Supplementary Material for Open-domain Visual Entity Recognition: Towards Recognizing Millions of Wikipedia Entities

Hexiang Hu[♠] Yi Luan[♠] Yang Chen[♠♡†] Urvashi Khandelwal[♠]
Mandar Joshi[♠] Kenton Lee[♠] Kristina Toutanova[♠] Ming-Wei Chang[♠]

♠ Google Deepmind ♡ Georgia Institute of Technology

1. Dataset Construction, Annotation, and Additional Statistics

In this section, we describes the complete details on data collection, curation, entity linking, and show additional statistics of the processed dataset (§1.1). Then we also discuss how we train annotators to annotate our task, and provide the concrete annotation interface(§1.2).

1.1. Data Collection & Pre-processing

Some of our member datasets have been reported to include non-imageable classes, classes with undesired social bias [11], or non-entity classes (*e.g.*, numbers). Therefore, we apply a filtering process to compose our dataset, based on the individual condition of each source dataset. Overall, to create the Entity split, we first apply a general safety filter [11] to remove non-imageable labels, non-entity labels, and labels with social bias. To create the Query split, we employed three expert annotators to write heuristic policies to filter each VQA dataset, and ensure our task is focusing on entity related questions. Concretely, questions related to counting, verification, or querying non-entity attributes (*e.g.*, dates), are removed. Then we apply the same safety filter.

Based on the filtered data, we developed a two-staged entity linking strategy to connect the label text to Wikipedia entities, on both Entity and Query splits. First, we obtain exact match based entity candidates by querying the Wikipedia search API (with the auto-suggestion disabled) with the raw label text. We reject candidates whose landing pages are identified as disambiguation pages. The Wikipedia API[1] automatically redirects queries (in our case, labels) matching entity aliases to their canonical form. For the labels which do not have an exact match in Wikipedia, we use a state-of-the-art text-based entity linker (*i.e.*, GENRE [4]) to obtain top candidate Wikipedia entity names. Finally, we link the label to the top ranked entity whose landing page is not a disambiguation page.

Using the entity linking process described earlier, we successfully connect a total of 24,895 class labels in OVEN-Wiki to corresponding Wikipedia entities. Overall, our dataset contains 20,801 unique entities. For the Entity split data, we generate a synthetic text query based on the super-category information of the label (either provided by source dataset or mined from Wikidata[2]), using templated language. For example, iNaturalist has provided detailed supercategory annotation on each class, such as `Plantae`, `Reptilia`, *etc*. For dataset that do not provide this information, we use the super-category mined from Wikidata, which is publicly crowd sourced and maintained. As a result, our templated query generator produces the query ``` ``what is the species of the plant in this image?'' ``` for the entity ``` ``Eryngium alpinum'' ```, whose super-category is `Plantae`. Due to space limit, we provide more explanation in Appendix. For all Wikipedia entities, we use the corresponding Wikipedia page and its associating multi-media content (*e.g.*, information box images, *etc*.) as the source of *multi-modal knowledge* about entities.

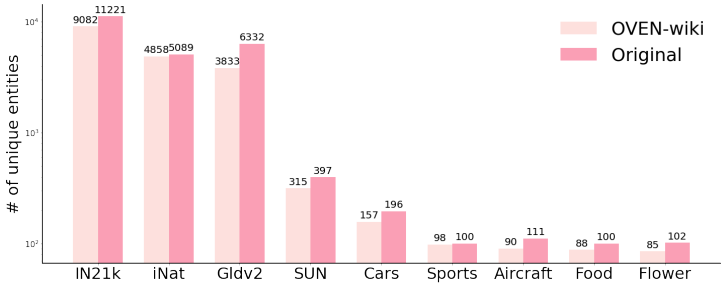
Specifically, Figure 1 shows the number of unique entities in both the Entity and Query splits, where we compare the total number of entities in each source dataset against its original population (after applied safety filter). Note that for the Google Landmarks v2 (Gldv2) dataset, we employed

† Work was done when interned at Google.

‡ Our dataset and evaluation toolkit is publicly available at https://open-vision-language.github.io/oven

[1] https://www.mediawiki.org/wiki/API

[2] Available at https://www.wikidata.org/wiki/Wikidata



	# Original Answers	# Entity Answers
VG	50,130	3,460
OK-VQA	4,214	1,600
Text-VQA	19,500	3,562
VQA v2	26,748	4,337
Visual7W	7,588	1,945

Figure 1: Number of unique entities on Entity split (left) and Query split (right). We compare it against the # of entities before applying pre-processing. Note that VQA datasets contain massive non-entity answers, or collapsed answers, which leads to a large reduction in numbers after pre-processing.

the cleaned data split from [12], where the total number of unique entities is significantly reduced. Because Gldv2 is automatically generated and has reported to contain noises particularly with tail entities [12], we removed entities with less than 50 instances for a improved precision (further reduces the # of entities in Gldv2 to ∼6k).

