Supplemental Material to "Benchmark for Compositional Text-to-Image Synthesis"

1 Appendix

In the following we provide additional training details (Section 1.1), details on our human evaluation (Section 1.2), additional qualitative examples (Section 1.3) and in-depth statistics for our new compositional splits (Section 1.4).

1.1 Training Details

The models (DMGAN, ControlGAN, DFGAN) that we evaluated on the proposed benchmark follow the default hyperparameters used in their respective codebases: `https://github.com/MinfengZhu/DMGAN`, `https://github.com/mrlibw/ControlGAN`, `https://github.com/tobran/DF-GAN`. With regards to CLIP-R-Precision, we take the pretrained RN101 model from `https://github.com/openai/CLIP` and finetune the model on the CUB and Oxford-Flowers datasets. Since CUB and Oxford-Flowers datasets are relatively small in size compared to the dataset that CLIP was pretrained on, we only finetune the final few layers. The exact layers we finetune are ["visual.layer4", "visual.attnpool", "transformer.resblocks.11", "ln_final", "text_projection", "logit_scale"]. We finetune the model using AdamW optimizer with a learning rate of 0.0005 and batch size of 256 for 100 epochs using the same contrastive loss used for pretraining the original CLIP model. The finetuned CLIP models and the code for computing the CLIP-R-Precision will be publically available. The experiments in this paper were run using NVIDIA Tesla P100 16G GPUs and Titan Xp 12G GPUs.

1.2 Human Evaluation on AMT

We conducted human evaluation on Amazon Mechanical Turk (AMT). We had two types of tasks: perceptual evaluation and correctness evaluation. Figure 1 shows the interface for the perceptual evaluation (identical for both C-CUB and C-Flowers datasets). Figures 2 and 3 show the interfaces for the correctness evaluation for C-CUB and C-Flowers, respectively (mostly identical except for a few details). We paid $\$0.1$ for each perceptual task and $\$0.2$ for each correctness task, $\$2,400$ in total, on average estimated $\$5$ per hour. One caveat is that time spent on the task varied significantly by worker: some were extremely fast, *e.g.* 20 seconds on average, while others took much longer, *e.g.* 100 seconds on average, with no visible gap in quality.

Submitted to the 35th Conference on Neural Information Processing Systems (NeurIPS 2021) Track on Datasets and Benchmarks. Do not distribute.

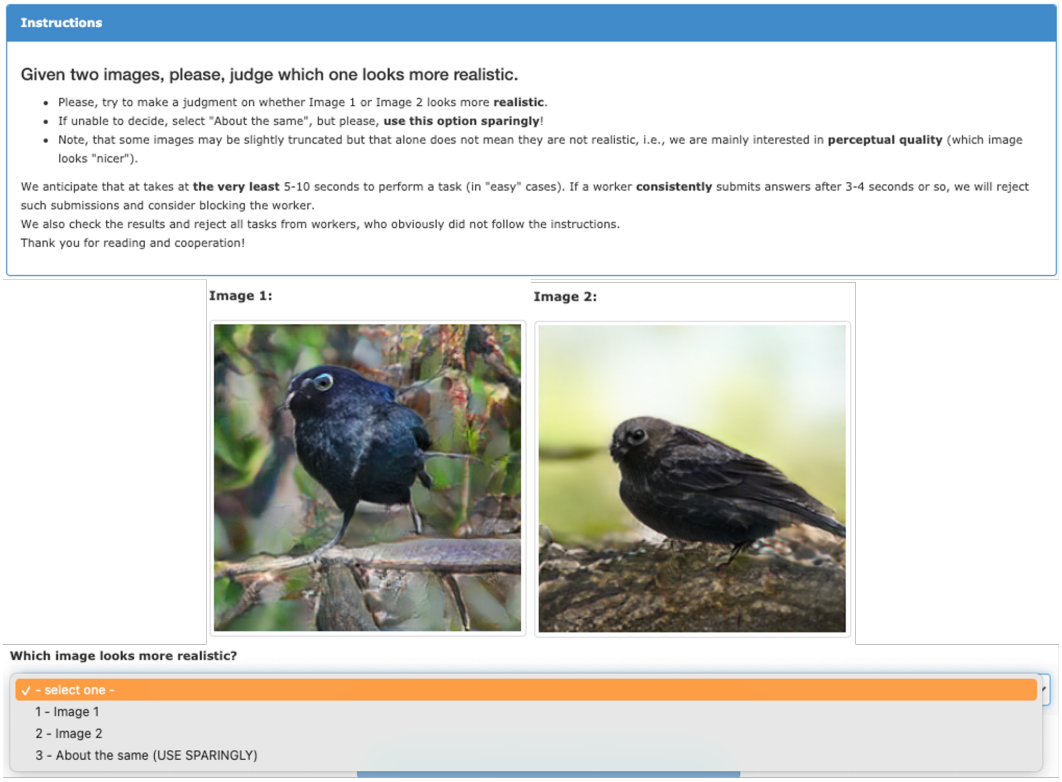

Figure 1: AMT Interface for the perceptual quality evaluation (the same for C-CUB and C-Flowers).

