A Generative Approach for Wikipedia-Scale Visual Entity Recognition

Supplementary Material

6. Implementation Details

We use the Large version of GIT [45] with a pretrained visual encoder and a decoder randomly initialized. The visual encoder is pre-trained with GIT trained for captioning on WebLI dataset [5, 45].

6.1. Entity-based pre-training

We use batch size of 4096, learning rate of $1e-5$ for the visual encoder and 1e^{-4} for the decoder, label smoothing of 0.3 and no weight decay. We use standard inception crop data augmentation for the images. By default and unless specified otherwise, we use code length $L = 4$ (see Fig. 5). Note that we only evaluate codes with $L > 1$ on OVEN, as the only way to ensure unique codes with $L = 1$ is to set $V = |\mathcal{E}|$. This is equivalent to the classification scenario and is not feasible for the million-scale label-space of OVEN. We evaluate $L = 1$ in Sec. 4.5 for datasets with a smaller label-space of 1k entities: namely ImageNet-LT [24] and Webvision [22]. Unless specified otherwise our models for the main results (i.e. in Sec. 4.2 and Sec. 4.3) are trained on Entity-WebLI with 55M images ($k = 100$) during 600k steps while models for ablations are trained on Entity-WebLI with 27M images ($k = 20$) for 200k steps.

Preventing data leakage. Webli is already deduplicated against the train, val, and test splits of 68 common vision/vision-language datasets (see PaLI paper [5]). To be sure, in our paper, we further removed pretraining images with a cosine similarity (with CLIP-L/14 visual features) above 0.95 with any of the OVEN images. We chose a 0.95 conservative threshold by looking at some examples: similarity 0.95 corresponds to conceptually similar images but clearly not duplicates (see Fig 8).

6.2. Finetuning on OVEN train set

We finetune models on OVEN training set for 30,000 steps with a batch size of 256 and a learning rate of 1e^{-7}. Label smoothing is set at 0.1. Note that the finetuning schedule is relatively short (30,000 steps) because we observe that long finetuning (or equivalently, using a large learning rate) causes the model to forget about the unseen categories.

	Zero-shot			Finetuned on OVEN		
Method	HM	Seen	Unseen	HM	Seen	Unseen
PaLI-17B [5]	1.8	4.4	1.2	16.0	28.3	11.2
GiT-Large [45]	1.7	4.1	1.2	7.0	17.6	4.3

Table 5. **Transferring captioning models to OVEN.** We report the harmonic mean (HM) of top-1 accuracy on the seen and unseen test splits for two captioning models: PALI-17B [5] and GiT-Large [45]. Numbers from GiT-Large are run by us. Note that GiT-Large has 42× less parameters thank PALI-17B.

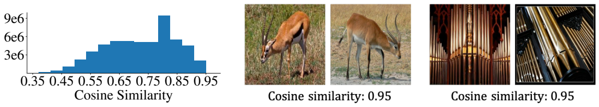


Figure 8. **Filtering out pretraining data too similar to OVEN test/val.**

Algorithm 1 GER-ALD codes.

```
Data: Code length L, Text tokenizer $\Phi(.)$, Entities $\mathcal{E}$
Result: $\mathcal{C} = \{c_e\}_{e \in \mathcal{E}}$
for $v \in [1, V]$ do
    $f_v = \frac{1}{\sum_{e \in \mathcal{E}} L_e} \sum_{e \in \mathcal{E}} \sum_{y_i^e \in \Phi(t_e)} \mathbb{1}_{y_i^e = v}$
end
$\mathcal{C} \leftarrow \varnothing$
for $e \in \mathcal{E}$ do
    sort $\Phi(t_e)$ by decreasing frequencies:
    $\{y_{s_i}^e\}_{i \in [1, L_e]}$ such that $f_{y_{s_i}^e} \leq f_{y_{s_{i+1}}^e}$
    for $i \in [1, L-1]$ do
        Set $c_i^e \leftarrow y_{s_i}^e$
    end
    $j = 0$
    while $c_e \in \mathcal{C}$ and $j \leq (L_e - L)$ do
        $c_L^e \leftarrow y_{s_{L+j}}^e$
        $j = j + 1$
    end
    while $c_e \in \mathcal{C}$ do
        $c_L^e \leftarrow v' \sim [1, V]$
    end
    $\mathcal{C} \leftarrow \{c_e\} \bigcup \mathcal{C}$
end
```