To give more details for the Figure **??** in the main text, we further present full super-category grouping information in Figure 2. As aforementioned, we have combined entities that belongs to general groups (*e.g.*, "object", "item" groups) or unpopular groups (*e.g.*, groups with less than 5 entities) into the "others" group. We also merged some sub-categories into super-categories, *e.g.*, "location"+"park"+"lake"+"river"+"mountain"→"location", "building"+"bridge"→"building".

1.2. Human Annotation Procedure & Interface

In order to verify the quality of OVEN-Wiki and to provide a human verified test set to evaluate on, we conduct human annotation on a subset of test set. The annotators are asked to correct the errors in the ¡image, query, answer¿ triplets. The details are as follows.

Annotation interface Figure 3 illustrates the annotation interface. The left side of Figure 3 are the input to the annotators which includes the original question, image and the answer (together with the wikipedia hyperlink). The annotators are asked to complete the following questions:

1. *Does the Wikipedia represent the correct meaning of the answer? Provide the Wikipedia link if not.*

   This question requires the annotators to correct the entity linking errors. The annotators use Google search to find the most suitable Wikipedia link if the provided one is not adequate. In our dataset, 8.4% of the entity links are reported wrong by more than 2 annotators, which are manually corrected later.

2. *Is the Wikipedia answer physically present in the image.*

   This question is mainly aimed at filtering out the OCR examples which are out of our scope. One example is that the image about a wall painted with the word "love" and the linked entity is the "love" Wikipedia. In our dataset, 10.3% of the answers are reported not physically present in the image by more than 2 annotators, which are discarded from the human evaluation set.

3. *Rewrite the question so that no other object can be the answer.*

   The annotators will rewrite the question is the answer is wrong or ambiguous. Annotators will make sure that the question can not be answered without the image and that the answers can not be included in the rewritten questions. In our annotation, 99.9% of the questions are being rewritten.

Instruction and Training We carefully design the training procedure to improve the annotation quality. We first conduct a "Self-study session" where the annotators will read the instructions and annotate a few toy examples. Then we conduct a "In-person tutorial" where we have an online video session in which we walk annotators through the full version of the instructions and discuss mistakes made in the self-study annotations. Finally we conduct a "Test exam" and the qualified annotators are accepted. In total, 30 annotators went through our training procedure and all of them were eventually accepted to work full-time on the main task.

Quality control We have a three way annotations where each examples are annotated by three annotators. We were giving regular feedback on the questions the annotators may have during the annotation and pointed out mistakes identified in annotators' past answers.

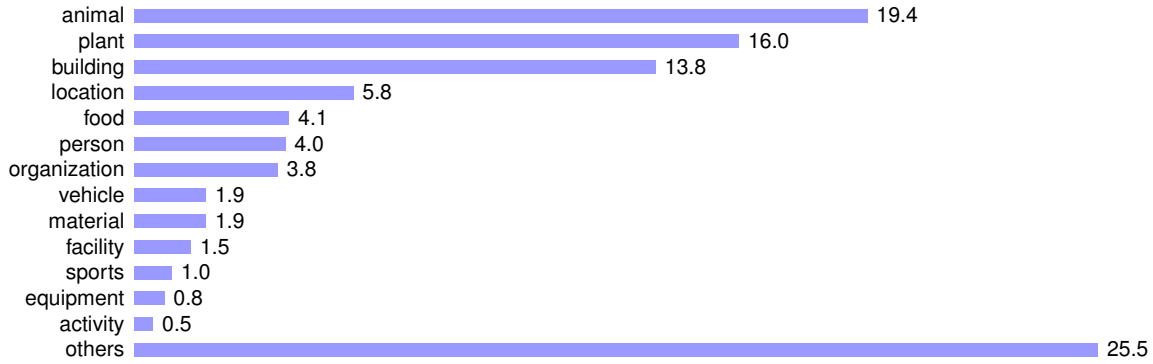


Figure 2: Distribution of the entities in our datasets (Grouped by their super category).

