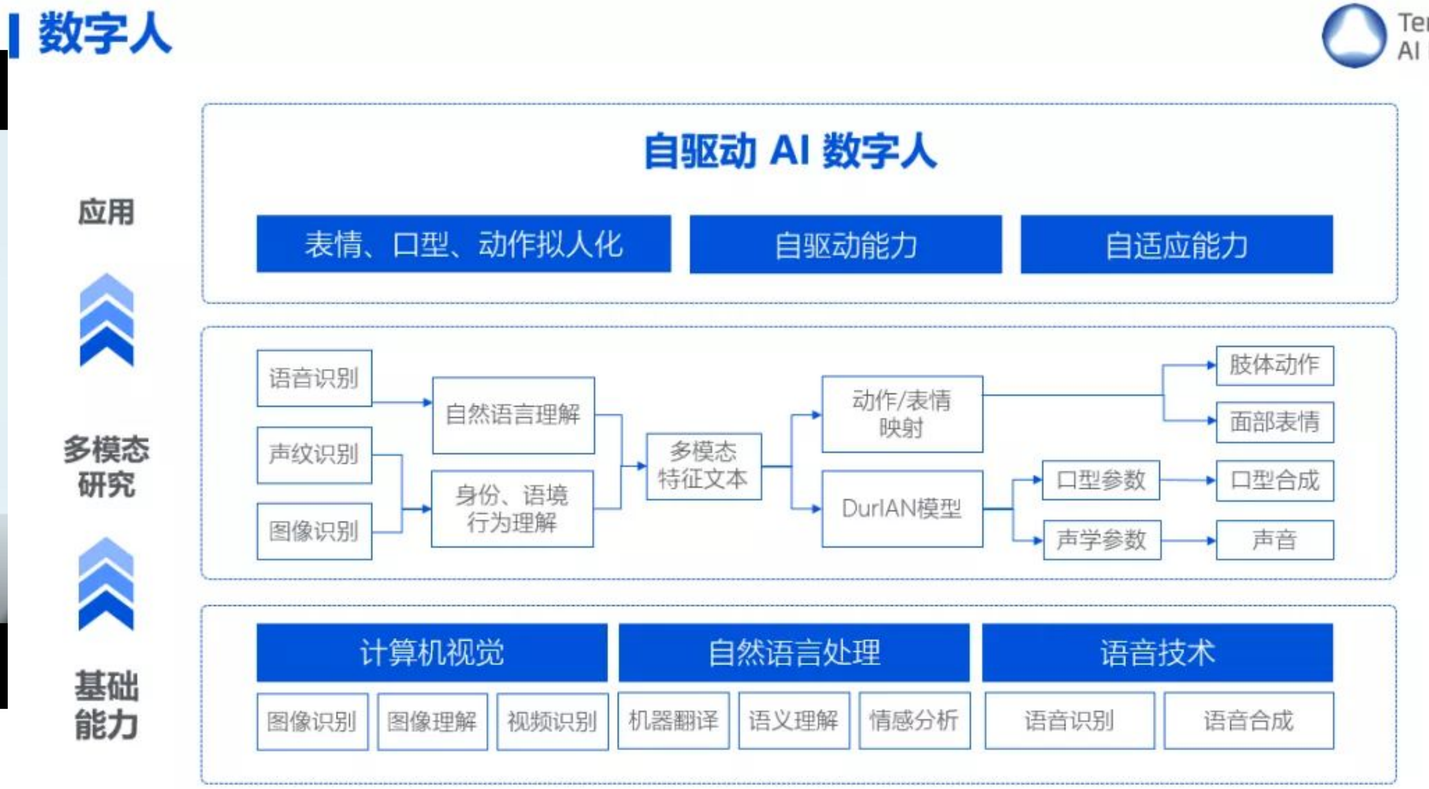


多模态技术在工业界的应用

- **腾讯AILab——数字人**



腾讯AI Lab神经网络渲染数字人



1 多模态数据vs单模态数据

2 学术界如何处理多模态数据

3 多模态技术在工业界的应用

4 **Jina在多模态场景下的应用**

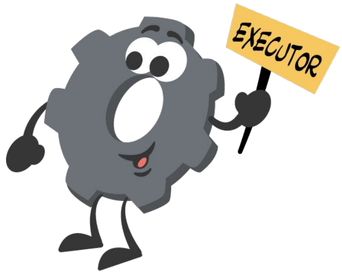
- 搭建视频搜索**demo**
- **DalleFlow**

用Jina快速搭建视频搜索demo

- **Jina的几个核心概念**

DocArray:

