Taxonomy-Aware Evaluation of Vision–Language Models

Supplementary Material

10. Elaborating on Text Similarity Measures

In this section, we elaborate on the different text similarity measures used in this paper and discuss trade-offs across the measures. The methods are ordered roughly in terms of computational overhead, with string-matching methods first (**Exact Match**, **Contained**), then methods based on n-grams (**ROUGE**, **METEOR**), and finally methods that use forward passes through pre-trained large neural networks (**BERTScore**, **SentenceBERT**, **NLI**, **CLIP**).

- **Exact Match (EM)** measures whether a model's predicted output exactly matches the label. This measure is commonly used in, e.g., question-answering systems.
- **Contained**, like **EM**, measures a model's prediction against a gold label by lexical matching. However, this measure is looser, i.e., as long as the prediction contains the label as an exact subsequence, it will be considered correct. Such a measure will have higher recall but lower precision than EM, since it may produce more false positives as well as fewer false negatives; as such, this is particularly susceptible to false positives in predictions which contain the gold label but mean something different in the text context [72]. In the iNaturalist21 setting, Contained will often match higher-level labels (e.g., the species label CHESTNUT-COLLARED SWIFT contains the genus label COLLARED SWIFT which contains the family label SWIFT.)
- **BLEU** [53] is an n-gram overlap measure used originally for evaluating machine translation. It measures how well a predicted string matches a reference (label) string, based on the matched n-grams between the two. **BLEU** is precision-oriented: it captures the proportion of n-gram spans in the predicted string that also appear in the reference string. Predictions that mostly coincide with the reference string with little extra text will score higher. We use **BLEU2** ($n = 2$, i.e., bigrams) with smoothing.
- **ROUGE** [42] is a traditional measure used in automatic summarization. Like **BLEU**, **ROUGE** is based on n-gram overlap, but unlike **BLEU**, **ROUGE** is a recall-oriented measure: if many n-grams contained in the reference string are contained in the predicted string, then the prediction will have a high **ROUGE** score regardless of whether it contains many other irrelevant n-grams. It will thus have a similar pattern of behavior as Contained. We use **ROUGE1**, i.e., measuring unigram recall.
- **METEOR** [5] scores a model's prediction against a gold label based on two factors: first, the unigram precision-recall harmonic mean, i.e., the harmonic (oftentimes weighted) average between the precision and recall of the prediction and label at the unigram level, and an alignment penalty, which intuitively aims to capture how close the ordering of the words in the prediction match the ordering of the words in the label. This alignment penalty is based on the number of consecutive unigram chunks that can be aligned between the prediction and label (where fewer chunks means a lower penalty). It also flexibly matches word stems, paraphrases, or synonyms, if unigrams do not match exactly. This measure captures tends to correlate better with human judgments than other lexical matching measures like **ROUGE**.
- **BERTScore** [82] is a representation-based measure which compares the prediction and label based on semantic similarity. To compute the BERTScore of a prediction against a label, one must first compute the token-level **BERT** representations [21] for each token in the prediction and label. Subsequently, the semantic similarity of each token-pair between prediction and label is computed using cosine similarity. These token-level cosine similarities are then aggregated to compute the precision, recall, and F1. By relying on similarity in contextual token representations, **BERTScore** is better at capturing paraphrases than the above measures based on lexical matching. It is also better at settings in which semantic similarity is an important criterion to judging the prediction and label, but not the exact tokens.
- **SentenceBERT** [58] is also a representation-based measure to compare the prediction and label based on semantic similarity. However, **SentenceBERT** aggregates the representations for a sentence into a single fixed-length representation representing the full sentence. Thus, in comparing the prediction and label, one computes the cosine similarity between the **SentenceBERT** representation for both prediction and label.
- **NLI** [11, 40, 73, 77] uses textual entailment to judge the specificity of a prediction. If the prediction entails the label, then the prediction is more specific than the label. If the label entails the prediction, then the label is more specific. If both entail each other, this suggests the prediction and label are perfect matches. If neither entails the other, then this suggests low hierarchical precision.
- **CLIP text-to-text** [55] compares the prediction to the label by comparing the cosine similarity of the **CLIP** text representation of the predicted string and label string.
- **CLIP image-to-text** [55] compares the prediction to an image sampled from the label category by comparing the cosine similarity of the **CLIP** text representation of the predicted string and the **CLIP** image representation of the image matching that label. Since **CLIP** is explicitly

optimized for matching images and text representations, we hypothesized **CLIP** may be better suited for comparing hierarchical similarity between text (prediction) and image (corresponding to label).

