GILT: Generating Images from Long Text

Ori Bar El, Ori Licht, Netanel Yosephian

Tel-Aviv University
{oribarel, oril, yosephian}@mail.tau.ac.il

Abstract

Creating an image reflecting the content of a long text is a complex process that requires a sense of creativity. For example, creating a book cover or a movie poster based on their summary or a food image based on its recipe. In this paper we present the new task of generating images from long text that does not describe the visual content of the image directly. For this, we build a system for generating high-resolution 256 \times 256 images of food conditioned on their recipes. The relation between the recipe text (without its title) to the visual content of the image is vague, and the textual structure of recipes is complex, consisting of two sections (ingredients and instructions) both containing multiple sentences. We used the recipe1M [11] dataset to train and evaluate our model that is based on a the StackGAN-v2 architecture [15].

1. Introduction

Generating images from text is a challenging task and has many applications in computer vision. Recent works have shown that Generative Adversarial Networks (GAN) are effective in synthesizing high-quality, realistic images from datasets with low variability and low-resolution [8, 3]. Further work also showed that given a text description, conditional GANs (cGAN) [5] generate convincing images related directly to the text content [9].

All recent text to image synthesis cGANs used a short visual description of the image, with low complexity and a consistent pattern of descriptions, and the images themselves had low variability. E.g. Zhange et al. [15, 16] used the CUB dataset [14] containing 200 bird species with 11,788 images and corresponding description and Oxford-102 dataset [6] containing 8,189 images of flowers from 102 different categories (see Figure 1). Recently the dataset recipe1M, containing 800K pairs of recipes and their corresponding images, was published as part of [11]. In comparison to the CUB and Oxford-102 datasets, this dataset has a high variability due to the variety of food categories and subcategories. Moreover, the text related to the image is complex. It consists of 2 sections (ingredients and instructions), that together might contain tens of lines (e.g. Figure 2).

Figure 1. Image samples from CUB and Oxford-102 datasets, and their corresponding text descriptions.

We propose a novel task of synthesizing images from long text, that is related to the image but does not contain visual description of it. Specifically, We propose a baseline to this task by combining the state-of-the-art Stacked Generative Adversarial Network [15] and the two proposals of recipe embeddings computed in im2recipe [11] to generate food images conditioned on their recipes. We also present extensive qualitative and quantitative experiments using hu-
man ranking, MS-SSIM [13] and inception scores [10], to compare the effectiveness of the two embedding methods.

Our code is available at https://github.com/netanelyo/Recipe2ImageGAN.

2. Related Work

Generating high-resolution images conditioned on text descriptions is a fundamental problem in computer vision. This problem is being studied extensively and various approaches were suggested to tackle it. Deep generative models, such as [16, 15, 12, 17], achieved tremendous progress in this domain. In order to get high-resolution images they used multi-stage GANs, where each stage corrects defects and adds details w.r.t. the previous stage. In [12], Xu et al. use Deep Attentional Multimodal Similarity Model (DAMSM) for text embedding. DAMSM learns two neural networks that map sub regions of the image and words of the sentence to a common semantic space. Thus, measures the image-text similarity at the word level to compute a fine-grained loss for image generation.

3. Learning Embeddings

Our cGAN uses the embedding of the entire recipe (except for its title) as a condition. To generate that embedding we leverage the methods used in [11]. They [11] proposed two types of embedding methods, where the second adds a semantic regularization loss component. Throughout this paper we will refer to the first method without semantic regularization as NOREG, and the second with semantic regularization as REG. The embedding methods are composed of the following steps (for the concrete architecture see the original paper).

1. Preliminary embedding of the ingredients.
2. Preliminary embedding of the cooking instructions.

3. Joint neural embedding of the entire recipe (using the concatenation of the former preliminary embeddings) and the image into a common space, using cosine similarity loss between the embeddings of recipe-image pairs.

4. Adding a semantic regularization loss using a high-level classification objective (used in REG only)

We employ these methods as explained in the original paper.

4. Stacked Generative Adversarial Networks

Originally, GANs [4] are a combination of two models that are trained to compete with each other. In the training process both the generator $G$ and the discriminator $D$ are trained. $G$ is optimized to reproduce images similar to the original data distribution, by generating images that are difficult for the discriminator $D$ to distinguish from the true images. $D$ is trained to distinguish between real images and fake synthetic ones, generated by $G$. This training is similar to solving a minmax of 2 players game, with the objective function, [4]

$$
\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}}[\log D(x)] + \mathbb{E}_{z \sim p_z}[\log(1 - D(G(z))],
$$

where $x$ is an image sampled from the real distribution $p_{data}$ and $z$ is the noise vector, which is sampled from a prior distribution $p_z$ (e.g Uniform or Gaussian), used by $G$ to generate the synthetic image.

In the case of Conditional GANs [12, 15, 16, 9] both the generator and the discriminator are compelled to consider another variable $c$. We denote $D(x, c)$ and $G(z, c)$ to be the generator $G$ and the discriminator $D$ conditioned by $c$, respectively. Meaning that $G$ is able to generate images, and $D$ discriminate them, conditioned on $c$.