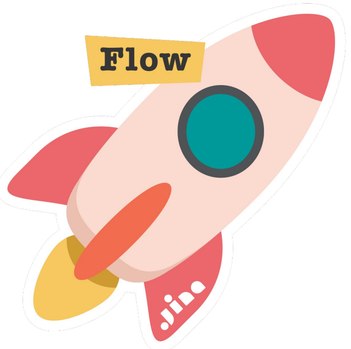
Basic data type to represent a piece of information(text, image, etc)



Executor:

How Jina process Documents.

(Segment, encode, index, etc)



Flow:

How Jina streamlines and distributes Executors

用Jina快速搭建视频搜索demo

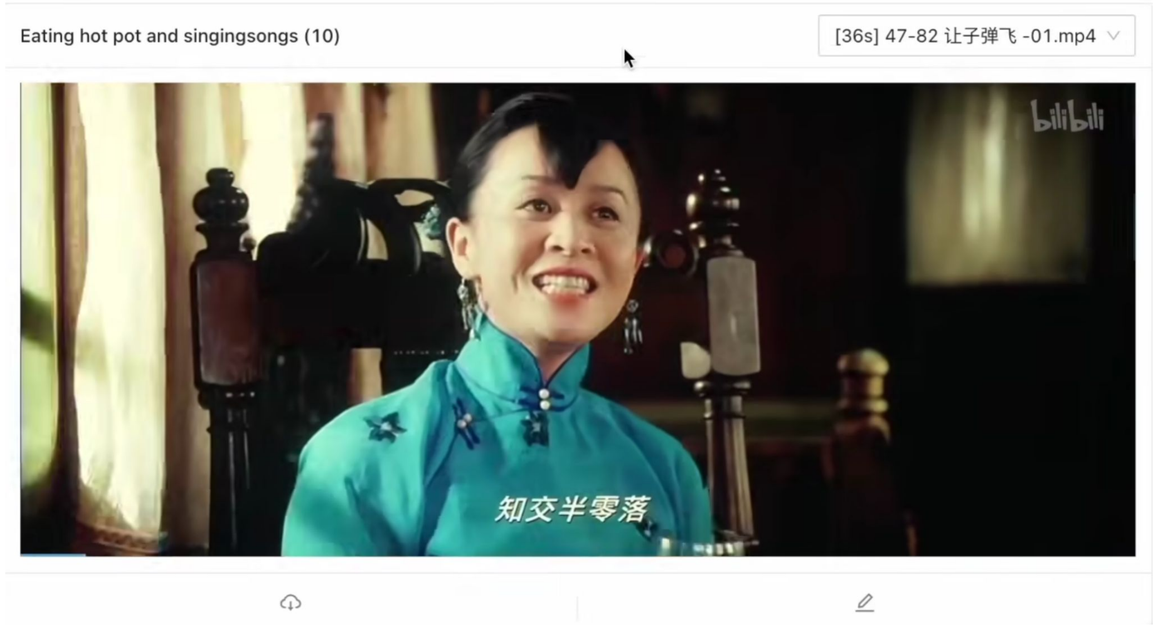
- 场景：up主想要快速在成千上万的视频素材库里搜索到和自己脚本内容相匹配的片段

- Bilibili：用 AI 搜索梦华录刘亦菲口红同款
- 线上测试：https://lipstick-db.senses.chat
- Github：lipstick-db