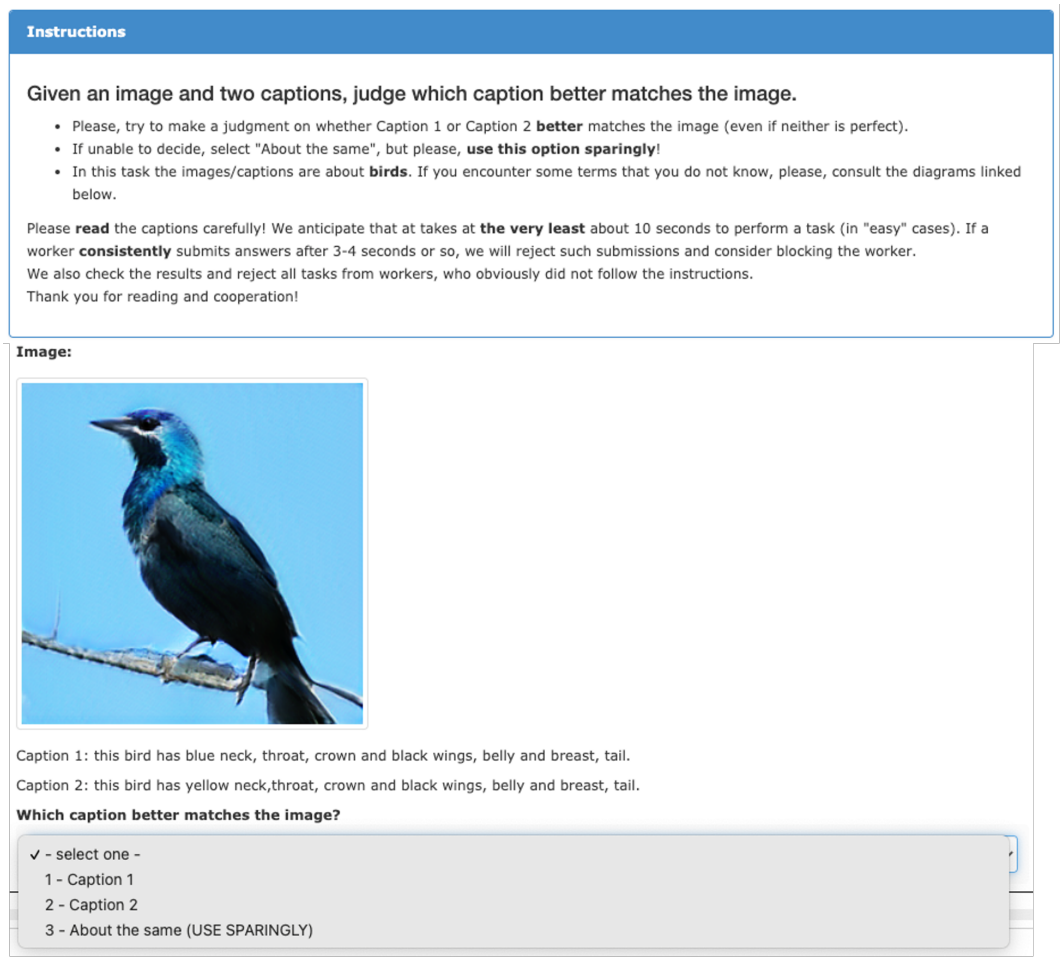

Figure 2: AMT Interface for the correctness evaluation on C-CUB.

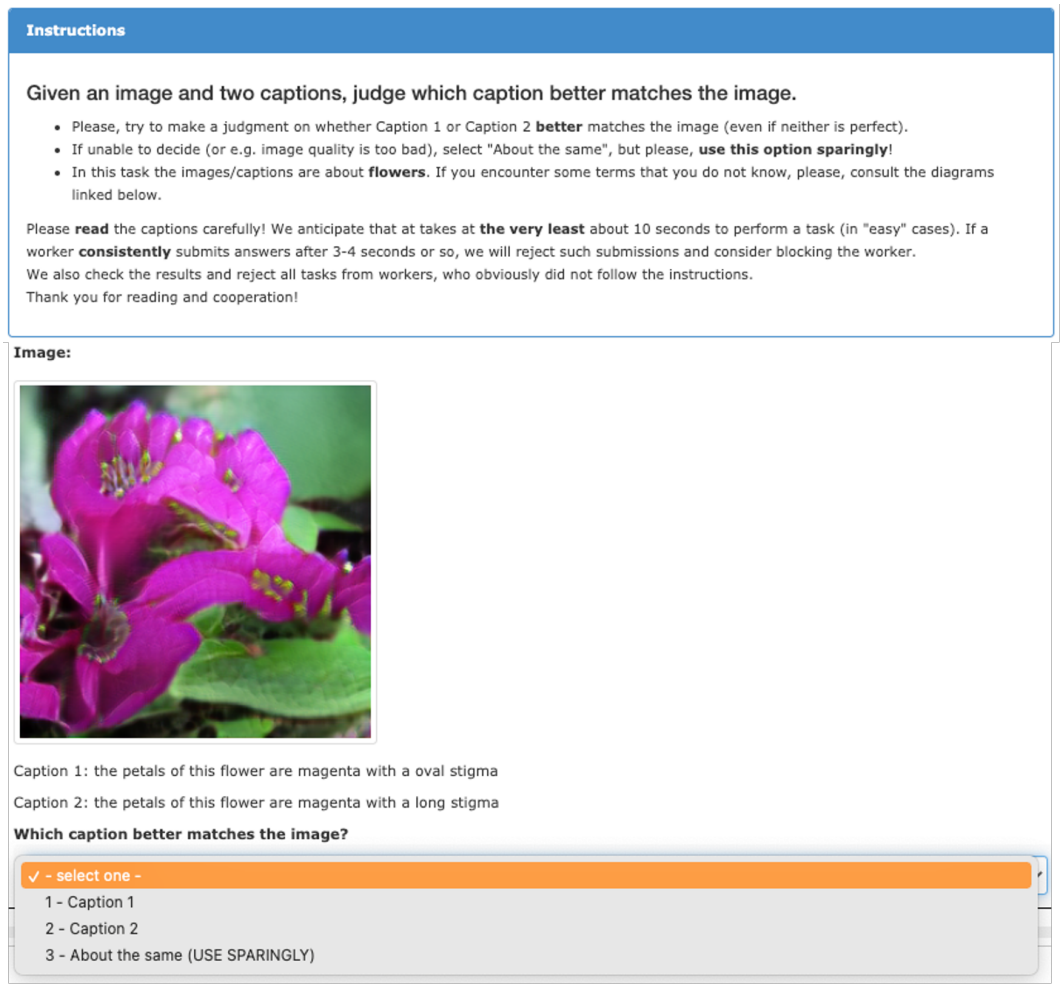

Figure 3: AMT Interface for the correctness evaluation on C-Flowers.

1.3 Qualitative Examples

We present additional qualitative examples from DMGAN on all four of our Seen/Swapped splits in Figures 4, 5, 6, 7.

