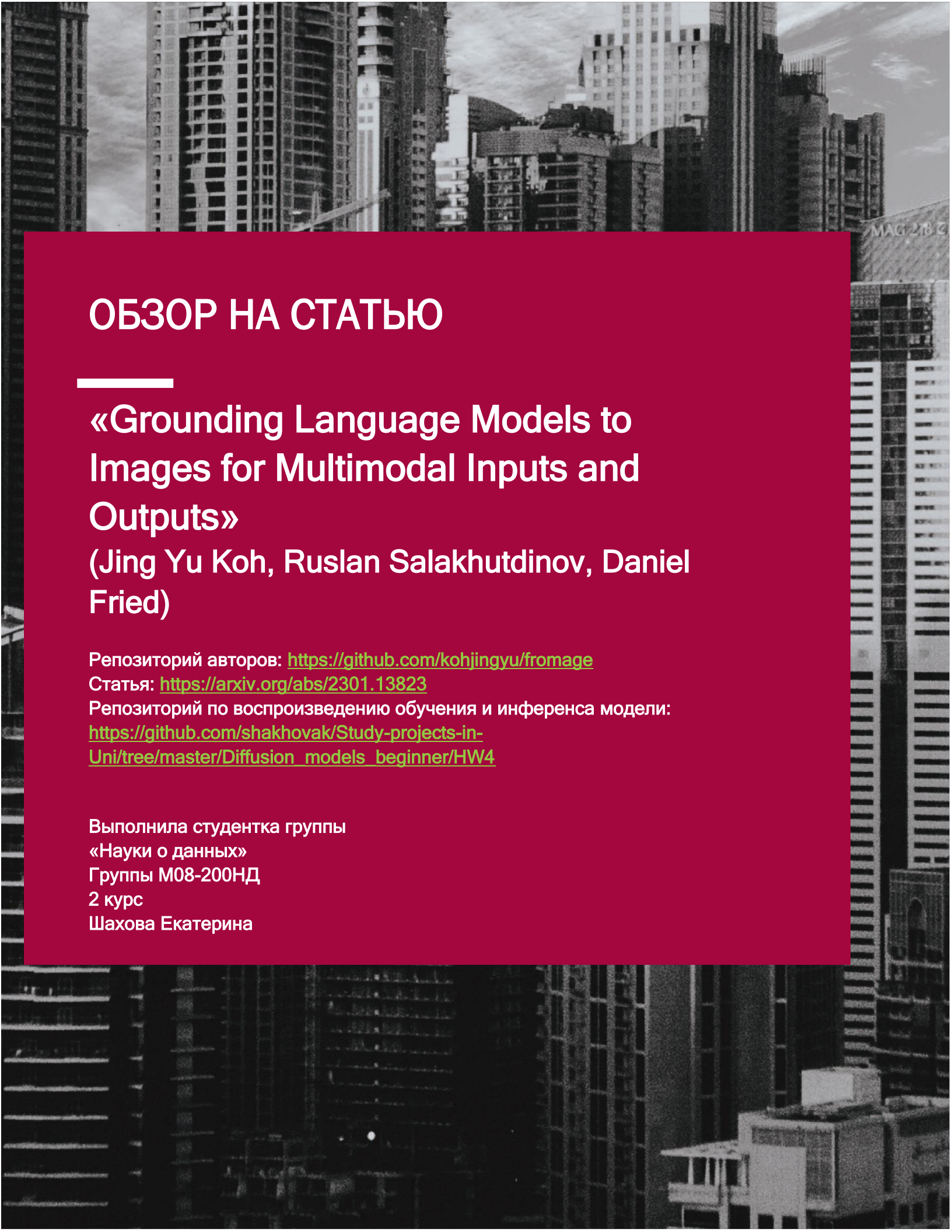


ОБЗОР НА СТАТЬЮ

«Grounding Language Models to Images for Multimodal Inputs and Outputs»

(Jing Yu Koh, Ruslan Salakhutdinov, Daniel Fried)

Репозиторий авторов: https://github.com/kohjingyu/fromage
Статья: https://arxiv.org/abs/2301.13823
Репозиторий по воспроизведению обучения и инференса модели: https://github.com/shakhovak/Study-projects-in-Uni/tree/master/Diffusion_models_beginner/HW4

Выполнила студентка группы
«Науки о данных»
Группы М08-200НД
2 курс
Шахова Екатерина

СОДЕРЖАНИЕ

Введение

Современные языковые модели (LLMs) в последнее время достигли больших успехов в решении многих задач в NLP сфере, связанных с обработкой естественного языка. Перед исследователями теперь стоит задача расширения спектра модальностей, используемых моделями. Мультимодальные языковые модели (MLLMs) учатся создавать истории на основе изображений и вести диалог в разных модальностях. Исследования также направлены на разработку способов сокращения ресурсоемкости MLLMs.