用Jina快速搭建视频搜索demo

- **场景：up主想要快速在成千上万的视频素材库里搜索到和自己脚本内容相匹配的片段**

Query: eating hot pot and singing songs



Query: mahjong mask

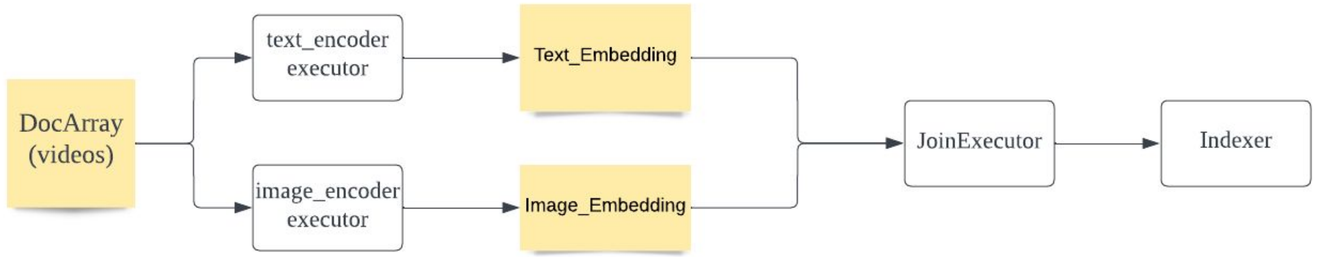
[Bilibili：输入关键词就能自动剪视频？我写了一个AI视频搜剪神器？]

[Github：Jina-Clip]

用Jina快速搭建视频搜索demo

- **How to do it?**

INDEX



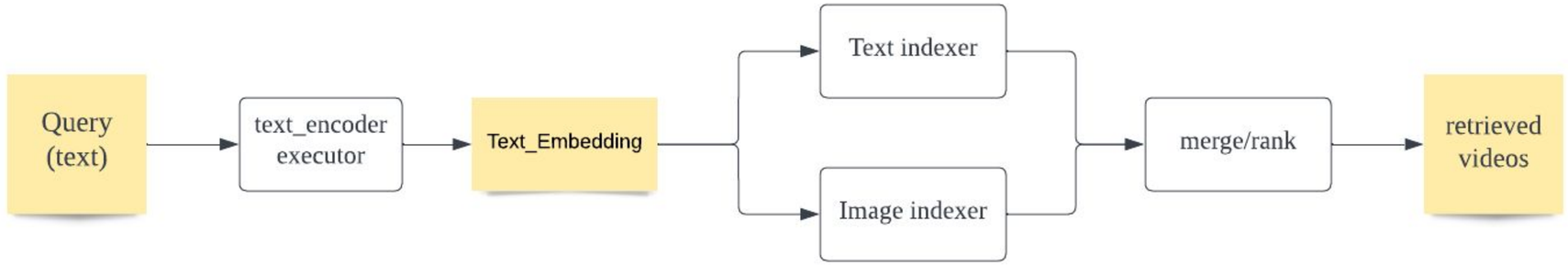

```python
@requests
def encode(self, docs: DocumentArray, **kwargs):
    """
    Encode all documents with the `text` attribute and store the embeddings in the
    `embedding` attribute.
    :param docs: DocumentArray containing the Documents to be encoded
    """
    for docs_batch in docs.batch(batch_size=128):
        text_batch = docs_batch.texts
        with torch.inference_mode():
            embeddings = self.model(text_batch)
            for doc, e in zip(docs_batch, embeddings):
                doc.embedding = e
```

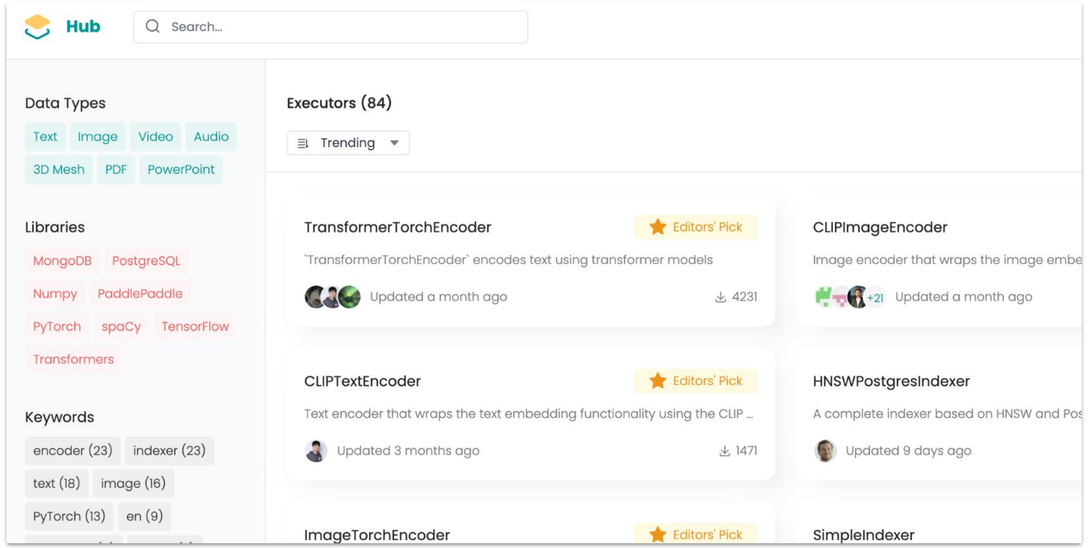
```python
@requests(on='/index')
def index(self, bdocs: 'DocumentArray', **kwargs):
    """
    All Documents to the DocumentArray
    :param docs: the docs to add
    """
    if docs:
        self._index.extend(docs)
```