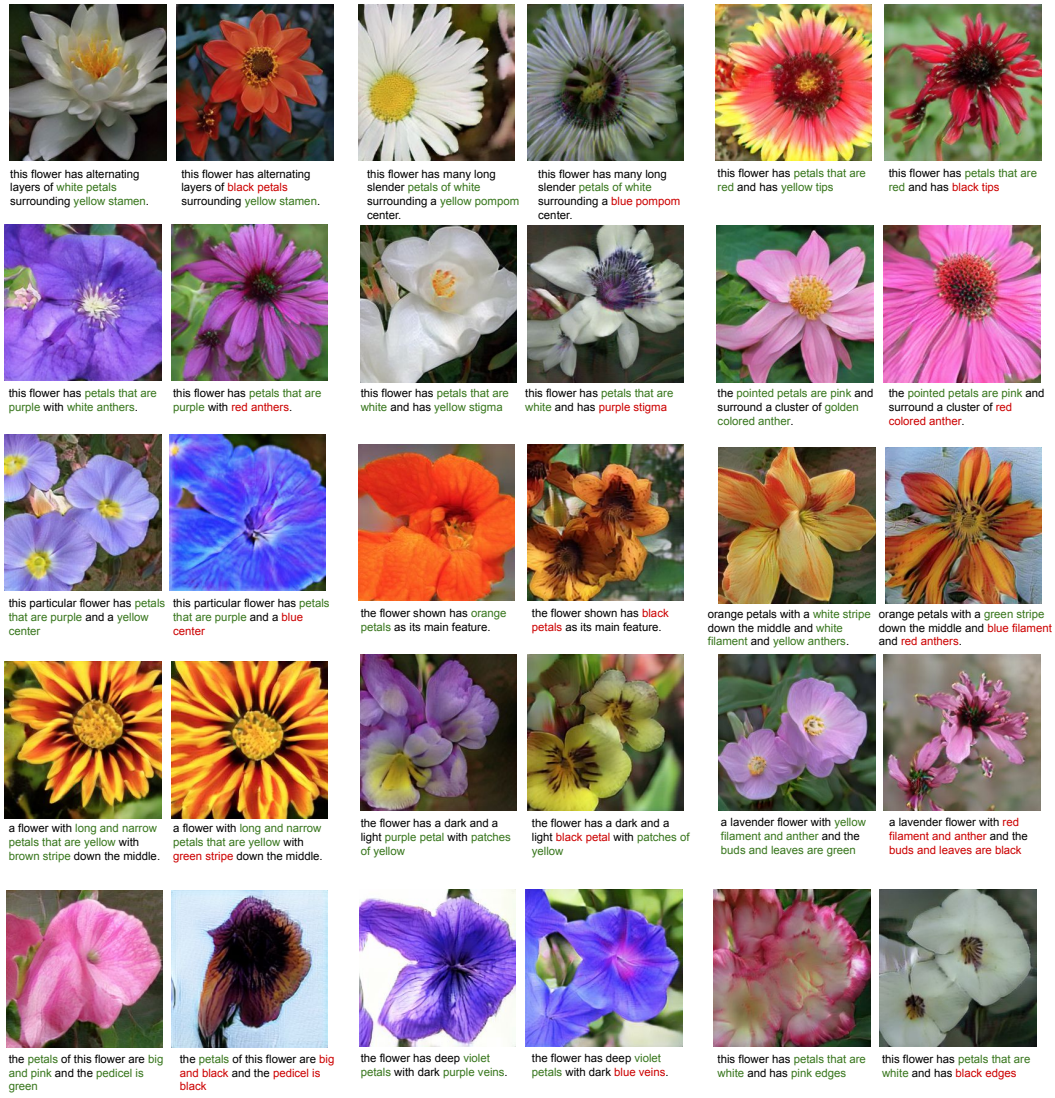

Figure 4: Seen adjective-noun pairs (illustrated in green; 1st, 3rd, and 5th columns) are swapped with unseen adjective-noun pairs (illustrated in red; 2nd, 4th, and 6th columns). A DMGAN model has been trained on the C-Flowers color dataset.

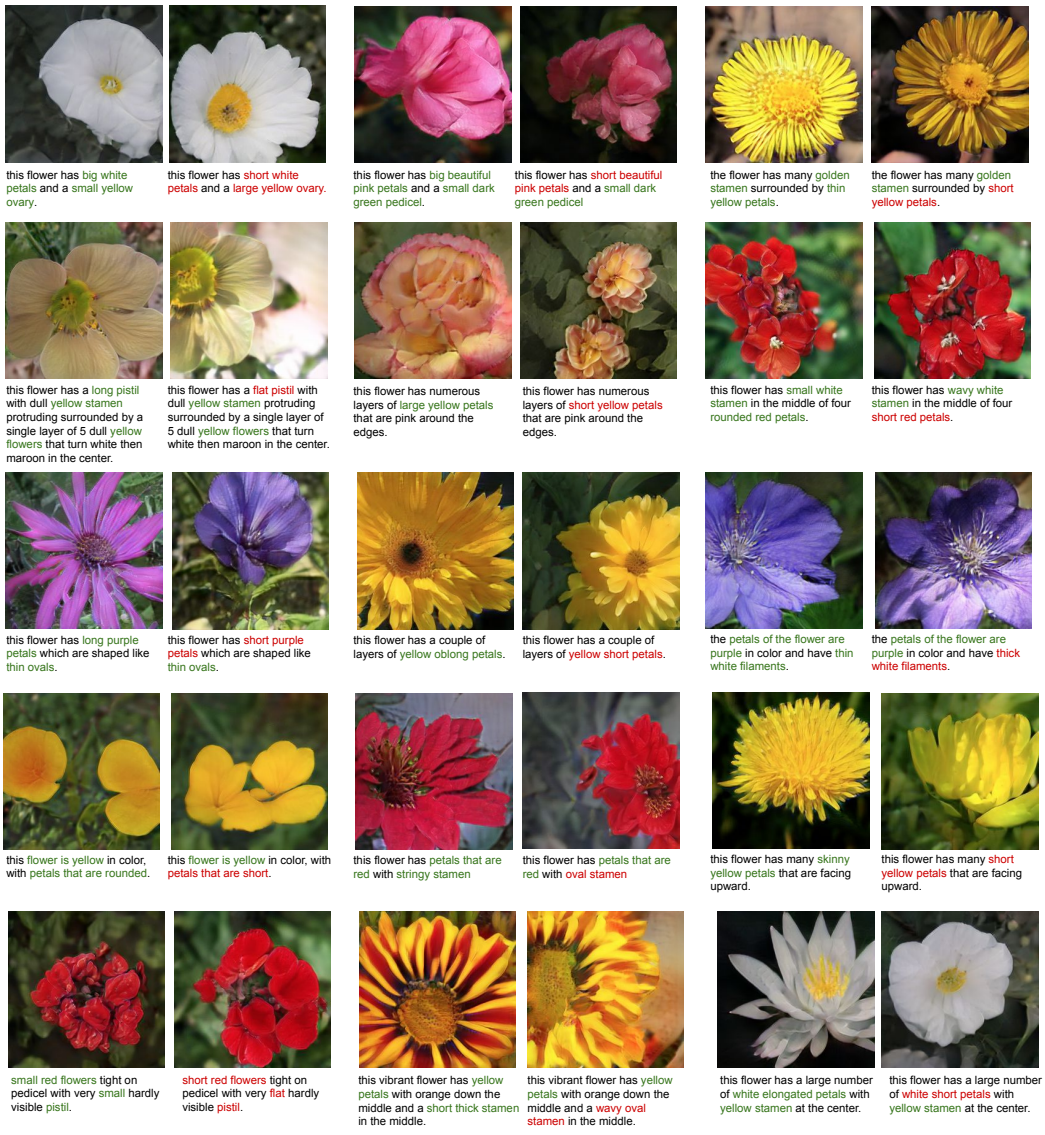

Figure 5: Seen adjective-noun pairs (illustrated in green; 1st, 3rd, and 5th columns) are swapped with unseen adjective-noun pairs (illustrated in red; 2nd, 4th, and 6th columns). A DMGAN model has been trained on the C-Flowers shape dataset.