В рамках курса по современным моделям машинного обучения нам предложено рассмотреть три статьи[1], датированные 2023 годом, касающиеся разных подходов к встраиванию новых модальностей в языковые модели. Я решила остановиться на статье Grounding Language Models to Images for Multimodal Inputs and Outputs (Koh J., Salakhutdinov R., Fried D., 2023). В статье авторы предлагают достаточно простой и необычной способ соединения замороженной языковой модели с также замороженным визуальным энкодором изображений через линейный слой. Данный слой обучается на задачи image captioning и image text retrieval. Особенно привлекло то, что для такого обучения не нужно много ресурсов и в результате модель способна вести мультимодальный диалог и строить выводы на основе визуальных данных (рис. 1).

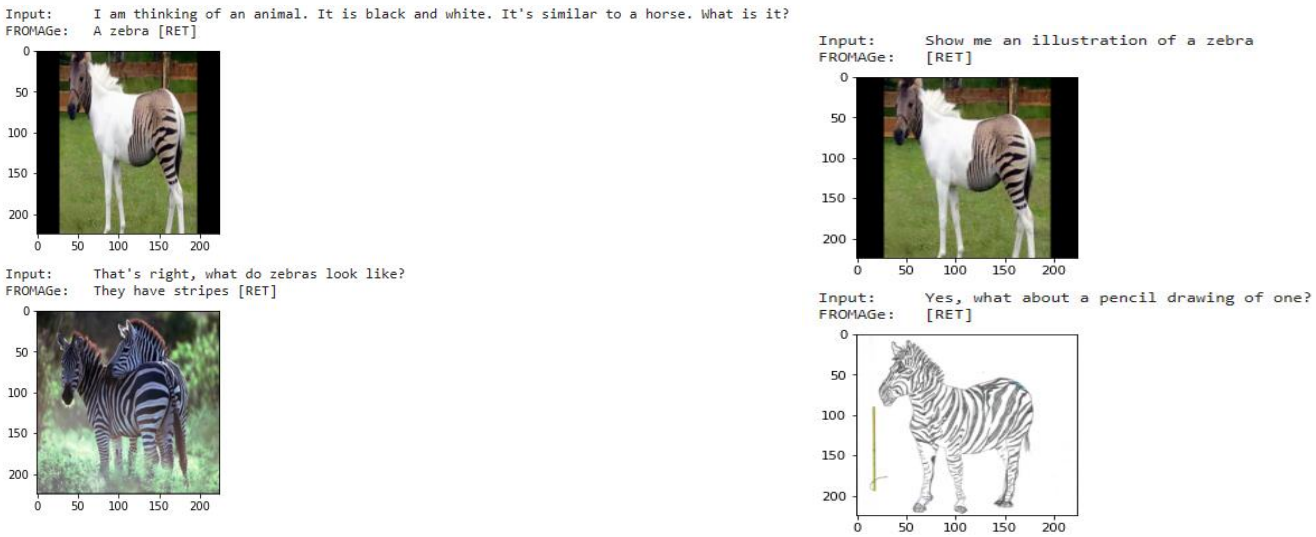


Рис 1. Диалог модели после подачи в качестве промпта в виде изображения

Авторы назвали свою модель FROMAGe (Frozen Retrieval Over Multimodal Data for Autoregressive Generation). Для проверки работы в инференсе я сделала клон соответствующего репозитория и запустила несколько задач, таких как генерация по промпту в виде изображения и мультимодальный диалог. Ноутбук с инференсом (ссылка)

[1] Две другие статьи:
ImageBind: One Embedding Space To Bind Them All – статья, посвященная созданию единого пространство векторных представлений для разных модальностей
Language Is Not All You Need: Aligning Perception with Language Models – статья, касающаяся методики встраивания новых модальностей в LLMs

запускается из клона репозитория авторов при дополнительной установке визуальных эмбеддингов для обучающей выборки (их нет в github, ссылка на скачивание здесь). После скачивания их необходимо сохранить в папку fromage_model в клоне репозитория для корректной работы блокнота в инференсе.