用Jina快速搭建视频搜索demo

- **How to do it?**

QUERY





1 多模态数据vs单模态数据

2 学术界如何处理多模态数据

3 多模态技术在工业界的应用

4 **Jina在多模态场景下的应用**

- 搭建视频搜索demo
- **DALLE Flow**

DALLE Flow

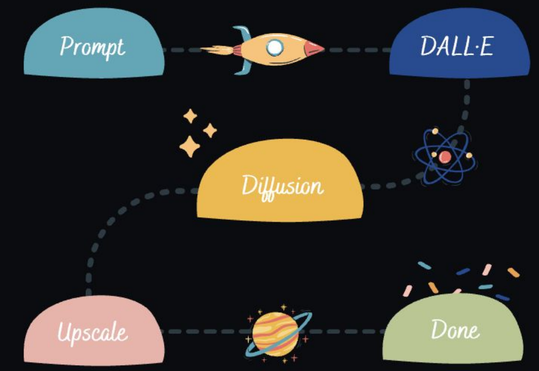
Quick Facts

- Client-server architecture
- Combine DALLE & Diffusion using Jina
- 288 Lines of Python
- 42 Lines of YAML
- a weekend-work

jina-ai / dalle-flow Public

Unwatch 39 | Fork 100 | Star 1.9k

README.md



A Human-in-the-Loop[?] workflow for creating HD images from text

Slack 3.1k | Open in Colab | docker build failing | image size 17.9 GB

DALL·E Flow is an interactive workflow for generating high-definition images from text prompt. First, it leverages DALL·E-Mega to generate image candidates, and then calls CLIP-as-service to rank the candidates w.r.t. the prompt. The preferred candidate is fed to GLID-3 XL for diffusion, which often enriches the texture and background. Finally, the candidate is upscaled to 1024x1024 via SwinIR.

DALL·E Flow is built with Jina in a client-server architecture, which gives it high scalability, non-blocking streaming, and a modern Pythonic interface. Client can interact with the server via gRPC/Websocket/HTTP with TLS.

Why Human-in-the-Loop? Generative art is a creative process. While recent advances of DALL·E unleash people's creativity, having a single-prompt-single-output UX/UI locks the imagination to a *single* possibility, which is bad no matter how fine this single result is. DALL·E Flow is an alternative to the one-liner, by formalizing the generative art as an iterative procedure.

```yaml
      - executors/glid3/executor.py
    env:
      CUDA_VISIBLE_DEVICES: 0  # change this if you have multiple GPU
      XLA_PYTHON_CLIENT_ALLOCATOR: platform  #
https://jax.readthedocs.io/en/latest/gpu_memory_allocation.html
    replicas: 1
    needs: [gateway]
  - name: rerank
    uses: ReRank
    uses_with:
      clip_server: grpcs://demo-cas.jina.ai:2096
    py_modules:
      - executors/rerank/executor.py
    needs: [dalle, diffusion]
  - name: upscaler
    uses: SwinIRUpscaler
    py_modules:
      - executors/swinir/executor.py
    uses_with:
      swinir_path: ../SwinIR
      store_path: dalle.db
    env:
      CUDA_VISIBLE_DEVICES: 0  # change this if you have multiple GPU
  - name: store
    uses: MyStore
    py_modules:
      - executors/store/executor.py
    uses_with:
      store_path: dalle-flow.db
```

```yaml
jtype: Flow
with:
  protocol: grpc
  port: 51005
  env:
    JINA_LOG_LEVEL: debug
executors:
  - name: dalle
    uses: executors/dalle/config.yml
    timeout_ready: -1  # slow download speed often leads to timeout
    env:
      CUDA_VISIBLE_DEVICES: 0  # change this if you have multiple GPU
      XLA_PYTHON_CLIENT_ALLOCATOR: platform  #
  https://jax.readthedocs.io/en/latest/gpu_memory_allocation.html
  - name: diffusion
    uses: GLID3Diffusion
    uses_with:
      glid3_path: ../glid-3-xl
      steps: 100
    py_modules:
      - executors/glid3/executor.py
    env:
      CUDA_VISIBLE_DEVICES: 0  # change this if you have multiple GPU
      XLA_PYTHON_CLIENT_ALLOCATOR: platform  #
  https://jax.readthedocs.io/en/latest/gpu_memory_allocation.html
    replicas: 1
    needs: [gateway]
  - name: rerank
    uses: ReRank
    uses_with:
      clip_server: grpcs://demo-cas.jina.ai:2096
    py_modules:
      - executors/rerank/executor.py
    needs: [dalle, diffusion]
  - name: upscaler
    uses: SwinIRUpscaler
    py_modules:
      - executors/swinir/executor.py
    uses_with:
      swinir_path: ../SwinIR
      store_path: dalle.db
    env:
      CUDA_VISIBLE_DEVICES: 0  # change this if you have multiple GPU
  - name: store
    uses: MyStore
    py_modules:
      - executors/store/executor.py
    uses_with:
      store_path: dalle-flow.db
```