Figure 6: Seen adjective-noun pairs (illustrated in green; 1st, 3rd, and 5th columns) are swapped with unseen adjective-noun pairs (illustrated in red; 2nd, 4th, and 6th columns). A DMGAN model has been trained on the C-CUB color dataset.

Figure 7: Seen adjective-noun pairs (illustrated in green; 1st, 3rd, and 5th columns) are swapped with unseen adjective-noun pairs (illustrated in red; 2nd, 4th, and 6th columns). A DMGAN model has been trained on the C-CUB shape dataset.

1.4 Compositional Splits Statistics

In this section, we present additional statistics for the C-CUB and C-Flowers datasets. With regards to C-CUB color, by comparing the training split in Figure 8 and test Unseen split in Figure 10, we see that the overall adjective and noun distributions are similar between the two even though the test Unseen split contains novel compositions. However, comparing the training split with test Swapped split in Figure 11, we notice that the adjective and noun distributions shift quite significantly due to the swaps. For instance, adjectives "purple" and "tan" become significantly more frequent, resulting in pairs like "purple bill" and "tan wing" to dominate the split. Similar observations can be made in other compositional splits (*e.g.* C-CUB Shape, C-Flowers Color, C-Flowers Shape). Thus, it becomes important to account for these imbalances during evaluation for test swapped splits.

1.5 Candidate Balancing for Evaluation on Test Swapped

As mentioned in the previous section, additional balancing needs to be done in order to prevent certain heldout pairs from overwhelming the metric computation. To do so, we first gather all the captions and the generated images by concept pairs. Then, the most frequent pair(s) is identified and the caption-image pairs belonging to that dominant pair(s) are subsammpled so that their numbers are only 1.25x larger than the next most dominant pair. The code block for such candidate balancing is provided in the following:

```
"""
gather_by_pair:
Dictionary that maps adj_noun pair to a list of entries that contain
    that pair.
An entry contains (conditioned text, generated_img)
dtype: "bird" or "flower"
split: "color" or "shape"
"""
def balance_candidates(gather_by_pair, dtype, split):
    change_type_counts = Counter()
    for change_type, data in gather_by_pair.items():
        change_type_counts[change_type] = len(data)
    top3 = change_type_counts.most_common(3)

    """
    for CUB color split where "purple bill" and "tan wing" are
    dominant
    """
    if dtype == 'bird' and split == 'color':
        max_num_dominant = int(min(top3[0][1], 1.25 * top3[-1][1]))
        dominant_1 = top3[0][0]
        dominant_2 = top3[1][0]

        dominant_1_cands = \
            random.sample(gather_by_pair[dominant_1],
                          max_num_dominant)
        dominant_2_cands = \
            random.sample(gather_by_pair[dominant_2],
                          max_num_dominant)

        all_cands = []
        for pair, entries in gather_by_pair.items():
            if pair == dominant_1 or pair == dominant_2:
                continue
            all_cands += entries
        all_cands += dominant_1_cands + dominant_2_cands
    else:
        max_num_dominant = int(min(top3[0][1], 1.25 * top3[1][1]))
        dominant = top3[0][0]

        dominant_cands = random.sample(gather_by_pair[dominant],
                                       max_num_dominant)
        all_cands = []
        for pair, entries in gather_by_pair.items():
            if pair == dominant:
                continue
            all_cands += entries
        all_cands += dominant_cands

    return all_cands
```

Listing 1: Candidate Balancing for Evaluation on Test Swapped

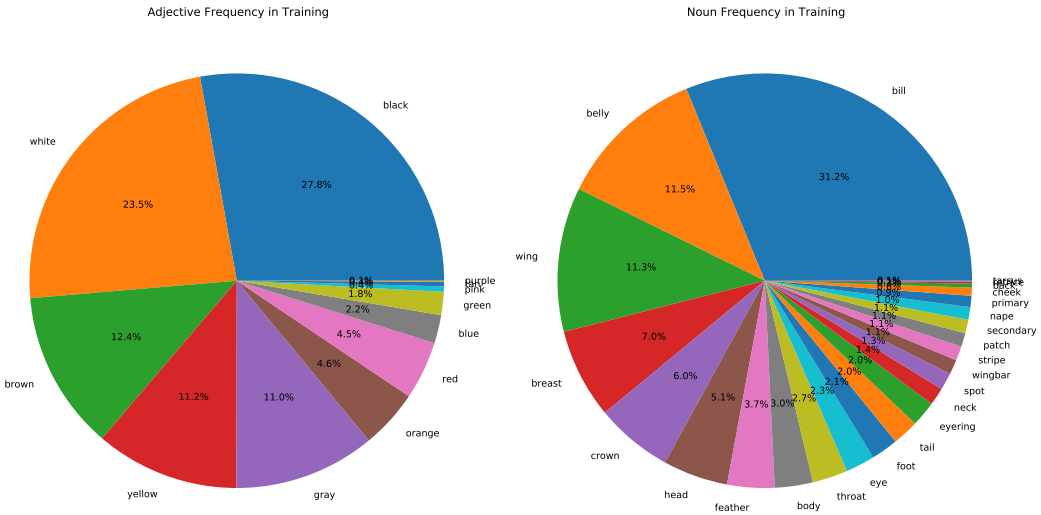

Figure 8: Adjective and noun distributions of training set for C-CUB Color.

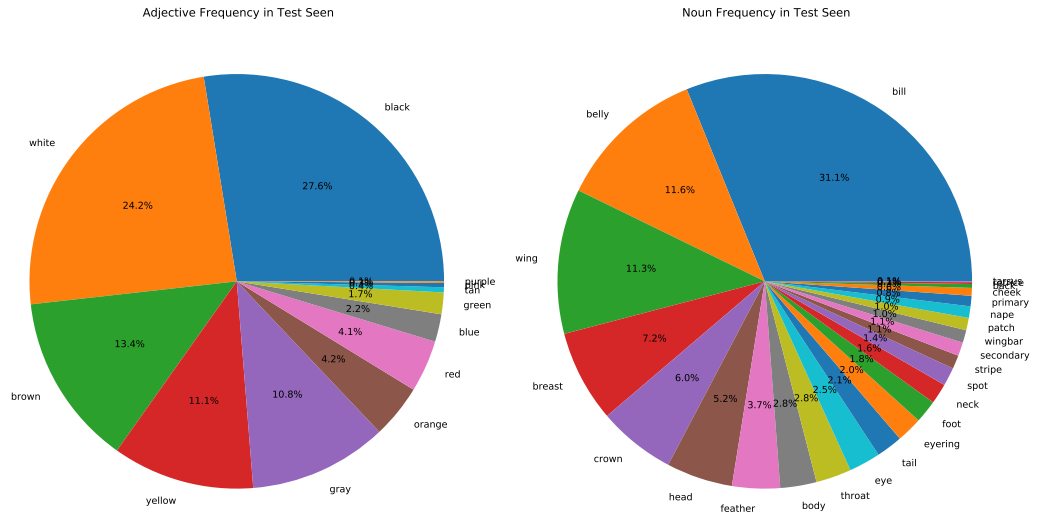

Figure 9: Adjective and noun distributions of test SEEN set for C-CUB Color.