Предлагаю рассмотреть более детально методологию авторов и попробовать повторить цикл создания и обучения их модели (ноутбук с детальным обучением здесь).

Данные для обучения

В качестве обучающей выборки авторы взяли датасет Conceptual Captions (CC3M, который включает в себя 3,3 млн пар текст-изображение (рис 2). Так как изображение представлено в виде url и нужно время, чтобы его скачать, то для воспроизведения эксперимента авторов я воспользуюсь только небольшим семплингом из 500 пар. Предобработку и сохранение семплинга датасета с Hugging Face можно посмотреть в ноутбуке (ссылка).

Рис 2. Пример изображений и их описаний из датасета CC3M

Архитектура модели

Как видно из рис. 3 архитектура модели, представленной авторами, имеет 3 основных компонента:

- LLM - Языковая модель из семейства OPT, обученная на предсказание токена в маскированной последовательности. Все слои модели заморожены.
- CLIP – Визуальный энкодер для извлечения информации из изображений. Все слои модели также заморожены
- Trained layers – дополнительные линейные слои, веса которых обновляются во время обучения. Данные слои ответственны за меппинг текста и изображения: text hidden fcs за меппинг image-to-text и visuals fc за text-to-image.

[27]:
```
Fromage(
  (model): FromageModel(
    (lm): OPTForCausalLM(
      (model): OPTModel(
        (decoder): OPTDecoder(
          (embed_tokens): Embedding(50267, 768)
          (embed_positions): OPTLearnedPositionalEmbedding(2050, 768)
          (final_layer_norm): LayerNorm((768,), eps=1e-05, elementwise_affine=True)
          (layers): ModuleList(
            (0-11): 12 x OPTDecoderLayer(
              (self_attn): OPTAttention(
                (k_proj): Linear(in_features=768, out_features=768, bias=True)
                (v_proj): Linear(in_features=768, out_features=768, bias=True)
                (q_proj): Linear(in_features=768, out_features=768, bias=True)
                (out_proj): Linear(in_features=768, out_features=768, bias=True)
              )
              (activation_fn): ReLU()
              (self_attn_layer_norm): LayerNorm((768,), eps=1e-05, elementwise_affine=True)
              (fc1): Linear(in_features=768, out_features=3072, bias=True)
              (fc2): Linear(in_features=3072, out_features=768, bias=True)
              (final_layer_norm): LayerNorm((768,), eps=1e-05, elementwise_affine=True)
            )
          )
        )
      )
      (lm_head): Linear(in_features=768, out_features=50267, bias=False)
    )
    (input_embeddings): Embedding(50267, 768)
    (visual_model): CLIPVisionModel(
      (vision_model): CLIPVisionTransformer(
        (embeddings): CLIPVisionEmbeddings(
          (patch_embedding): Conv2d(3, 1024, kernel_size=(14, 14), stride=(14, 14), bias=False)
          (position_embedding): Embedding(257, 1024)
        )
        (pre_layrnorm): LayerNorm((1024,), eps=1e-05, elementwise_affine=True)
        (encoder): CLIPEncoder(
          (layers): ModuleList(
            (0-23): 24 x CLIPEncoderLayer(
              (self_attn): CLIPAttention(
                (k_proj): Linear(in_features=1024, out_features=1024, bias=True)
                (v_proj): Linear(in_features=1024, out_features=1024, bias=True)
                (q_proj): Linear(in_features=1024, out_features=1024, bias=True)
                (out_proj): Linear(in_features=1024, out_features=1024, bias=True)
              )
              (layer_norm1): LayerNorm((1024,), eps=1e-05, elementwise_affine=True)
              (mlp): CLIPMLP(
                (activation_fn): QuickGELUActivation()
                (fc1): Linear(in_features=1024, out_features=4096, bias=True)
                (fc2): Linear(in_features=4096, out_features=1024, bias=True)
              )
              (layer_norm2): LayerNorm((1024,), eps=1e-05, elementwise_affine=True)
            )
          )
        )
        (post_layernorm): LayerNorm((1024,), eps=1e-05, elementwise_affine=True)
      )
    )
    (text_hidden_fcs): ModuleList(
      (0): Sequential(
        (0): Linear(in_features=768, out_features=256, bias=True)
        (1): Dropout(p=0.0, inplace=False)
      )
    )
    (visual_embeddings): Linear(in_features=1024, out_features=768, bias=True)
    (visual_fc): Linear(in_features=1024, out_features=256, bias=True)
    (image_dropout): Dropout(p=0.0, inplace=False)
  )
)
```

LLM

CLIP

Trained layers:

Методика обучения

Модель последовательно обучается на двух задачах: image captioning и image text retrieval (рис 4).