11. Mapping Predictions Onto a Taxonomy

The algorithm we use for mapping predictions onto the taxonomies is given in Alg. 1. It is described in section §6 in the main paper. We use the parameters `k=10, thr_topk=0.0015, thr_top2=0.001, thr_vote=4`.

Both Wikidata and iNaturalist21 taxonomies come with canonical and alternative label names. We do not make use of the alternative non-English Wikidata labels. For the iNaturalist21 dataset, we make use of both the canonical Linnean Latin names, and the English common names. In Alg. 1, we compare predictions both to BIRD and Latin AVES. The text similarity measures use the common (English) name whenever it is available.

For the direct comparison in Alg. 1 we do basic tokenization: we strip punctuation, whitespace, and lowercase. In the case of **Exact Match**, **Contained**, **ROUGE**, and **BLEU** we also stem the words (e.g.. removing the difference between singular and plurals).

12. Visualization of Correlations

The data used to compute the correlations reported in Tab. 1 are visualized in Fig. 8.

13. Prompts Used for the VLMs

The full prompts used for the VLM generations are given in Tab. 7 and Tab. 8. The prompts vary slightly due to differences in how the models are prompted. For iNat21, the question is always: *What species is this?*, while for OVEN the question varies by object type, e.g., *What is the model of this aircraft?*

14. Ranking of VLMs

We plot the ranking of different VLMs in §7.1, the numbers behind the plot are given in Tab. 4 (OVEN) and Tab. 5 (iNaturalist21).

15. Bird Classifier Example

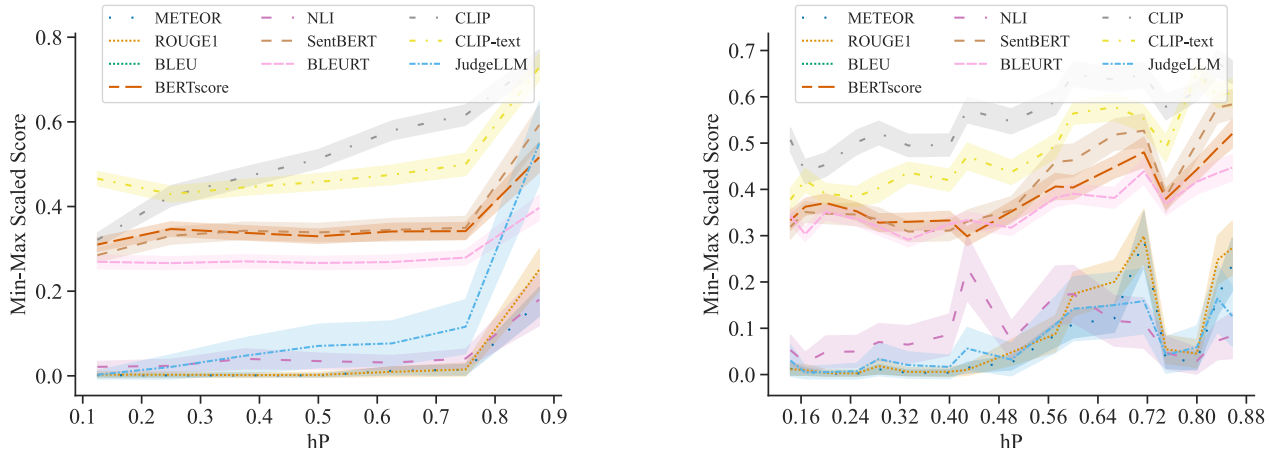
We use ChatGPT to iterate on prompts for classifying birds without giving wrong information as described in §7.2. The model is given a "system prompt" that describes its task. These prompts are given in Tab. 6