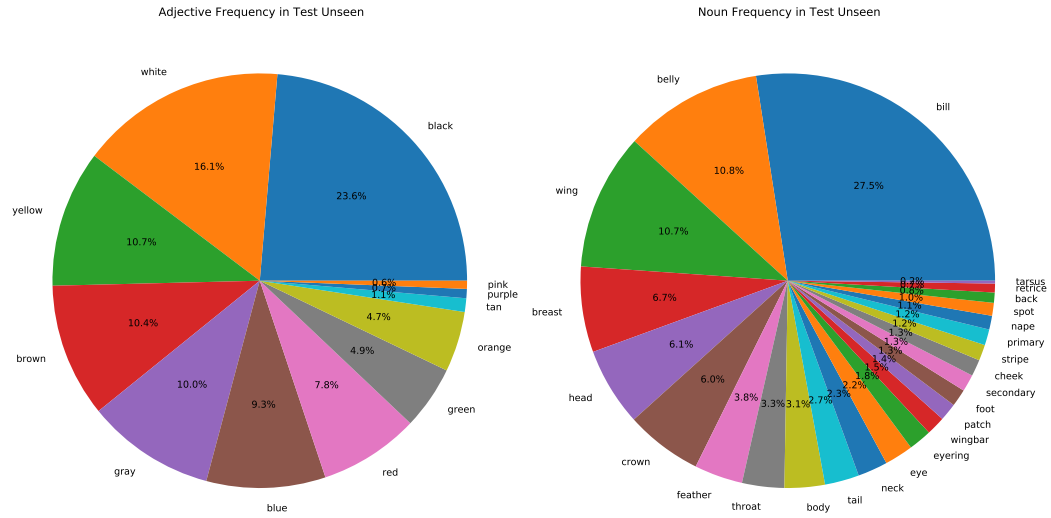

Figure 10: Adjective and noun distributions of test UNSEEN set for C-CUB Color.

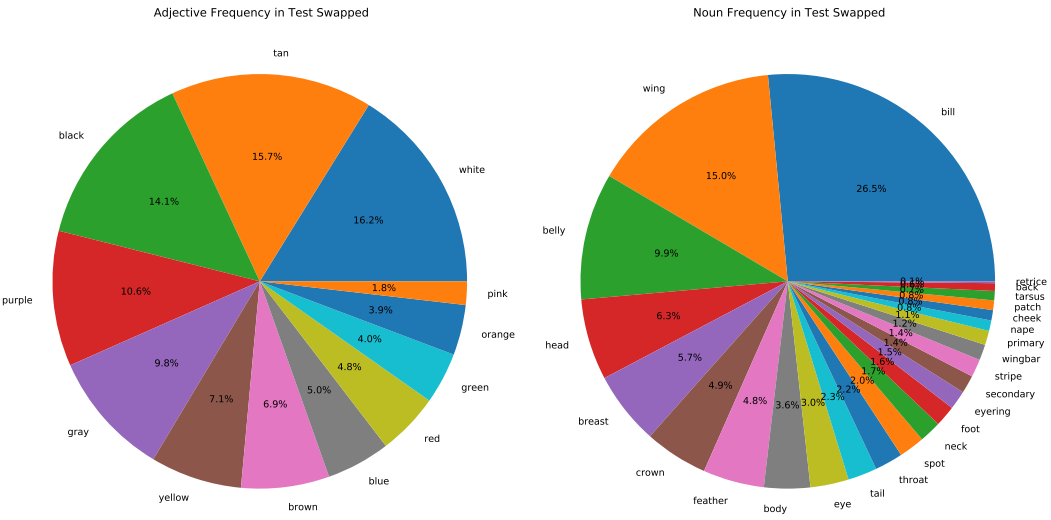

Figure 11: Adjective and noun distributions of test SWAPPED set for C-CUB Color.

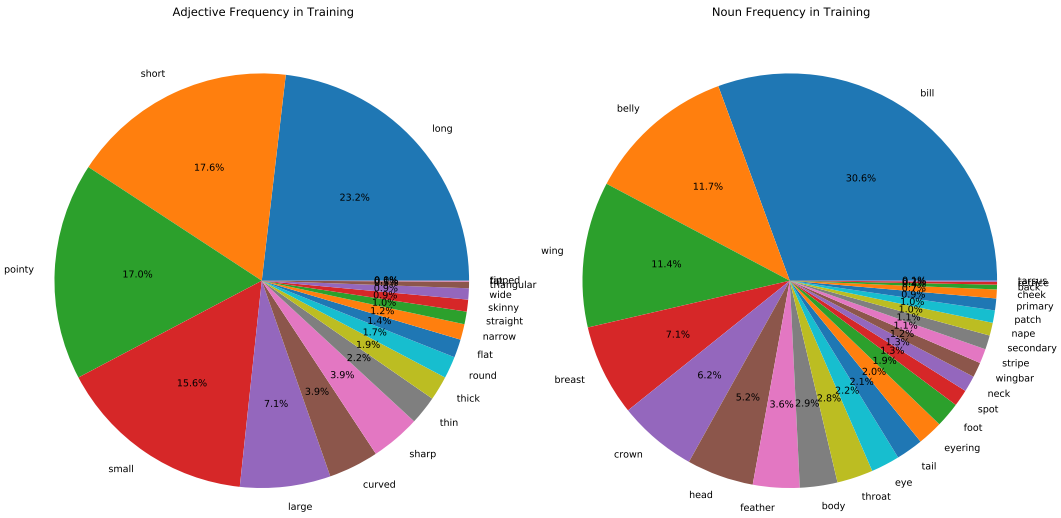

Figure 12: Adjective and noun distributions of training set for C-CUB Shape.