6.3. Training on ImageNet-LT and Webvision

We train the model on ImageNet-LT and Webvision datasets with the batch size of 512 and a learning rate of 1e^{-4} for both encoder and decoder. We do not use any label smoothing but apply a dropout of 0.1. We use $L = 2$ for these experiments with GER-ALD because unlike very large label-space this is enough not to resort to random tokens when ensuring that codes are unambious.

6.4. Implementation details about the baselines

Dual encoder with CLIP-L/14. We use a learning rate of 3e^{-7}, a batch size of 4096 and we train for $200,000$ steps since training for longer deteriorates the performance. Dur-

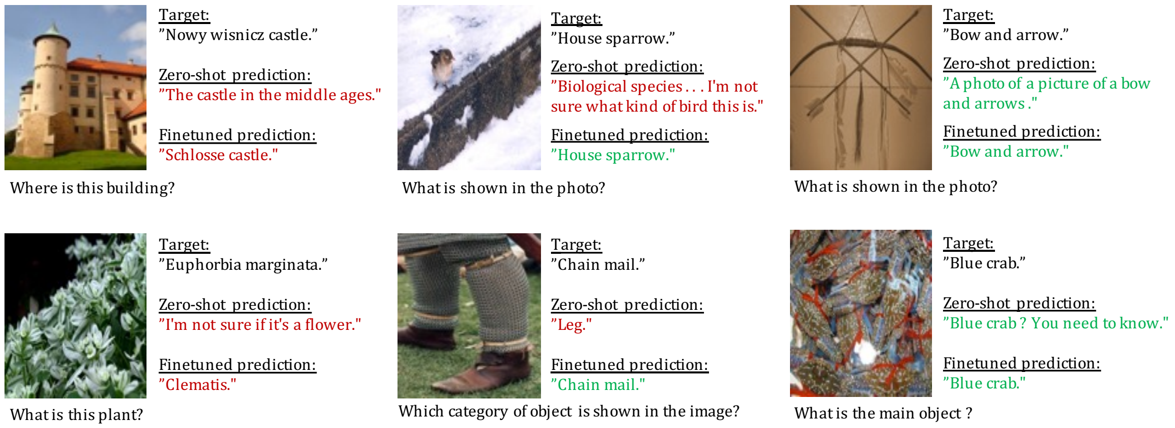


Figure 9. **Zero-shot versus finetuned captioning models predictions.** We qualitatively compare the predictions of the captioning GiT-Large model when evaluated on OVEN in a zero-shot manner or after finetuning on OVEN train set.

ing finetuning on OVEN training set, we find it important to still include some pretraining data: we randomly sample, with a probability of 90%, elements from the pretraining dataset. Otherwise, when finetuning solely on OVEN, the model becomes too specialized for the seen categories and is not capable of discriminating between all the negative entities. Alternatives could be to freeze some layers of the network during finetuning to prevent catastrophic forgetting.

GER-ATOMIC. We benchmark different values for the choice of L: $\{2, 4, 8\}$ and V: $\{512, 4096, 32768\}$ when using atomic codes. Our default is to use $L = 2$ and $V = 4096$. Note that this corresponds to more than 16M possible different codes and we use only a subset of 6M of those unique codes.

GER-HKC. For HKC, we first represent each Wikipedia entity with a text embedding using the sentence-t5 [28] encoder. We have experimented with different ways of creating such embeddings, such as creating text embeddings from the Wikipedia titles, Wikipedia article summaries, and Wikipedia title and article summaries combined together. We have observed that Wikipedia title and article summaries combined together produces the best embeddings. We have tried different values of k: $\{10, 100, 1000, 4096, 8142\}$, and found that $k = 4096$ achieves the best performance in the validation set.