16. Example predictions and positions

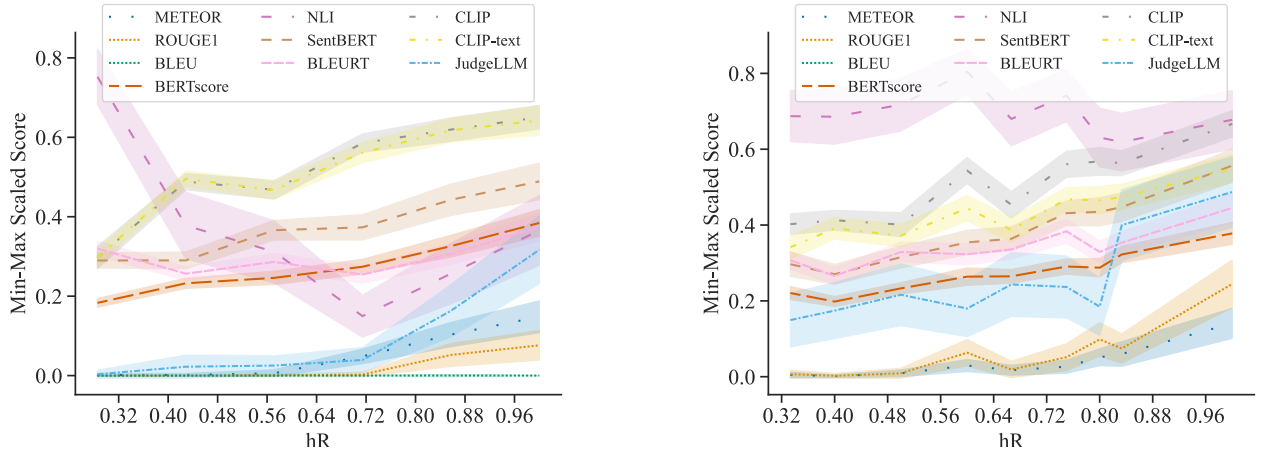
Some example VLM answers, mapped taxonomic predictions, and ground truth reference labels of the images are

Algorithm 1 Mapping predictions onto a taxonomy

```python
def anc(node):
  ancs = [node]
  par = node.parent
  while par:
    ancs.append(par)
    par = par.parent
  return ancs

def n_gram(text, n):
  spl = text.split()
  return [" ".join(spl[i:i+n]) for i in range(len(spl)-n+1)]

def map_tax(pred, T, m, k, thr_topk, thr_top2, thr_vote):
  """
  Map a prediction to the most similar node in a taxonomy.
  *Inputs*
  pred: str, model prediction
  T: object, taxonomy tree relating the nodes
  m: function, similarity measure
  k: int, number of top similar nodes to consider
  thr_topk: float, max difference between top-1 and top-k
  thr_top2: float, max difference between top-1 and top-2
  thr_vote: int, min number of votes for node to be selected
  """
  S = [(m(pred, v.label), v) for v in T]  # similarity
  S.sort(key=lambda x: -x[0])  # sort nodes by similarity
  S, V = zip(*S)  # store sorted similarity S and nodes V
  S_k = softmax(S[:k])  # normalize top-k similarity scores

  # Contains check:
  # return most specific node where pred. contains the label
  cand = None
  for v in V:
    if v.label in pred:
      if cand is None
          or len(anc(v)) > len(anc(cand)):
        cand = v  # store most specific node
    if cand is not None and V.index(v) == k - 1:
      return cand  # We found a hit in top-k
  if cand is not None:
    return cand

  # n-gram check:
  # return most specific node with overlapping n-grams
  for n in (4, 3, 2):
    cand = None
    pred_ngrams = n_gram(pred, n)
    for v in V:
      v_ngrams = n_gram(v.label, n)
      if pred_ngrams.intersect(v_ngrams):
        if cand is None
            or len(anc(v)) > len(anc(cand)):
          cand = v  # store most specific node
      if cand is not None and V.index(v) == k - 1:
        return cand  # We found a hit in top-k
    if cand is not None:
      return cand

  # Voting:
  # return most specific common ancestor in top-k nodes
  # with minimum thr_vote number of occurrences, if multiple
  # such nodes, choose the most frequent one.
  if S_k[0] - S_k[1] < thr_top2
    and S_k[0] - S_k[-1] < thr_topk:
    votes = defaultdict(lambda: defaultdict(int))
    for v in V[:k]:
      for i, node in enumerate(anc(v)):
        votes[i][node] += 1
    for i, counts
        in sorted(votes.items(), key=lambda x: -x[0]):
      node, count = max(counts.items(), key=lambda x: x[1])
      if count > thr_vote: return node

  # Fallback: return most similar node
  return V[0]
```



(a) Similarity measures plotted against **hP**.