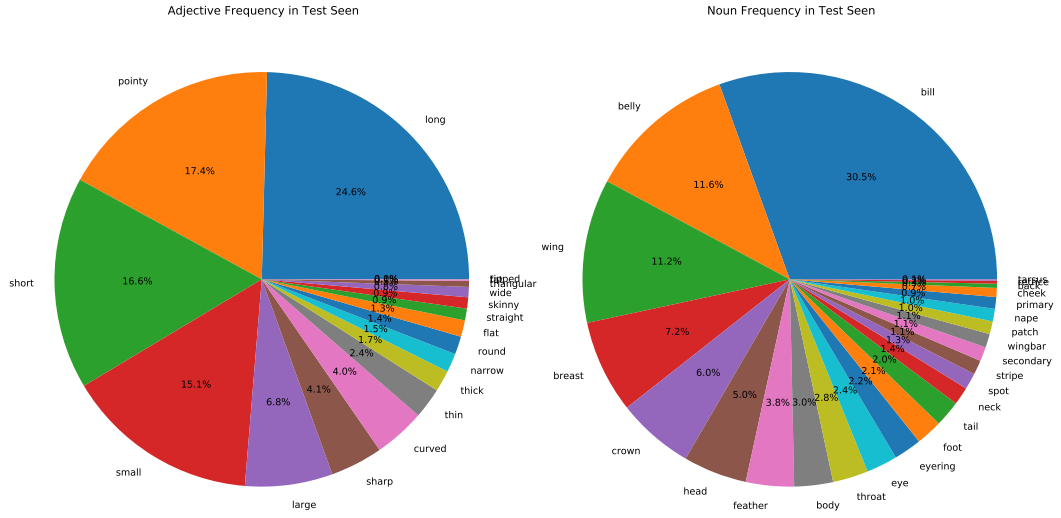

Figure 13: Adjective and noun distributions of test SEEN set for C-CUB Shape.

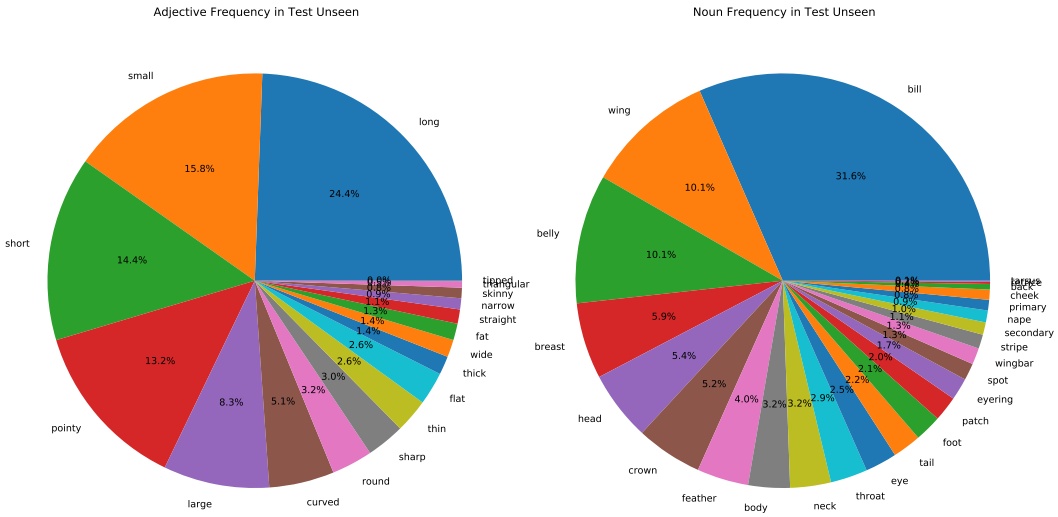

Figure 14: Adjective and noun distributions of test UNSEEN set for C-CUB Shape.

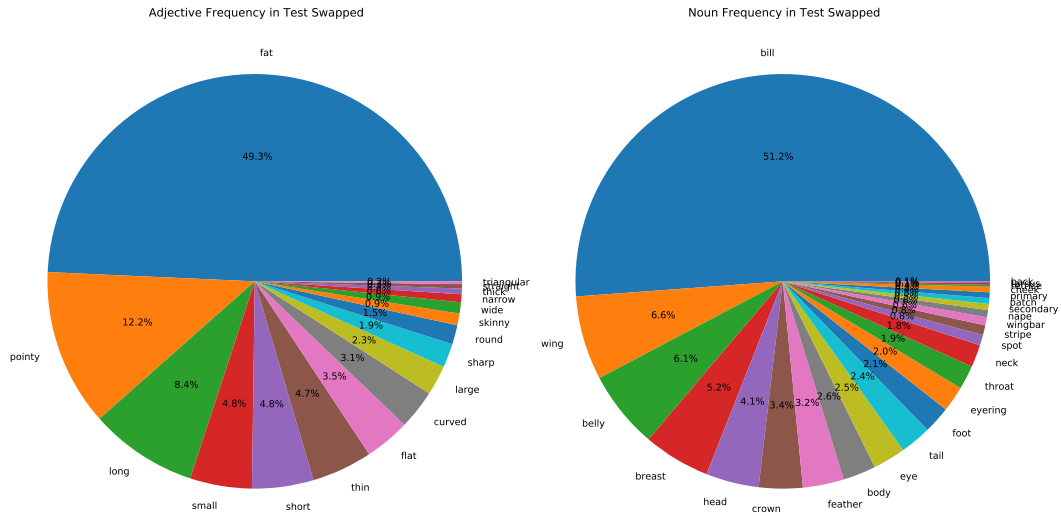

Figure 15: Adjective and noun distributions of test SWAPPED set for C-CUB Shape.

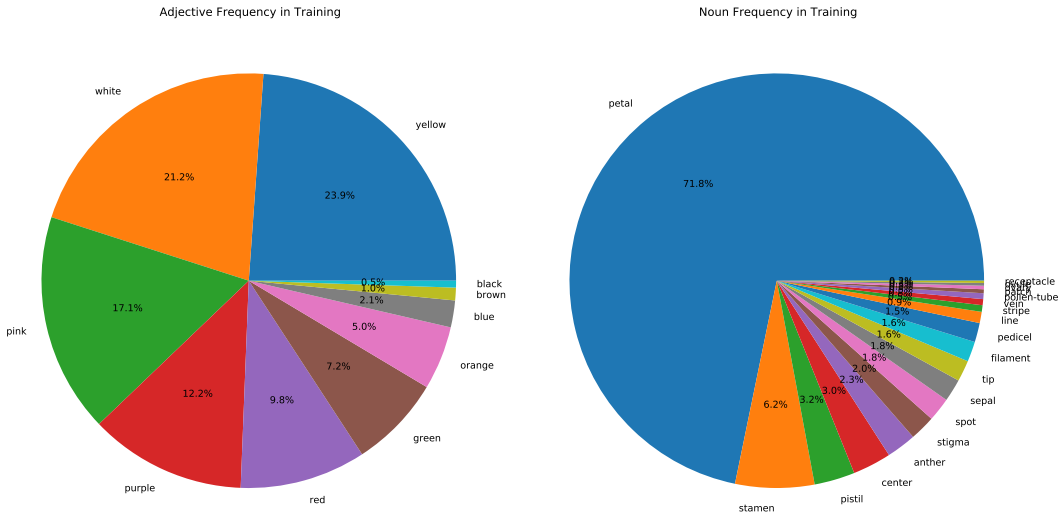

Figure 16: Adjective and noun distributions of training set for C-Flowers Color.