The StackGAN-v2 model, introduced in StackGAN++, by Zhang et al. [15], is an end-to-end network for modeling a series of multi-scale image distributions. The architecture of this model is consisted of several generators and discriminators in a tree-like structure framework (for the concrete architecture see the original paper). Given a noise vector $z \sim p_z$ and condition $c$ StackGAN-v2 generates images from low-resolution to high-resolution from different branches of the tree. In our case $c$ is one of the recipe embeddings from section 4. Overall, we have one model for each of the two embeddings.

5. Implementation details

We compare two types of embedding methods from [11]. The first method is based on cosine-similarity loss only and is of size 1024. The second method uses additionally a high-level classification objective to compute embedding of size
For training the StackGAN-v2 [15] model we used a batch size of 24. Trying to use a larger batch size resulted in a mode-collapse. The text-embedding dimension parameter presented in StackGAN-v2 was of size 128. At first we used this parameter with our training and got poor results. We realized that by projecting the rich text on this small dimension, we might omit discriminative subtleties between different recipes. As a result we used 1024 as the text-embedding dimension parameter for both of the embeddings methods. In order to accelerate the training process we used hdf5 (hierarchical data format) to map files to memory. All neural models were implemented using PyTorch framework. All other parameters are identical to [11] and [15]. The models were trained on 3 Nvidia Titan-X GPUs, each has 12GB of memory, for 100 epochs on each of the embedding methods.

6. Experiments

To evaluate our models, we conduct quantitative, in the form of Inception Score (IS) [10], and qualitative, in the form of Human Rankings (HR), evaluations. We compare the aforementioned evaluation methods on images generated by two different text embedding methods, which both are computed using [11]. In addition, we show outputs from several state-of-the-art and previous state-of-the-art text-to-image synthesis models, which indicate that generating re-
alistic food-images based on their description is a challenging task, and all the more so based on their recipe. Moreover, we examine the diversity of the generated images using MS-SSIM [13].

### 6.1. Datasets and evaluation metrics

Recipe1M [11] contains over 1 million recipes and 800k food images. Due to hardware limitations, we used a training set of 52k and an evaluation set of 24k recipe-image pairs. In the pre-processing stage, the images were down-scaled from $256 \times 256$ to $128 \times 128$ and $64 \times 64$, in-order to train on different image scales. Further more, the images were cropped and horizontally flipped randomly. This was a best-effort to focus on the food object in the image, but from time to time, the cropping eliminated important details from the original image.

**Evaluation metrics.** Even though evaluating generative models is often a difficult task (as mentioned in [1]), to compare between the generated images in both embedding methods quantitatively (numerically) we use Inception-Score,

$$IS = \exp(E_x D_{KL}(p(y|x)||p(y))),$$

(2)

Where $x$ denotes a single generated sample, $y$ is the predicted label, $p(y|x)$ and $p(y)$ are the conditional and marginal class distributions, respectively, and $D_{KL}$ is the Kullback-Leibler (KL) divergence. Intuitively, the IS measures the diversity, with respect to ImageNet [2] classes, and clarity of the generated images. Therefore, the KL-divergence (hence the IS), of a successful generator should be large. We evaluate the IS on the evaluation set, which contains 24k randomly chosen samples. In spite of the suboptimality of IS, stated in [1], it is the most popular method to evaluate generative models.

| Metric      | REG         | NOREG       |
|-------------|-------------|-------------|
| Inception Score | 4.42 ± 0.17 | 4.55 ± 0.20 |
| Human Ranking  | Q1: 2.62    | Q1: 2.88    |
|              | Q2: 2.24    | Q2: 2.70    |
|              | Q3: 3.05    | Q3: 3.72    |

Table 1. Inception scores and average human rankings of our results.

Figure 6. Example results of food images generated by HDGAN [17], AttnGAN [12] and StackGAN++ [15] conditioned on text descriptions.

Due to the aforementioned suboptimality of IS and the fact that it does not reflect the correlation between the generated image and the recipe it is conditioned on, a qualitative evaluation metric was used. Therefore, 30 people were asked to rank, a total of 10 samples, in several aspects:

1. The strength of the relation between a generated image and its corresponding recipe.
2. The strength of the relation between a generated image and its corresponding real image.
3. In which degree the image appears to be a real food image.

The final human rankings are the average of the above.
| Embedding Type | MS-SSIM score |
|---------------|---------------|
| REG           | 0.17          |
| NOREG         | 0.07          |

Table 2. MS-SSIM[13] score of randomly chosen images from our results.

6.2. Quantitative and qualitative results

As one can see by the examples shown in Figure 6, the state-of-the-art and previous state-of-the-art text-to-image synthesis models yield unsatisfying results, in spite of the concise and visually descriptive text on which it is conditioned.

We compare our model between the two mentioned text embedding methods, i.e., embedding with semantic regularization and without it. The inception scores and average human ranks for our models are reported in Table 1. As can be seen in the table, the embedding without semantic regularization achieves the better IS and HR (in all aspects) scores. Representative examples are compared in Figures 3, 4 and 5.