(b) Similarity measures plotted against **hR**.

Figure 8. **Text similarity measures vs. taxonomy-aware measures.** iNaturalist21 (left) and Wikidata (right) measures as a function of **hP** (top) and **hR** (bottom). Error bars are standard deviation scaled by $\frac{1}{5}$ to improve readability while allowing for a relative comparison of the measures.

given in §16. The prediction on the taxonomy is obtained using Alg. 1.

17. Hierarchical Labels in Visual Recognition

Hierarchical labels have been studied for different aspects of visual representation learning, including contrastive representation learning [14], weakly-supervised object localization [13], open-set recognition [38, 68], and category discovery [57, 83]. A well-studied use case of hierarchical labels is to inform the learning of recognition models to reduce the 'severity' of mistakes [8, 17, 19, 22, 37, 66], e.g., by directly optimizing for the accuracy-specificity trade-off [19], allowing a model to be accurate at the cost of specificity. Similarly, incorporating knowledge graphs as expert-level human judgment has been shown to be beneficial [48]. Hierarchical recognition has also been explored in an open-world setting [7, 20, 39, 64, 70] in which the goal is to relate novel, unseen categories to the training categories, e.g., by placing unseen categories on the taxonomy [70], or find the closest common taxonomic ancestor between the training data and the out-of-distribution sample [7]Taxonomic information has also been used to speed up annotation [67]. While not relying on explicit taxonomies, a taxonomic structure has also been extracted from text to enrich vision and language datasets to benefit contrastive learning [24]. Computing semantic similarity between concepts or entities in taxonomies and knowledge graphs has been widely studied [43, 59, 60, 85, 86]. Most methods quantify similarity based on the information encoded in graph or taxonomy nodes, often using empirical probabilities to weight graph edges. More recent approaches [2, 85] incorporate corpus statistics alongside hi-

Model	EM	Contained	ROUGE1	METEOR	SentBERT	BERTScore	NLI	hR	hP	hF
Fuyu	23.1	23.1	43.2	29.0	55.8	57.4	76.1	0.660	0.166	0.265
GPT4V	78.9	78.9	81.0	70.4	85.0	87.1	88.6	0.876	0.168	0.282
OLM12B	44.5	44.5	59.3	43.2	67.8	69.1	81.9	0.727	0.145	0.242
QwenVL	50.2	50.2	65.1	48.5	73.2	73.1	86.4	0.804	0.151	0.254
LLaVA	18.2	18.2	25.9	19.5	41.7	50.0	51.3	0.445	0.151	0.225
OLM3B	14.0	14.0	31.9	19.3	46.2	50.8	64.8	0.538	0.144	0.227
QVLChat	34.4	34.4	52.5	35.5	62.7	63.3	80.6	0.707	0.159	0.260
ILmXC2	36.8	36.8	50.6	38.4	61.5	64.1	76.7	0.687	0.161	0.261

Table 4. **VLM evaluation for the OVEN dataset.** Mean similarity measure results using the minimal templates for each model.

Model	EM	Contained	ROUGE1	METEOR	SentBERT	BERTScore	NLI	hR	hP	hF
GPT4V	14.4	14.5	15.7	18.9	47.6	54.6	37.6	0.687	0.336	0.451
LLaVA	0.409	0.409	1.28	5.19	36.9	45.8	20.3	0.630	0.346	0.447
OLM12B	4.07	4.10	5.28	10.9	42.5	50.2	29.2	0.680	0.300	0.416
Fuyu	0.577	0.577	1.39	4.91	36.2	45.5	22.1	0.615	0.316	0.417
QwenVL	5.49	5.56	7.08	11.2	42.3	50.5	33.2	0.672	0.319	0.433
OLM3B	0.882	0.882	1.95	7.69	40.0	47.6	24.1	0.666	0.325	0.437
ILmXC2	1.34	1.34	2.44	7.60	39.2	47.4	22.9	0.654	0.296	0.408
QVLChat	1.15	1.15	2.23	6.32	38.1	46.3	23.2	0.635	0.357	0.457

Table 5. **VLM evaluation for the iNaturalist21 dataset.** Mean similarity measure results using the minimal templates for each model.

erarchical structures. In this work, we rely on unweighted node distances, as we focus on fine-grained classification domains where corpus frequencies are sparse and potentially unreliable.