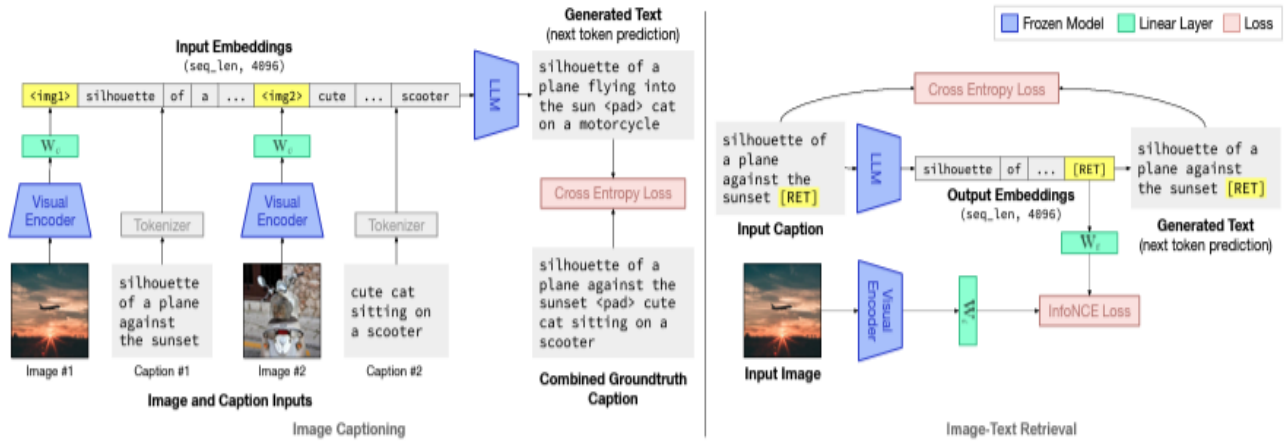


Рис 4. Основные задачи обучения для модели

Image captioning представляет из себя классическую задачу обучения предсказания следующего токена, основываясь на соединенных токенах изображения и текста с частично маскированным текстом. В качестве loss функции используется negative loss likelihood, который рассчитывается по формуле ниже для всех пар в батче (авторы в коде называют ее CeLoss):

$$\mathcal{L}_c = -\frac{1}{N}\sum_{i=1}^{N} l_c(x_i, y_i)$$

Image text retrieval представляет из себя задачу подбора изображения при условии полученного текста. При обучении используются конкатенированные вектора текстового описания и изображения вместе со специальным токеном. С помощью contrastive loss функции вектора, полученные из последнего слоя языковой модели (x), сравниваются с векторами, полученного из энкодера (y). Расчет данной loss функции включает расчет косинусной близости между векторами:

$$\text{sim}(x, y) = \frac{(h_\theta(x)^T \mathbf{W}_t)(v_\phi(y)^T \mathbf{W}_i)^T}{\|h_\theta(x)^T \mathbf{W}_t\| \|v_\phi(y)^T \mathbf{W}_i)^T\|}$$

На основе полученной косинусной близости определяется InfoNCE[2] loss для всего батча на задачах text to image и image to text (в коде авторы называют его ContLoss):

$$\mathcal{L}_{\text{t2i}} = -\frac{1}{N}\sum_{i=1}^{N}\left(\log\frac{\exp(\text{sim}(x_i, y_i)/\tau)}{\sum_{j=1}^{N}\exp(\text{sim}(x_i, y_j)/\tau)}\right)$$

$$\mathcal{L}_{\text{i2t}} = -\frac{1}{N}\sum_{i=1}^{N}\left(\log\frac{\exp(\text{sim}(y_i, x_i)/\tau)}{\sum_{j=1}^{N}\exp(\text{sim}(y_i, x_j)/\tau)}\right)$$