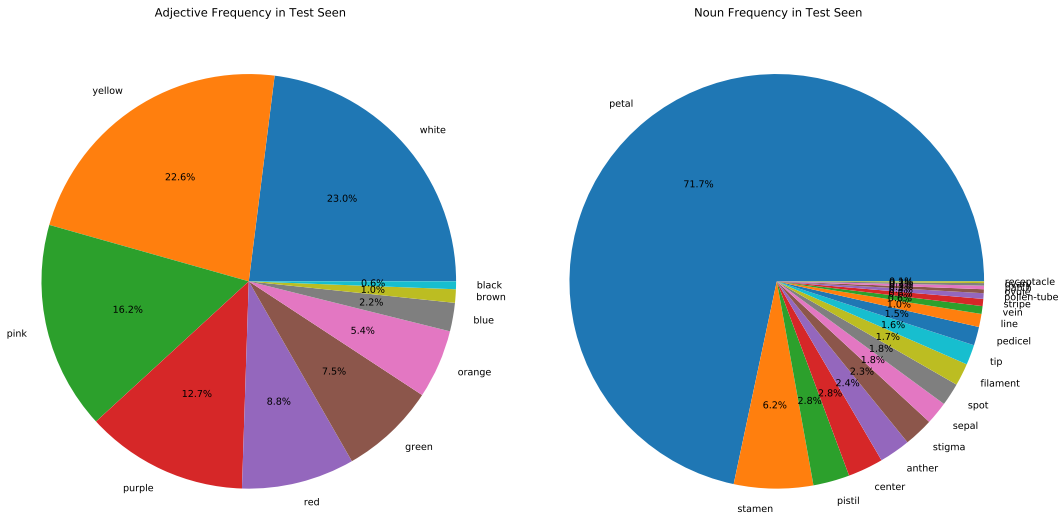

Figure 17: Adjective and noun distributions of test SEEN set for C-Flowers Color.

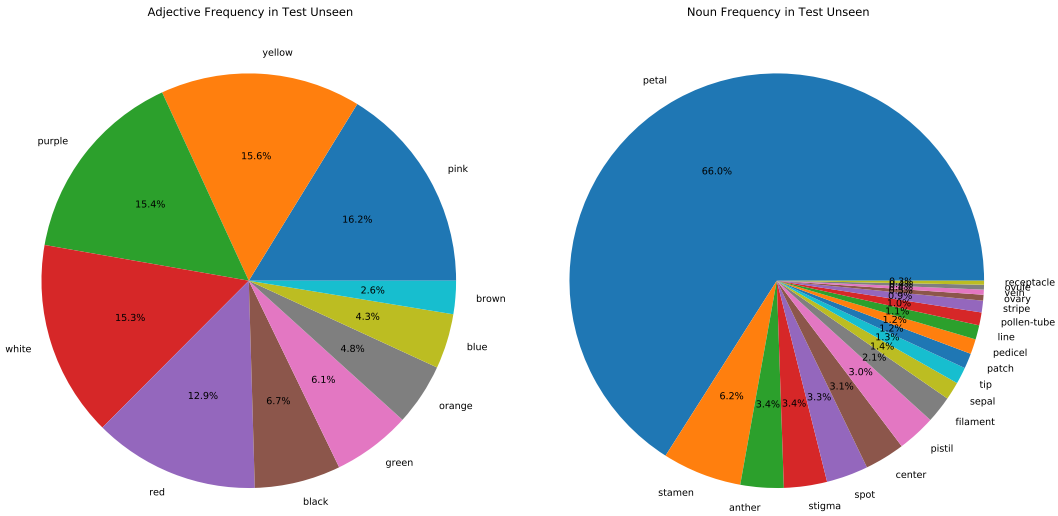

Figure 18: Adjective and noun distributions of test UNSEEN set for C-Flowers Color.

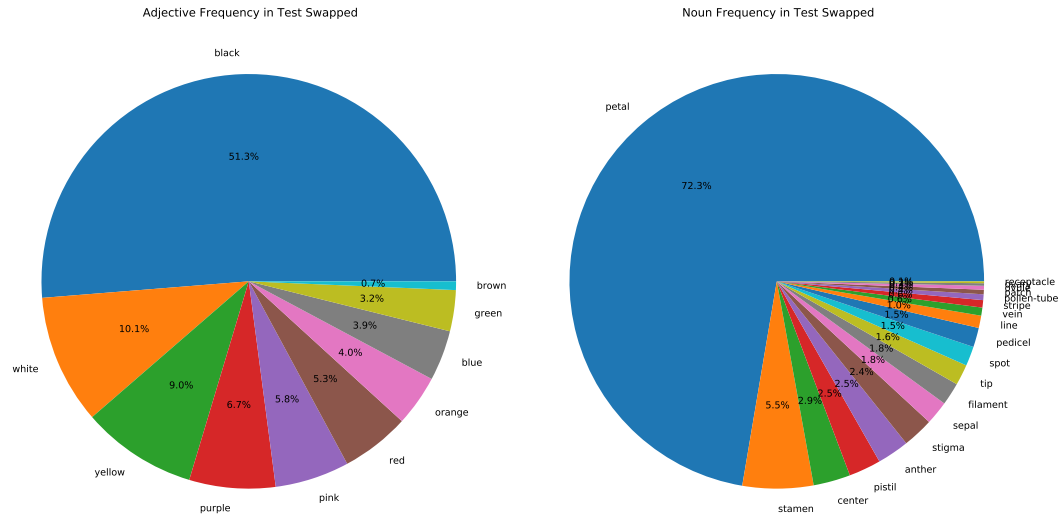

Figure 19: Adjective and noun distributions of test SWAPPED set for C-Flowers Color.