7. More Experimental Results

7.1. Failure case analyses

We show in Fig. 10 that our method works well across different entity types. In many cases, our codes for animals (*e.g. 'Glaucous winged gull'*), persons (*e.g. 'List of celebrities who own wineries and vineyards'*) or organizations (*e.g. Gladney Center for Adoption'*) are interpretable, as shown in the qualitative examples in Fig 11 and Fig 12. We see failure cases when semantics is difficult to infer from entity name alone. This is often the case for scientific denomination of species as show in Fig. 10. In future work, using external tools or Wikipedia page content could improve results in such cases.

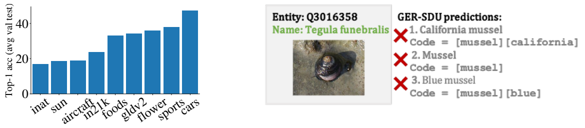
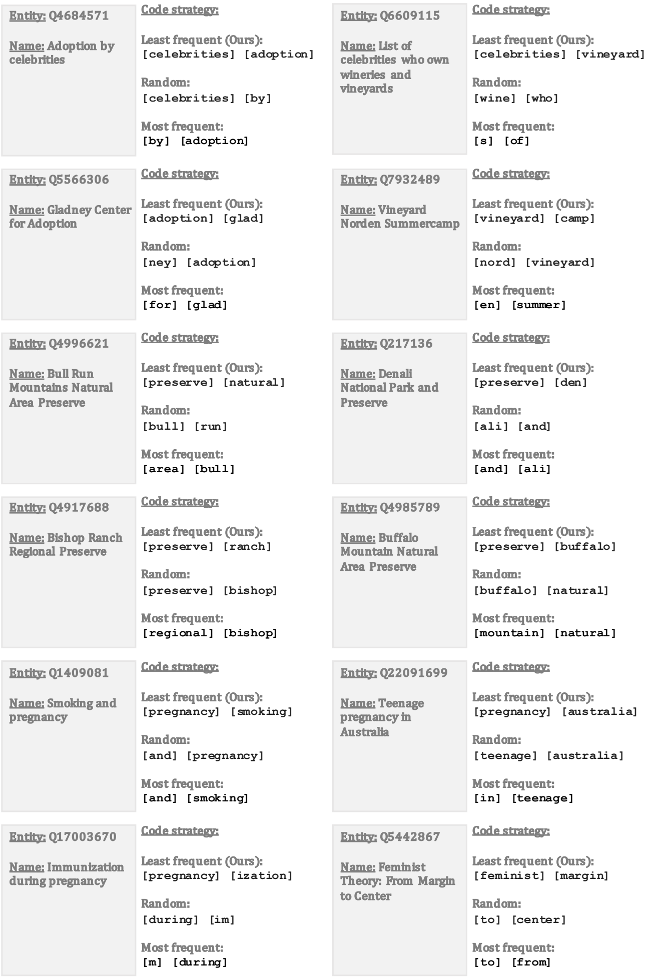


Figure 10. **(left): Accuracy per sub-task in OVEN. (right): A failure case example in iNaturalist ('inat') sub-task.**

7.2. Zero-shot OVEN with captioning models

Quantitative evaluation. In Table 5, we transfer two captioning models, namely PALI-17B and GiT-Large [45], both pre-trained on WebLI [5] to the OVEN task. We observe in Table 5 that these models transfer poorly in a zero-shot manner. This can be explained by the major discrepancy between the pre-training captioning task (*i.e*. describing an image with a caption) and the target entity recognition task.

Visual examples. We show some visual examples of predictions from the validation test between the zero-shot and finetuned GiT-Large models in Figure 9 where we clearly see the difference of output between the zero-shot and finetuned GiT-Large models. In the left column, we show two examples where both zero-shot and finetuned models fail. However, even though the finetuned model fails to find the correct category of castle or plant, it still tries to output a fine-grained category of castle or plant. This is not the case



Entity: Q4684571
Name: Adoption by celebrities
Code strategy:
Least frequent (Ours):
[celebrities] [adoption]
Random:
[celebrities] [by]
Most frequent:
[by] [adoption]

Entity: Q6609115
Name: List of celebrities who own wineries and vineyards
Code strategy:
Least frequent (Ours):
[celebrities] [vineyard]
Random:
[wine] [who]
Most frequent:
[s] [of]