Так как обучение проходит последовательно для каждой задачи в рамках одного батча, то лоссы от каждой задачи суммируются. Авторы предлагают также использовать гиперпараметры для взвешивания каждого вида функции потерь, чтобы больше управлять обучением и варьировать результаты.

$$\mathcal{L} = \lambda_c\mathcal{L}_{\text{c}} + \lambda_r(\mathcal{L}_{\text{t2i}} + \mathcal{L}_{\text{i2t}})$$

Авторы обучали модель в течение 1 эпохи с батчем размера 180 позиций (18 тыс. шагов на 1 эпоху). При повторении эксперимента я обучала модель на 6 эпохах и батче в 16 позиций. Все остальные параметры обучения, такие как learning rate, optimizer и т. д. я оставила неизменными, какие были указаны авторами.

В итоге у меня получилось повторить цикл обучения, хотя на небольшом наборе данные (500 пар) результаты получились не очевидными.

Benchmarking

После обучения авторы тестировали возможности модели на таких задачах, как генерация изображений исходя из мультимодального контекста. Для такого рода задач нет установленных метрик, поэтому в качестве сравнения исследователи предложили сравнивать результаты работы по точности подбора правильного изображения, рассчитанным с помощью Recall метрики. Результаты сравнивали с моделью CLIP при использовании вариаций входных данных в виде только текста или текста + изображение (рис. 7).

2 InfoNCE, где NCE обозначает Noise-Contrastive Estimation – тип контрастного лосса

Model	Inputs	R@1	R@5	R@10
CLIP ViT-L/14	1 caption	**11.9**	**25.5**	**32.2**
FROMAGe		11.3	24.6	32.1
CLIP ViT-L/14	5 captions	5.9	19.5	28.0
FROMAGe		**11.9**	**23.8**	**31.7**
BLIP†	5 captions	6.2	16.8	23.4
CLIP ViT-L/14†	5 captions	8.8	22.3	29.8
FROMAGe †	5 captions	13.2	28.5	36.7
CLIP ViT-L/14	5 captions, 4 images	2.4	21.3	34.0
FROMAGe †	5 captions, 4 images	**18.2**	**42.7**	**51.8**

Рис. 7. Сравнение качества подбора изображения через метрику Recall@k[3]. Специальным значком отмечены случаи, когда подбираются изображения, которых раньше не было в запросе.

FROMAGe значительно лучше сработал при мультимодальных запросах и при большем контексте.

Другой эксперимент автором подразумевал тестирование модели в рамках мультимодального диалога (рис.8) и включал задачи:

- IT2T - выбрать правильный ответ из 100 возможных на вопрос, содержащий текст и изображения, метрика Recall@k, NDCG[4], MRR[5]
- T2I - выбрать подходящее изображение в качестве ответа на вопрос из текста, метрика Recall@k
-

			IT2T					T2I		
Model	Trainable Params	Finetuning Data	NDCG	MRR	R@1	R@5	R@10	R@1	R@5	R@10
ViLBERT (Lu et al., 2019)	114M	3.1M	11.6	6.9	2.6	7.2	11.3	-	-	-
CLIP ViT-L/14 (Radford et al., 2021)	300M	400M	10.9	8.5	3.1	8.7	15.9	17.7	38.9	50.2
Flamingo (Alayrac et al., 2022)	10.2B	1.8B	**52.0**	-	-	-	-		Incapable	
ESPER (Yu et al., 2022b)	4M	0.5M	22.3	**25.7**	14.6	-	-		Incapable	
FROMAGe (ours)	5.5M	3.1M	16.5	22.0	**17.6**	**20.1**	**25.1**	**20.8**	**44.9**	**56.0**

Рис.8. Сравнение качества ведения моделью мультимодального диалога с помощью zero-shot

[3] Recall@k - специальная метрика, которая показывает пропорцию подходящих позиций из рекомендованных (кол-во рекомендаций равно k) и общее кол-во позиций.
[4] NDCG Normalized Discounted Cumulative Gain - метрика качества ранжирования.
[5] MRR mean reciprocal rank - также является метрикой качества ранжирования по релевантности.

В целом модель показывает неплохие результаты по сравнению со своими более тяжеловесными коллегами с учетом ресурсов на обучение. На мой взгляд, успешное использование авторами концепции встраивания модальностей через небольшие линейные слои подтверждает, что данное направление в обучении MLLMs имеет перспективы.