Approach	Template
hR/**hP** aware	*You are an AI assistant helping to generate effective prompt templates for vision-language models to identify birds in images. The prompt templates should help the model provide accurate species identification that don't contain wrong information, if in doubt we want to back off and ensure the hP value stays as close to 1 as possible. Analyze the previous prompts and their performance metrics (hR: Hierarchical Recall, hP: Hierarchical Precision) to generate an improved template which produces as high scores for both hR and hP as possible. The goal is to make the model we are tuning never return any false information – this is measured by the hP. Then we also want as much correct information as possible, this is the hR. hP=1 is a priority. A simple baseline for this would be to always answer "Bird". But this would always give 0.5 for hR which is as low as we can get, so really try and make this higher, while still aiming for hP=1. Note that the target model is a small Llama 3.2 vision, so it will be sensitive to variations in the prompt. Here's an explanation of the two metrics. Given a taxonomy of entities: * Hierarchical Precision (hP) captures the amount of incorrect information in the prediction, i.e., deviations from the correct path, relative to the extent of the shared, correct, path. * Hierarchical Recall (hR) measures the amount of correct information in the prediction: how much the predicted node's path, to the root, intersects with the correct path. It thus penalizes missing coverage of the target path, especially early deviations. This can be seen as measuring the specificity of the prediction, in particular when the hP is 1. * Both HP and HR achieve the highest score (1.0) when the target and predicted paths coincide. Less-than-perfect scores capture the extent to which the paths deviate as described above. Based on these previous prompt templates and their performance metrics: {prompt_history} Generate a new prompt template that might perform better. Focus on: 1. What worked well in high-performing templates 2. Avoiding patterns from low-performing templates 3. Recall the task is to identify birds without providing false information, while still trying to be specific (A high HP=1 is prioritized, then when it is close to 1 we want to improve HR) Return only the new template text, without any explanations.* And each entry in *prompt_history* is given by *Template: {prompt} Hierarchical Recall (HR): {hr} Hierarchical Precision (HP): {hp}"*
Accuracy aware	*You are an AI assistant helping to generate effective prompt templates for vision-language models to identify birds in images. The prompt templates should help the model provide accurate species identification that don't contain wrong information. Analyze the previous prompts and their performance metric (acc: how often the model's prediction is correct on average over the dataset) to generate an improved template which produces as high an accuracy as possible. Based on these previous prompt templates and their performance metrics: {prompt_history} Generate a new prompt template that might perform better. Focus on: 1. What worked well in high-performing templates 2. Avoiding patterns from low-performing templates 3. Recall the task is to identify birds without providing false information, while still trying to be specific Return only the new template text, without any explanations.* And each entry in *prompt_history* is given by *Template: {prompt} Hierarchical Accuracy: {acc}*.

Table 6. **System prompts for the bird classifier example.** These are the system prompts used to prompt ChatGPT to iterate on the classification prompt.

Model	Template
GPT-4	*{} Answer in the format A: <answer>.*
LLaVA	*{} Answer in the format A: <answer>.*
Fuyu	*Q: {} A:*
ILXC2	*<ImageHere>{} Answer in the format A: <answer>,*
OmniLMM12B	*{} Answer in the format A: <answer>.*
OmniLMM3B	*{} Answer in the format A: <answer>.*
QwenVL	*Q: {} A:*

Table 7. **Barebone prompt templates.** An overview of the prompts used with the various VLMs.

Model	Template
GPT-4	{} *Do not give any extra text. Do not answer in a full sentence. Do not specify your certainty about the answer. Give your best guess if you are not sure. Be as specific as possible. Answer in the format A: <answer>.*
LLaVA	{} *Do not give any extra text. Do not answer in a full sentence. Do not specify your certainty about the answer. Give your best guess if you are not sure. Be as specific as possible. Answer in the format A: <answer>.*
Fuyu	*Q:* {} *A:*
ILXC2	*<ImageHere >*{} *Do not give any extra text. Do not answer in a full sentence. Do not specify your certainty about the answer. Give your best guess if you are not sure. Be as specific as possible. Answer in the format A: <answer>.*
OmniLMM12B	{} *Do not give any extra text. Do not answer in a full sentence. Do not specify your certainty about the answer. Give your best guess if you are not sure. Be as specific as possible. Answer in the format A: <answer>.*
OmniLMM3B	{} *Do not give any extra text. Do not answer in a full sentence. Do not specify your certainty about the answer. Give your best guess if you are not sure. Be as specific as possible. Answer in the format A: <answer>.*
QwenVL	*Q:* {} *A:*

Table 8. **Specific prompt templates.** An overview of the prompts used with the various VLMs with the goal of providing a specific answer.