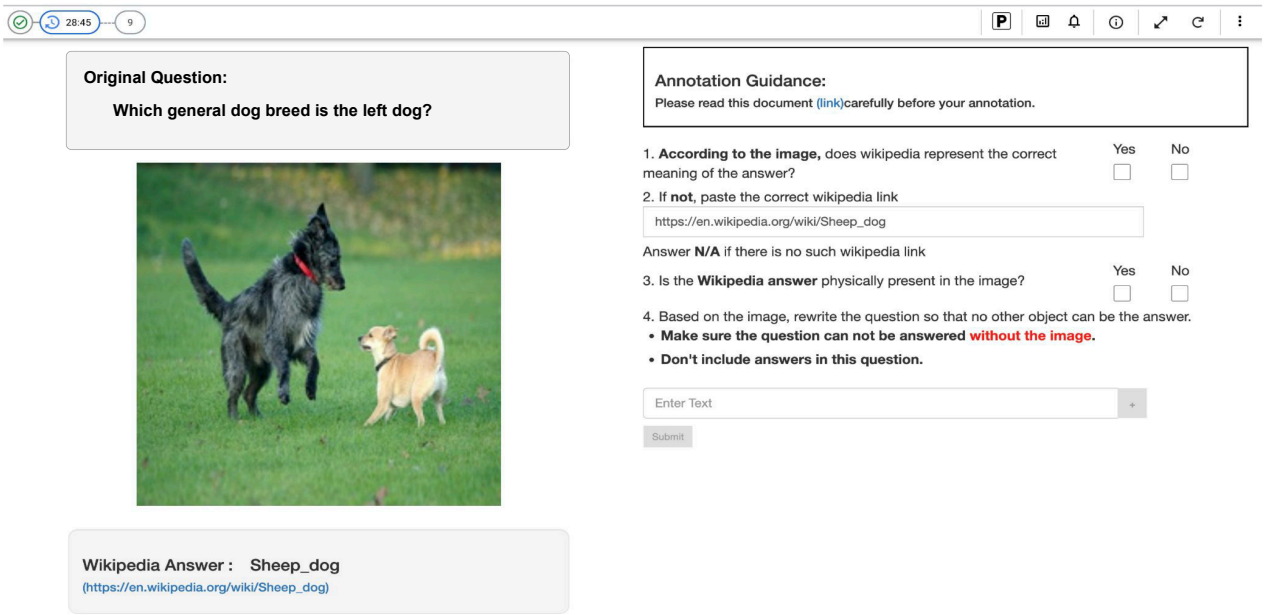


Figure 3: Annotation inferface

On average, it took annotators 4.6 minutes to answer each question with the time consumption slightly decreasing as annotators get familiar with the task. The compensation rate for the task was set to be $17.8/hour which is higher than the minimum hourly wage in the US.

We filtered out all the examples where the wikipedia links are marked as wrong or the Wikipedia answers are marked as "Not physically present in the image".

2. Implementation Details of the baseline systems

In this section, we provide implementation details on the CLIP variants and PaLI model for the OVEN task.

2.1. CLIP Fusion Model

As aforementioned, we implemented this multi-modal dual encoder via taking pre-trained CLIP image and text encoders as featurizers. The CLIP model is based on a ViT-Large, with a total of over 400M parameters, pre-trained on a 400M prviate image-text dataset collected by OpenAI. Based on this model, we build two 2-layer Transformer models, on top of two CLIP models as the left and right encoder, for encoding the query representation and the entity representation, respectively. The 2-layer Transformers follows the same architecture as T5 Transformer [8], but with 2 layers, 12 attention heads, with each attention head of 64 dimensions, and the embedding size of 768. We then fine-tune this

composed model on the OVEN-Wiki's training data, using a in-batch contrastive learning objective [8], with a batch size of 4,096. We optimize the model for 10K steps in the fine-tuning stage, with Adafactor optimizer [9] and a initial learning rate of 0.001. There are 1k steps for the warmup, followed by a square root LR decay schedule with final learning rate of 1e-6.

2.2. CLIP2CLIP Model

Different from CLIP Fusion, CLIP2CLIP is a model that adds minimum new parameters to the pre-trained CLIP encoders. Same as other models, we initialize both the query encoder and the target encoder separately with the pre-trained CLIP model. Specifically, we use the pre-trained CLIP encoders for both left and right encoders, to encode the image and text modality for both the query representation and the entity representation. We then compute the four dot product similarity scores on the <input image, target text>, <input text, target image>, <input image, target image>, and <input text, target text> pairs, which is then combined via a learnable similarity weights into one logit score. For the Wikipedia images used in the retrieval, we apply the same image processing pipeline whenever the image is available. When the Wikipedia entity does not have an infobox image, we use a black image to represent the visual support.