Преимущества и недостатки

При повторении эксперимента авторов я сразу обратила внимание на достаточно простую концепцию самой модели и процесса обучения. Итоговая модель не является тяжеловесной и в ней нет большого количества дополнительных слоев. Изменяя веса только линейного слоя, авторы добились отличной экономии ресурсов при сопоставимых результатах с более сложными моделями. Даже в качестве учебного эксперимента мне удалось довольно быстро обучить модель на несколько эпох.

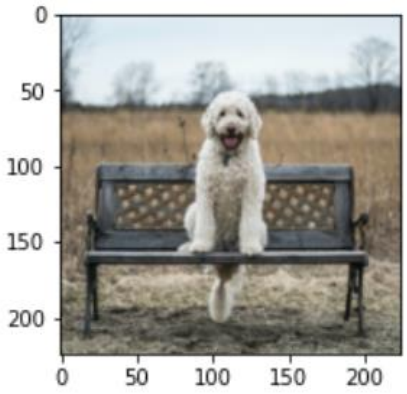
При этом модель наследует все недостатки языковых моделей, такие как генерация повторяющихся данных (рис. 9) и игнорирование запроса пользователя.

Рис.9. Пример проблемной генерации при «недоученном» линейном слое при повторении цикла обучения, предложенного авторами

Помимо прочего авторы не реализовали в модели возможность генерации новых изображений, поэтому модель ограничена теми изображениями, на которых она обучалась. На рисунке 10 показан результат работы модели при запросе показать похожие изображения на то, которое она не видела в обучающей выборке. Именно

поэтому и существует необходимость использовать готовые визуальные эмббединги для работы в инференсе, что не добавляет модели удобства и простоты в использовании. Увеличение обучающей выборки повлечет увеличение размера файла с эмббедингами для инференса и дополнительные сложности при попытке, например, встроить модель в пользовательское приложение.

С другой стороны, отсутствие возможности генерации новых изображений помогает лучше контролировать результаты работы модели, ограничивая неэтичные и агрессивные результаты уже на этапе формирования учебной выборки.




```
Input:     What is this image?
FROMAGe:   This is a dog sitting on a bench.
Input:     Is the image in color?
FROMAGe:   No, it is black and white.
Input:     Where do you think it was taken?
FROMAGe:   It was taken in a park.
Input:     What about a photo of similar dogs?
FROMAGe:   I have not seen any similar dogs.
Input:     Show me any other dog
FROMAGe:   I have not seen any other dog.
```

Рис. 10. Неудачная попытка запросить похожее изображение у обученной авторами модели (здесь уже модель из репозитория авторов)

Другой возможной причиной, почему модель не смогла подобрать похожие изображения, может быть, большая предрасположенность модели к генерации текста, чем изображений при превалирующем использовании текстовой информации на этапе обучения LLMs. В качестве возможного решения этой проблемы авторы предлагают использовать дополнительный множитель для логитов изображений, чтобы утяжелять их по сравнению с текстовыми.

Одним из преимуществ модели, предложенной авторами, является также возможность использовать разные LLM и энкодеры, так как они выбираются в виде параметров. Это

позволяет как уменьшать, так и увеличивать параметры конечной модели, как сделала я, выбрав меньшую языковую модель для экономии ресурсов при повторе эксперимента авторов.

Исходя из недостатков и преимуществ модели, можно говорить и о дальнейших перспективах этого исследования. Одним из вариантов является добавление возможности генерации новых изображений, а не только текста. Данное направление является перспективным и попытка попробовать добавить в архитектуру, например модели диффузионного типа, может принести интересные результаты. Хотя есть вероятность утяжеления модели.

Также имеет смысл понять, как можно заменить визуальные эмббединги, так как при увеличении количества изображений в обучающей выборке будет расти их размер и довольно проблематично станет использовать модель в приложениях. Увеличение же обучающей выборки позволит сделать модель более универсальной.

Заключение.

Предложенный авторами метод добавления модальностей в языковую модель показал себя достаточно эффективным при экономном использовании ресурсов. На таких задачах как генерация по изображению и ведению мультимодального диалога, модель показала неплохие результаты, хотя не без недостатков.

Архитектура модели и процесс обучения достаточно просты, чтобы можно было детально воспроизвести эксперименты авторов, меняя итоговые параметры модели в сторону увеличения или уменьшения.

В целом, исследование мне кажется перспективным с потенциалом для дальнейшего развития.