Entity: Q5566306
Name: Gladney Center for Adoption
Code strategy:
Least frequent (Ours):
[adoption] [glad]
Random:
[ney] [adoption]
Most frequent:
[for] [glad]

Entity: Q7932489
Name: Vineyard Norden Summercamp
Code strategy:
Least frequent (Ours):
[vineyard] [camp]
Random:
[nord] [vineyard]
Most frequent:
[en] [summer]

Entity: Q4996621
Name: Bull Run Mountains Natural Area Preserve
Code strategy:
Least frequent (Ours):
[preserve] [natural]
Random:
[bull] [run]
Most frequent:
[area] [bull]

Entity: Q217136
Name: Denali National Park and Preserve
Code strategy:
Least frequent (Ours):
[preserve] [den]
Random:
[ali] [and]
Most frequent:
[and] [ali]

Entity: Q4917688
Name: Bishop Ranch Regional Preserve
Code strategy:
Least frequent (Ours):
[preserve] [ranch]
Random:
[preserve] [bishop]
Most frequent:
[regional] [bishop]

Entity: Q4985789
Name: Buffalo Mountain Natural Area Preserve
Code strategy:
Least frequent (Ours):
[preserve] [buffalo]
Random:
[buffalo] [natural]
Most frequent:
[mountain] [natural]

Entity: Q1409081
Name: Smoking and pregnancy
Code strategy:
Least frequent (Ours):
[pregnancy] [smoking]
Random:
[and] [pregnancy]
Most frequent:
[and] [smoking]

Entity: Q22091699
Name: Teenage pregnancy in Australia
Code strategy:
Least frequent (Ours):
[pregnancy] [australia]
Random:
[teenage] [australia]
Most frequent:
[in] [teenage]

Entity: Q17003670
Name: Immunization during pregnancy
Code strategy:
Least frequent (Ours):
[pregnancy] [ization]
Random:
[during] [im]
Most frequent:
[m] [during]

Entity: Q5442867
Name: Feminist Theory: From Margin to Center
Code strategy:
Least frequent (Ours):
[feminist] [margin]
Random:
[to] [center]
Most frequent:
[to] [from]

Figure 11. **Token selection strategies in GER-ALD.** We qualitatively compare different alternative token selection strategies for GER-ALD: most frequent token or random token selection. We use $L = 2$ for this qualitative evaluation since this is easier to visually interpret that $L = 4$, however the trends are consistent. Quantitative evaluation is in the Table 3 of the main paper.

of the zero-shot model which gives a generic description of the entity, for example "The castle in the middle ages.". In the middle column, we show examples where zero-shot fails but the finetuned model finds the correct category. Finally, in the last column we show cases where the zero-shot model succeeds, but even when it does we observe that the generated caption is cluttered (for example with "a photo of a picture") while the finetuned model directly outputs the entity name.

Overall, the observation that models pre-trained from WebLi captions do not generalize well to OVEN entity recognition motivated us to create our entity-based pretraining described in Section 3.3.

7.3. Entities with long names

In Figure 12, we show more visual examples of GER-ALD and GER-CAPTION predictions for entities with long names.

7.4. Different token selection strategies

In Figure 11, we show visual examples of codes generated with alternatives of selecting the least frequent token in GER-ALD. We compare with selecting instead the most frequent token and with selecting a random token in the entity name. Quantitative evaluation is in Table 3 of the main paper. In Figure 11, we observe that codes generated with least frequent token strategy are the most semantically structured. Indeed, in this case the entities "Adoption by celebrities" and "List of celebrities who own wineries and vineyards." share a common token (the token corresponding to "celebrities") while there is no intersection of token between those two entities for the most frequent or the random strategies. We observe the same effect across several group of entities that we intuitively expect to have shared tokens, for example with "smoking and pregnancy", "teenage pregnancy in Australia" and "Immunization during pregragncy", or with "Denall National Park and Preserve" and "Bishop Ranch Regional Preserve".