To make sure that the learnable similarity weights is initialized properly, we perform a grid search to find a roughly good similarity weights for the CLIP2CLIP model (using OVEN-Wiki's training data). Then we took this similarity weights to initialize the CLIP2CLIP model and fine-tune all parameters on OVEN-Wiki's training set, under the same contrastive learning objective. Different from other models, given that this model has most of its parameters pre-trained, we realized that it works the best to early stop the model. As a result, we only fine-tune this model for 2k steps, with an initial learning rate of 1e-4, and a square root LR decay schedule with final learning rate of 1e-6.

2.3. PaLI Model

As aforementioned, we have evaluated two variants of PaLI models, the model with 3B total parameters (*i.e.*, PaLI-3B) and the model with 17B parameters (*i.e.*, PaLI-17B). The PaLI-17B model reuses 13B parameter from the mT5-XXL [10] and 4B parameters from the ViT-e [13], which were pre-trained Web Text and JFT-3B datasets, and then jointly trained on the WebLI [2] dataset with 10B image and text pairs, under a variety of pre-training objectives, including object recognition, split captioning, visual question answering, etc. Similarly, the PaLI-3B model reuses 1B parameters from mT5-Large [10], and 1.8B parameters from the ViT-G [13], under the same pre-training recipe. To fine-tune PaLI on our dataset, we finetue the pre-trained PaLI model using its Visual QA interface, and inject the OVEN text queries into the PaLI's VQA prompt. As a concrete example, we convert a original query of `what species is the animal in the image?` into the format of `Answer in en: what species is the animal in the image?`, as input to the PaLI model. The objective of fine-tuning process is then to maximize the likelihood of answer generation, same as its standard VQA fine-tuning practices. Similarly, we employ the Adafactor optimizer to optimize the fine-tuning, with a total of 2K fine-tuning steps, with a warmup of 1K steps and linear LR decay schedule.

3. Additional Review on Related Works

In this section, we continue the review of related works omitted in the main text.

Entity linking (EL) is the task of grounding entity mentions in the text by linking them to entries in a given knowledge base. Supervised EL [7] has demonstrated its strong performance when all entities are in-distribution during the evaluation. Because KB is updating all the time, recent works [1, 3, 5, 6, 14] focus on a more realistic setting where entity linking needs to be achieved in the zero-shot, with a large portion of entities (to be evaluated) completely unseen during the training. OVEN is a visual analog of zero-shot EL, and targets at developing generalizable models that recognize entities unseen in the training. Among all EL literature, visually assisted EL [15] is most relevant to this work, whose goal is to use the associated image of text to improve the precision of text EL. OVEN is different as its text queries do not mention the name of the entities, which put visual understanding and reasoning into the central position.

4. Additional Experimental Results

In this section, we provide the validation performances on the proposed OVEN-Wiki (see Table 1), as well as complete results of ablation studies (see Figure 5 and Figure 4).

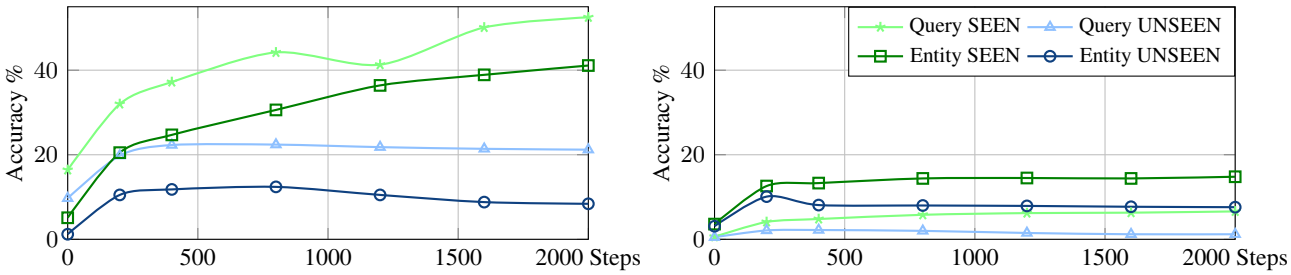
References

[1] Jan A Botha, Zifei Shan, and Dan Gillick. Entity linking in 100 languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7833–7845, 2020. 4

[2] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. Pali: A jointly-scaled multilingual language-image model. *arXiv preprint arXiv:2209.06794*, 2022. 4