Flow configuration

- port
- protocol
- environment variables

Flow spec

DALLE

- init kwargs
- GPU config
- executor config

Diffusion

- init kwargs
- GPU config
- executor config

Rerank

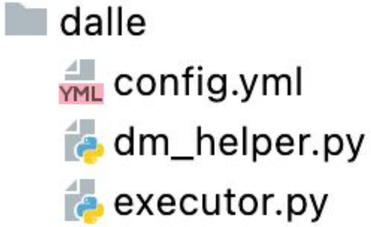
- init kwargs
- executor config

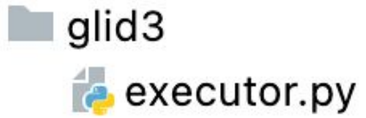
Upscaler

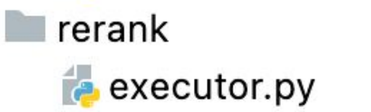
- init kwargs
- executor config

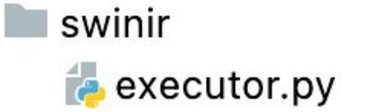
Store

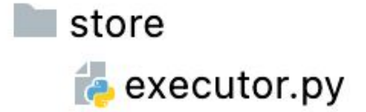
- init kwargs
- executor config











"ocean beach front view in Van Gogh style"

Github jina-ai/dalle-flow

"the statue of liberty wearing a vr headset"

Github jina-ai/dalle-flow

欢迎加入 Jina AI 开源社区 解锁数据应用的可能性

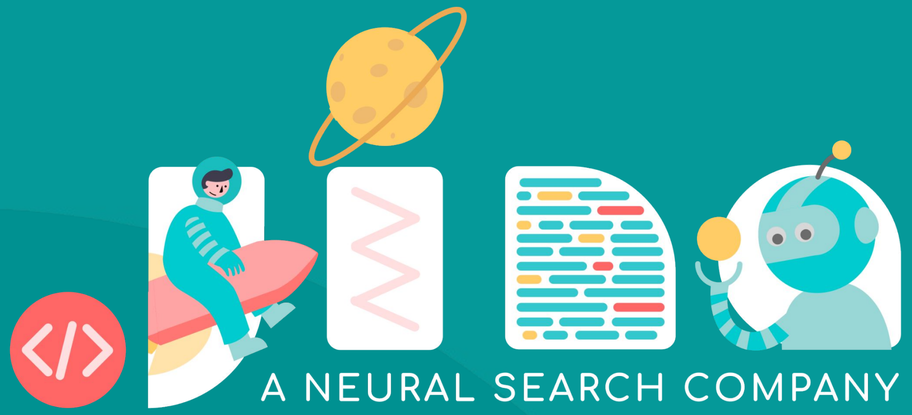
官网：https://jina.ai/

GitHub：https://github.com/jina-ai

加入全球开发者社区：https://slack.jina.ai/

扫码关注 Jina AI 公众号

添加小助手加入技术交流群



Thank you!

Global jina.ai @JinaAI_ get.jina.ai slack.jina.ai