7.5. Numbers corresponding to Fig. 3 main paper

We report in Table 6 the numbers corresponding to the experiments shown in Figure 3 of the main paper.

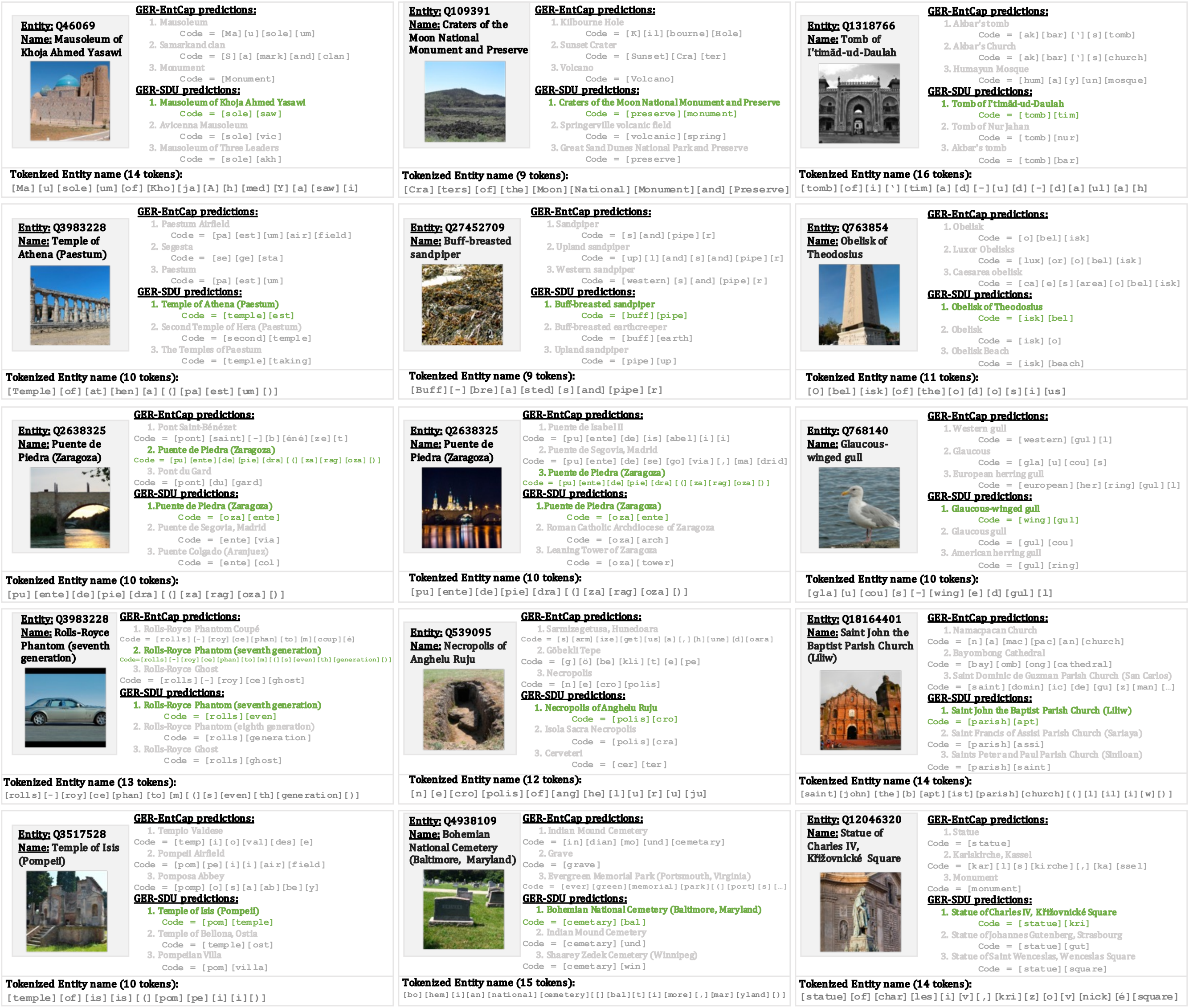

Entity: Q46069
Name: Mausoleum of Khoja Ahmed Yasawi
GER-EntCap predictions:
1. Mausoleum
Code = [Ma][u][sole][um]
2. Samarkand clan
Code = [S][a][mark][and][clan]
3. Monument
Code = [Monument]
GER-SDU predictions:
1. Mausoleum of Khoja Ahmed Yasawi
Code = [sole][saw]
2. Avicenna Mausoleum
Code = [sole][vic]
3. Mausoleum of Three Leaders
Code = [sole][akh]
Tokenized Entity name (14 tokens):
[Ma][u][sole][um][of][Kho][ja][A][h][med][Y][a][saw][i]
Entity: Q109391
Name: Craters of the Moon National Monument and Preserve
GER-EntCap predictions:
1. Kilbourne Hole
Code = [K][il][bourne][Hole]
2. Sunset Crater
Code = [Sunset][Cra][ter]
3. Volcano
Code = [Volcano]
GER-SDU predictions:
1. Craters of the Moon National Monument and Preserve
Code = [preserve][monument]
2. Springerville volcanic field
Code = [volcanic][spring]
3. Great Sand Dunes National Park and Preserve
Code = [preserve]
Tokenized Entity name (9 tokens):
[Cra][ters][of][the][Moon][National][Monument][and][Preserve]
Entity: Q1318766
Name: Tomb of I'timād-ud-Daulah
GER-EntCap predictions:
1. Akbar's tomb
Code = [ak][bar][`][s][tomb]
2. Akbar's Church
Code = [ak][bar][`][s][church]
3. Humayun Mosque
Code = [hum][a][y][un][mosque]
GER-SDU predictions:
1. Tomb of I'timād-ud-Daulah
Code = [tomb][tim]
2. Tomb of Nur Jahan
Code = [tomb][nur]
3. Akbar's tomb
Code = [tomb][bar]
Tokenized Entity name (16 tokens):
[tomb][of][i][`][tim][a][d][-][u][d][-][d][a][ul][a][h]
Entity: Q3983228
Name: Temple of Athena (Paestum)
GER-EntCap predictions:
1. Paestum Airfield
Code = [pa][est][um][air][field]
2. Segesta
Code = [se][ge][sta]
3. Paestum
Code = [pa][est][um]
GER-SDU predictions:
1. Temple of Athena (Paestum)
Code = [temple][est]
2. Second Temple of Hera (Paestum)
Code = [second][temple]
3. The Temples of Paestum
Code = [temple][taking]
Tokenized Entity name (10 tokens):
[Temple][of][at][hen][a][(][pa][est][um][)]
Entity: Q27452709
Name: Buff-breasted sandpiper
GER-EntCap predictions:
1. Sandpiper
Code = [s][and][pipe][r]
2. Upland sandpiper
Code = [up][l][and][s][and][pipe][r]
3. Western sandpiper
Code = [western][s][and][pipe][r]
GER-SDU predictions:
1. Buff-breasted sandpiper
Code = [buff][pipe]
2. Buff-breasted earthcreeper
Code = [buff][earth]
3. Upland sandpiper
Code = [pipe][up]
Tokenized Entity name (9 tokens):
[Buff][-][bre][a][sted][s][and][pipe][r]
Entity: Q763854
Name: Obelisk of Theodosius