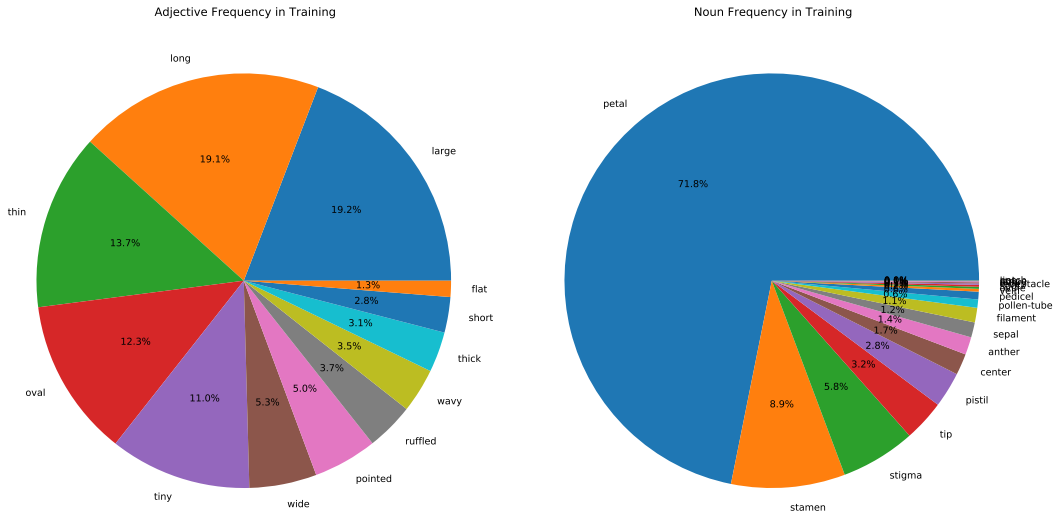

Figure 20: Adjective and noun distributions of training set for C-Flowers Shape.

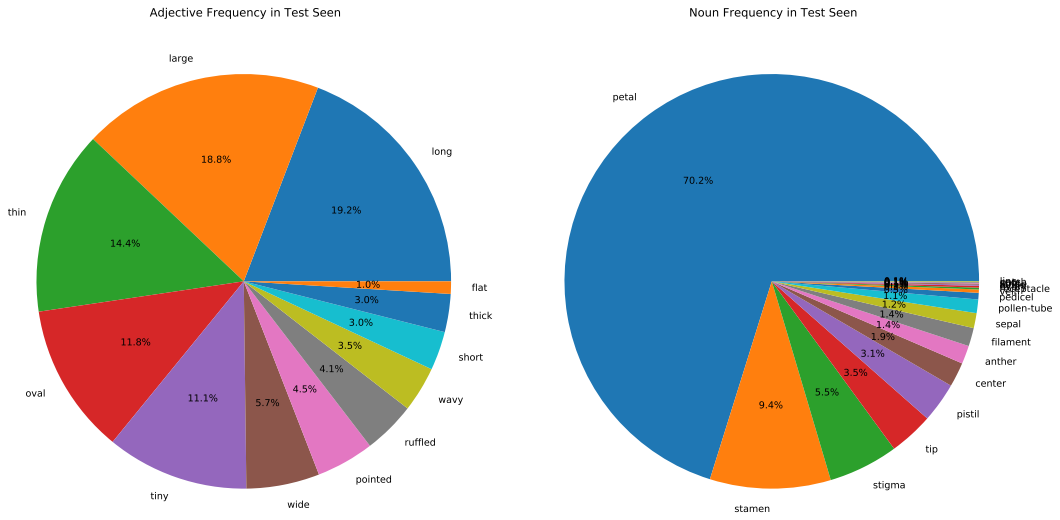

Figure 21: Adjective and noun distributions of test SEEN set for C-Flowers Shape.

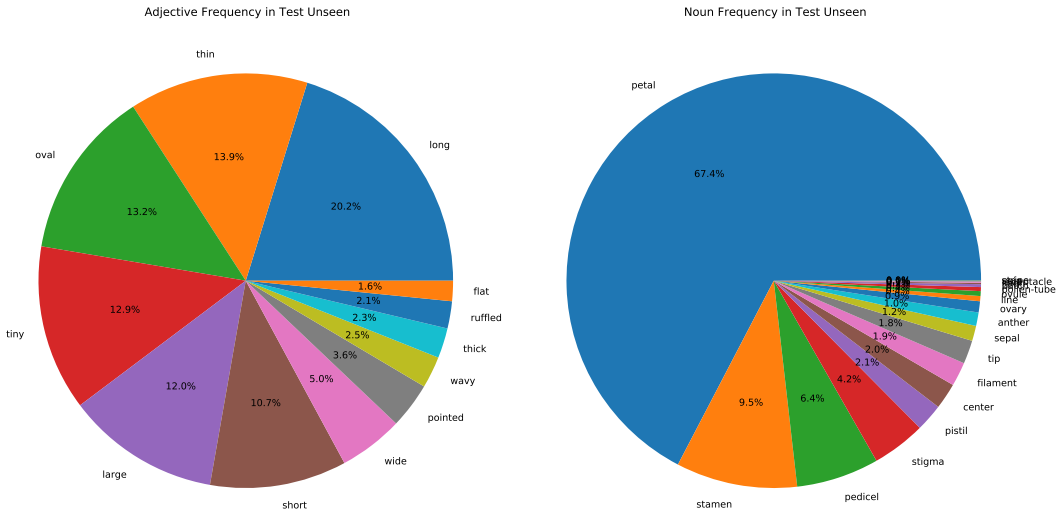

Figure 22: Adjective and noun distributions of test UNSEEN set for C-Flowers Shape.

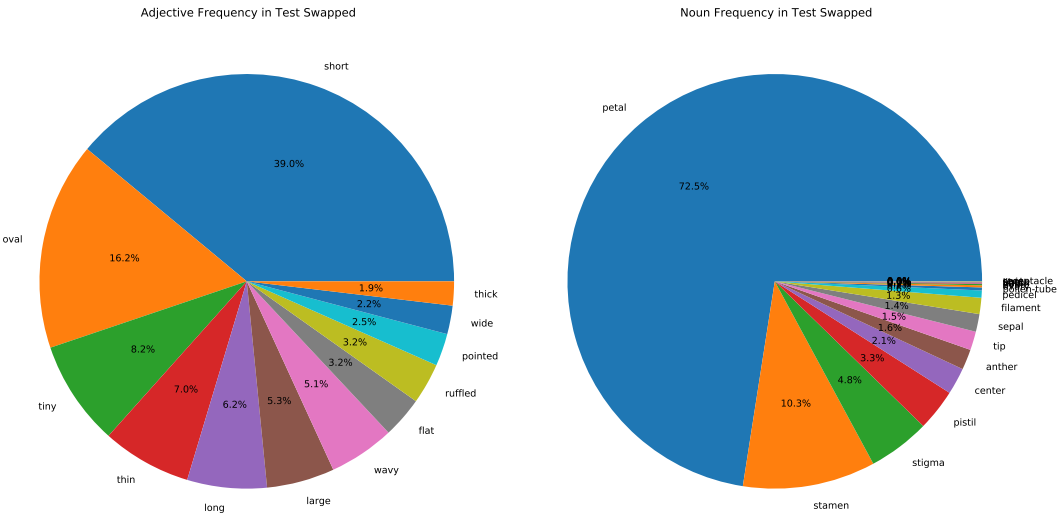

Figure 23: Adjective and noun distributions of test SWAPPED set for C-Flowers Shape.