Model	Reference	VLM Answer	Predicted Node
LLaVA	Strawberry Anemone	*Anemone*	Actiniid Sea Anemones
OmniLMM12B	Velvety Tree Ant	*ant*	Carpenter Ants, Typical Sugar Ants and Allies
GPT-4	Carex pilosa	*Equisetum (horsetail) species*	horsetails
QwenVL	Nuttall's Snapdragon		Animals
QwenVLChat	sycamore maple	*maple*	maples
LLaVA	Canadian wood nettle	*Plant*	dicots
OmniLMM3B	Oysterplant	*Aloe vera*	aloe vera
Fuyu	Cuckoo-pint	*This plant is a banana plant*	gingers, bananas, and allies
OmniLMM3B	Familiar Bluet	*dragonfly*	Skimmers
OmniLMM3B	Texan Crescent	*butterfly*	Brush-footed Butterflies
Fuyu	Yellow-shouldered Slug Moth	*moth*	Butterflies and Moths
InternLmXC2	Meadow Pipit	*sparrow*	New World Sparrows
OmniLMM12B	Chihuahuan Nightsnake	*snake*	Colubrid Snakes
InternLmXC2	fall phlox	*Asteraceae*	sunflowers, daisies, asters, and allies
Fuyu	Crested Pigeon	*This is two pigeons*	Pigeons and Doves
OmniLMM12B	Drummond's rain lily	*tulip*	monocots
QwenVL	Strawberry Poison-dart Frog	*Strawberry poison frog*	Animals
LLaVA	Red-masked Parakeet	*Parrot*	New World and African Parrots
Fuyu	Hutton's Vireo	*bird*	Perching Birds
OmniLMM3B	honewort	*ivy*	dicots
OmniLMM3B	heath wood-rush	*grass*	grasses
QwenVL	Cheetah	*cheetah*	Cheetah
LLaVA	Gray Petaltail	*Dragonfly*	Skimmers
InternLmXC2	Pima Rhatany	*This is a purple flower*	dicots
QwenVLChat	Margined Leatherwing Beetle	*bug*	True Bugs, Hoppers, Aphids, and allies
OmniLMM12B	Brazos rain-lily	*grass*	grasses
OmniLMM12B	The Gem	*Lepidoptera*	Butterflies and Moths
GPT-4	Blue Crane	*Crane*	Pelicans, Herons, Ibises, and Allies
QwenVLChat	Twice-stabbed Stink Bug	*bugs*	True Bugs, Hoppers, Aphids, and allies
QwenVLChat	Nacoleia rhoeoalis	*butterfly*	Brush-footed Butterflies
InternLmXC2	White Bream	*This is a school of fish swimming in the ocean*	Perch-like Fishes
QwenVL	Texan Crescent	*This is a butterfly*	Brush-footed Butterflies
OmniLMM12B	Siberian Iris	*Iris*	Irises
QwenVL	long beech fern	*Polystichum acrostichoides*	Christmas fern
QwenVL	Black-mandibled Toucan	*Toucan*	Toucans
QwenVL	Hazel Grouse	*Ruffed Grouse*	Ruffed Grouse
OmniLMM12B	Hickory Tussock Moth	*caterpillar*	Underwing, Tiger, Tussock, and Allied Moths
QwenVL	Goatsbeard	*Actaea pachypoda*	doll's eyes
Fuyu	Definite Tussock Moth	*This caterpillar is a species of moth*	Butterflies and Moths

Table 9. **Examples of model predictions on the iNaturalist21 dataset.** 'Model' refers to the model used to answer the question given an image. 'VLM Answer' shows the output from the model, 'Reference' is the true label, and 'Predicted Node' is the node label we get from mapping onto the taxonomy using Alg. 1.