GER-EntCap predictions:
1. Obelisk
Code = [o][bel][isk]
2. Luxor Obelisks
Code = [lux][or][o][bel][isk]
3. Caesarea obelisk
Code = [ca][e][s][area][o][bel][isk]
GER-SDU predictions:
1. Obelisk of Theodosius
Code = [isk][bel]
2. Obelisk
Code = [isk][o]
3. Obelisk Beach
Code = [isk][beach]
Tokenized Entity name (11 tokens):
[O][bel][isk][of][the][o][d][o][s][i][us]
Entity: Q2638325
Name: Puente de Piedra (Zaragoza)
GER-EntCap predictions:
1. Pont Saint-Bénézet
Code = [pont][saint][-][b][éné][ze][t]
2. Puente de Piedra (Zaragoza)
Code = [pu][ente][de][pie][dra][(][za][rag][oza][)]
3. Pont du Gard
Code = [pont][du][gard]
GER-SDU predictions:
1.Puente de Piedra (Zaragoza)
Code = [oza][ente]
2. Puente de Segovia, Madrid
Code = [ente][via]
3. Puente Colgado (Aranjuez)
Code = [ente][col]
Tokenized Entity name (10 tokens):
[pu][ente][de][pie][dra][(][za][rag][oza][)]
Entity: Q2638325
Name: Puente de Piedra (Zaragoza)
GER-EntCap predictions:
1. Puente de Isabel II
Code = [pu][ente][de][is][abel][i][i]
2. Puente de Segovia, Madrid
Code = [pu][ente][de][se][go][via][,][ma][drid]
3. Puente de Piedra (Zaragoza)
Code = [pu][ente][de][pie][dra][(][za][rag][oza][)]
GER-SDU predictions:
1.Puente de Piedra (Zaragoza)
Code = [oza][ente]
2. Roman Catholic Archdiocese of Zaragoza
Code = [oza][arch]
3. Leaning Tower of Zaragoza
Code = [oza][tower]
Tokenized Entity name (10 tokens):
[pu][ente][de][pie][dra][(][za][rag][oza][)]
Entity: Q768140
Name: Glaucous-winged gull
GER-EntCap predictions:
1. Western gull
Code = [western][gul][l]
2. Glaucous
Code = [gla][u][cou][s]
3. European herring gull
Code = [european][her][ring][gul][l]
GER-SDU predictions:
1. Glaucous-winged gull
Code = [wing][gul]
2. Glaucous gull
Code = [gul][cou]
3. American herring gull
Code = [gul][ring]
Tokenized Entity name (10 tokens):
[gla][u][cou][s][-][wing][e][d][gul][l]
Entity: Q3983228
Name: Rolls-Royce Phantom (seventh generation)
GER-EntCap predictions:
1. Rolls-Royce Phantom Coupé
Code = [rolls][-][roy][ce][phan][to][m][coup][é]
2. Rolls-Royce Phantom (seventh generation)
Code=[rolls][-][roy][ce][phan][to][m][(][s][even][th][generation][)]
3. Rolls-Royce Ghost
Code = [rolls][-][roy][ce][ghost]
GER-SDU predictions:
1. Rolls-Royce Phantom (seventh generation)
Code = [rolls][even]
2. Rolls-Royce Phantom (eighth generation)
Code = [rolls][generation]
3. Rolls-Royce Ghost
Code = [rolls][ghost]
Tokenized Entity name (13 tokens):
[rolls][-][roy][ce][phan][to][m][(][s][even][th][generation][)]
Entity: Q539095
Name: Necropolis of Anghelu Ruju
GER-EntCap predictions:
1. Sarmizegetusa, Hunedoara