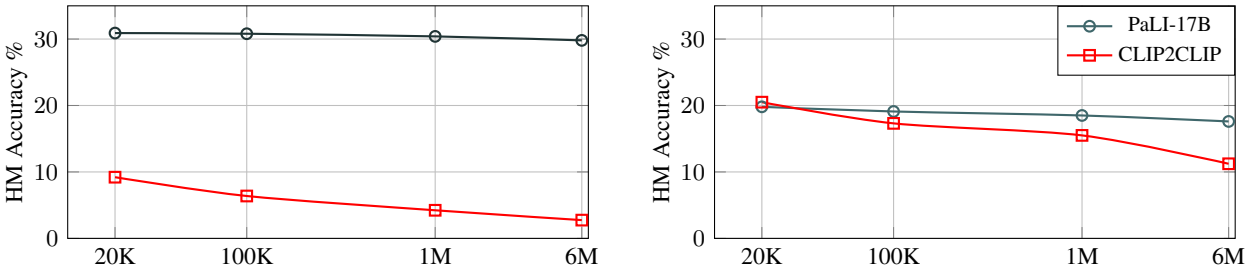
	# Params	Entity Split$_{(\text{Dev})}$ SEEN	UNSEEN	HM	Query Split$_{(\text{Dev})}$ SEEN	UNSEEN	HM	Overall$_{(\text{Dev})}$ HM
Dual Encoders:								
● CLIP$_{\text{ViTL14}}$	0.42B	5.4	5.3	5.4	0.8	1.4	1.0	1.7
● CLIP Fusion$_{\text{ViTL14}}$	0.88B	32.7	4.3	7.7	33.4	2.2	4.2	5.4
● CLIP2CLIP$_{\text{ViTL14}}$	0.86B	12.6	10.1	11.2	4.1	2.1	2.8	4.4
Encoder Decoder:								
◆ PaLI-3B	3B	21.6	6.6	10.1	33.2	14.7	20.4	13.5
◆ PaLI-17B	17B	30.6	12.4	17.6	44.2	22.4	29.8	22.1

Table 1: Comparison between the fine-tuned models on the OVEN-Wiki **validation** set.



(a) PaLI-17B (b) CLIP2CLIP$_{\text{ViTL14}}$

Figure 4: **Fine-tuning PaLI or CLIP2CLIP for large # of steps** increases the SEEN entity accuracy but hurts the UNSEEN entity accuracy.



(a) Query Split (b) Entity Split

Figure 5: **Impact of # Wikipedia Candidates on PaLI and CLIP2CLIP.** Increasing the size of Wikipedia makes the tasks difficult.

[3] Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. Autoregressive entity retrieval. *arXiv preprint arXiv:2010.00904*, 2020. 4

[4] Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. Autoregressive entity retrieval. In *ICLR*, 2021. 1

[5] Nicola De Cao, Ledell Wu, Kashyap Popat, Mikel Artetxe, Naman Goyal, Mikhail Plekhanov, Luke Zettlemoyer, Nicola Cancedda, Sebastian Riedel, and Fabio Petroni. Multilingual autoregressive entity linking. *Transactions of the Association for Computational Linguistics*, 10:274–290, 2022. 4

[6] Lajanugen Logeswaran, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Jacob Devlin, and Honglak Lee. Zero-shot entity linking by reading entity descriptions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3449–3460, 2019. 4

[7] David Milne and Ian H Witten. Learning to link with wikipedia. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 509–518, 2008. 4

[8] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 3, 4

[9] Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*, pages 4596–4604. PMLR, 2018. 4

[10] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mt5: A massively multilingual pre-trained text-to-text

transformer. *arXiv preprint arXiv:2010.11934*, 2020. 4

[11] Kaiyu Yang, Klint Qinami, Li Fei-Fei, Jia Deng, and Olga Russakovsky. Towards fairer datasets: Filtering and balancing the distribution of the people subtree in the imagenet hierarchy. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 547–558, 2020. 1

[12] Shuhei Yokoo, Kohei Ozaki, Edgar Simo-Serra, and Satoshi Iizuka. Two-stage discriminative re-ranking for large-scale landmark retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 1012–1013, 2020. 2

[13] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12104–12113, 2022. 4

[14] Wenzheng Zhang, Wenyue Hua, and Karl Stratos. Entqa: Entity linking as question answering. In *International Conference on Learning Representations*, 2021. 4

[15] Qiushuo Zheng, Hao Wen, Meng Wang, and Guilin Qi. Visual entity linking via multi-modal learning. *Data Intelligence*, 4(1):1–19, 2022. 4