Code = [s][arm][ize][get][us][a][,][h][une][d][oara]
2. Göbekli Tepe
Code = [g][ö][be][kli][t][e][pe]
3. Necropolis
Code = [n][e][cro][polis]
GER-SDU predictions:
1. Necropolis of Anghelu Ruju
Code = [polis][cro]
2. Isola Sacra Necropolis
Code = [polis][cra]
3. Cerveteri
Code = [cer][ter]
Tokenized Entity name (12 tokens):
[n][e][cro][polis][of][ang][he][l][u][r][u][ju]
Entity: Q18164401
Name: Saint John the Baptist Parish Church (Liliw)
GER-EntCap predictions:
1. Namacpacan Church
Code = [n][a][mac][pac][an][church]
2. Bayombong Cathedral
Code = [bay][omb][ong][cathedral]
3. Saint Dominic de Guzman Parish Church (San Carlos)
Code = [saint][domin][ic][de][gu][z][man][...]
GER-SDU predictions:
1. Saint John the Baptist Parish Church (Liliw)
Code = [parish][apt]
2. Saint Francis of Assisi Parish Church (Sariaya)
Code = [parish][assi]
3. Saints Peter and Paul Parish Church (Siniloan)
Code = [parish][saint]
Tokenized Entity name (14 tokens):
[saint][john][the][b][apt][ist][parish][church][(][l][il][i][w][)]
Entity: Q3517528
Name: Temple of Isis (Pompeii)
GER-EntCap predictions:
1. Tempio Valdese
Code = [temp][i][o][val][des][e]
2. Pompeii Airfield
Code = [pom][pe][i][i][air][field]
3. Pomposa Abbey
Code = [pomp][o][s][a][ab][be][y]
GER-SDU predictions:
1. Temple of Isis (Pompeii)
Code = [pom][temple]
2. Temple of Bellona, Ostia
Code = [temple][ost]
3. Pompeian Villa
Code = [pom][villa]
Tokenized Entity name (10 tokens):
[temple][of][is][is][(][pom][pe][i][i][)]
Entity: Q4938109
Name: Bohemian National Cemetery (Baltimore, Maryland)
GER-EntCap predictions:
1. Indian Mound Cemetery
Code = [in][dian][mo][und][cemetary]
2. Grave
Code = [grave]
3. Evergreen Memorial Park (Portsmouth, Virginia)
Code = [ever][green][memorial][park][(][port][s][...]
GER-SDU predictions:
1. Bohemian National Cemetery (Baltimore, Maryland)
Code = [cemetary][bal]
2. Indian Mound Cemetery
Code = [cemetary][und]
3. Shaarey Zedek Cemetery (Winnipeg)
Code = [cemetary][win]
Tokenized Entity name (15 tokens):
[bo][hem][i][an][national][cemetery][(][bal][t][i][more][,][mar][yland][)]
Entity: Q12046320
Name: Statue of Charles IV, Křižovnické Square
GER-EntCap predictions:
1. Statue
Code = [statue]
2. Karlskirche, Kassel
Code = [kar][l][s][kirche][,][ka][ssel]
3. Monument
Code = [monument]
GER-SDU predictions:
1. Statue of Charles IV, Křižovnické Square
Code = [statue][kri]
2. Statue of Johannes Gutenberg, Strasbourg
Code = [statue][gut]
3. Statue of Saint Wenceslas, Wenceslas Square
Code = [statue][square]
Tokenized Entity name (14 tokens):
[statue][of][char][les][i][v][,][kri][z][o][v][nick][é][square]

Figure 12. **Qualitative study of GER-ALD versus GER-CAPTION.** Visual examples of predictions for long entity name from 9 to 16 tokens. For these visualizations with GER-ALD, we use SentencePiece tokenizer [20] and $L = 2$ in this evaluation since this leads to more visually interpretable codes. Tokens are symbolized between brackets. We report the top-3 predictions for GER-ALD and for GER-CAPTION codes, and color in green the correct predictions. We observe that GER-ALD codes are easier to predict as they contain less clutter than GER-CAPTION codes. Interestingly, we see that with GER-ALD the top-3 predictions usually share a common token which re-group different semantically close entities.

Pretraining size (M)	10.6	14.7	26.6	40.3	54.9
GER-ATOMIC	0.9	6.8	11.4	13.8	15.3
GER-ALD	10.2	11.7	14.4	16.1	17.5
Relative Δ (%)	1029	71	26	17	14

Architecture size (M)	114	179	397
GER-ATOMIC	0.7	5.4	11.4
GER-ALD	5.6	9.5	14.4
Relative Δ (%)	648	77	26

# entities (M)	0.02	0.03	0.12	1.00	6.08
GER-ATOMIC	34.6	34.0	29.7	21.5	11.4
GER-ALD	33.5	33.5	30.4	23.6	14.4
Relative Δ (%)	-3.1	-1.6	2.2	9.6	26.4

Table 6. **Semantically-structured (GER-ALD) versus unstructured (GER-ATOMIC) codes.** We report the numbers corresponding to Figure 3 of the main paper. The pretraining dataset sizes of 10.6M, 14.7M, 26.6M, 40.3M and 54.9M correspond respectively to setting k to 2, 5, 20, 50 and 100. The architecture sizes with 114M, 179M and 397M parameters correspond respectively to variant Small, Base and Large of the model. The label space sizes with 20549, 30549, 120549, 1000549 and 6084491 different entities correspond respectively to having 0, 10k, 100k, 1M and 6M entities acting as distractors.