Human rankings. 10 corresponding pairs of generated images were chosen, from each embedding method evaluation results (i.e., images that were generated conditioned on the same recipe). Our subjects were asked to rank the images in the aforementioned aspects, on a scale of 1 to 5. As mentioned earlier, the model trained conditioned on the cosine-similarity based embedding method yielded results that are close to real-like images. It is worth mentioning that there were real food images that were given less-than-or-equal rank in comparison to generated images (in the cosine-similarity embedding evaluation).

6.3. Diversity

The most successful method to evaluate image similarity, as mentioned in [7], is multi-scale structural similarity (MS-SSIM [13]). The method attempts to discount the image aspects that are not important to the human eye. To evaluate the diversity of the images generated by our model, 200 random images from the evaluation set were chosen, and we calculated the MS-SSIM score for each pair. The results can be seen in Table 2. One can see that the embedding method without semantic regularization, achieved better score (lower is better), i.e., generated more diversified images. These results might be explained by the fact that when using semantic regularization, the classification based regularization aims to map the recipe embedding to one of the 1048 classes in a discrete manner, instead of utilizing the entire space.

7. Conclusions

In this paper, we propose an end-to-end system for high-resolution long text to image synthesis using Stacked Generative Adversarial Network (StackGAN-v2). We compare two embedding types, one based on cosine-similarity (NOREG) and the second combines a high-level classification objective (REG). The proposed methods prove their ability to generate photo-realistic images of food from their text recipe (ingredients and instructions only). Herein, we provide a baseline for this novel task. It is worth mentioning that the quality of the images in the recipe1M dataset [11] is low in comparison to the images in CUB [14] and Oxford-102 datasets [6]. This is reflected by lots of blurred images with bad lighting conditions, “porridge-like images” and the fact that the images are not square shaped (which makes it difficult to train the models). This fact might give an explanation to the fact that both models succeeded in generating “porridge-like” food images (e.g. pasta, rice, soups, salad) but struggles to generate food images that have a distinctive shape (e.g. hamburger, chicken, drinks). From the results, it is evident that the method NOREG outperforms the method REG, by generating more vivid images with more photorealistic details. Moreover, the inception score and diversity measures of the former is better than the latter. Overall, we show that although REG outperforms NOREG in a classification task (see [11]) it is inferior for generating new images.

References

[1] S. Barratt and R. Sharma. A Note on the Inception Score. ArXiv e-prints, Jan. 2018.
[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pages 248–255. Ieee, 2009.
[3] E. Denton, S. Chintala, A. Szlam, and R. Fergus. Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks. ArXiv e-prints, June 2014.
[4] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Networks. ArXiv e-prints, June 2014.
[5] M. Mirza and S. Osindero. Conditional Generative Adversarial Nets. ArXiv e-prints, Nov. 2014.
[6] M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing, Dec 2008.
[7] A. Odena, C. Olah, and J. Shlens. Conditional Image Synthesis With Auxiliary Classifier GANs. ArXiv e-prints, Oct. 2016.
[8] A. Radford, L. Metz, and S. Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. ArXiv e-prints, Nov. 2015.
[9] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative Adversarial Text to Image Synthesis. ArXiv e-prints, May 2016.
[10] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. CoRR, abs/1606.03498, 2016.
[11] A. Salvador, N. Hynes, Y. Aytar, J. Marin, F. Oflie, I. Weber, and A. Torralba. Learning cross-modal embeddings for
cooking recipes and food images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017. 1, 2, 3, 4, 5

[12] Q. H. H. Z. G. X. H. X. H. Tao Xu, Pengchuan Zhang. Attention: Fine-grained text to image generation with attentional generative adversarial networks. 2018. 2, 4

[13] Z. Wang, E. P. Simoncelli, and A. C. Bovik. Multiscale structural similarity for image quality assessment. In The Thirteenth Asilomar Conference on Signals, Systems Computers, 2003, volume 2, pages 1398–1402. Vol.2, Nov 2003. 2, 5

[14] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010. 1, 5

[15] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. Metaxas. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. arXiv: 1710.10916, 2017. 1, 2, 3, 4

[16] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In ICCV, 2017. 1, 2

[17] Z. Zhang, Y. Xie, and L. Yang. Photographic text-to-image synthesis with a hierarchically-nested adversarial network. CoRR, abs/1802.09178, 2018. 2, 4
Supplementary Materials

More Results of Generated Images by Both Embedding Methods and their Corresponding Recipes.

**Real image**

*Sourdough with pancetta*
- ingredients:
  - 1 loaf
  - 1 pancetta
  - 2 loaves
  - 1 bacon
  - 1 garlic
  - 1 chili pepper
  - 1 cup
  - 1 pepper
  - 1 pinch
  - 1 parmesan cheese
  - Extra virgin oil

*Add tocino, apple, chili pepper, garlic, zest and juice of 1 lemon*

**With semantic regularization**

*Hot Pepper and Garlic Shrimp*
- ingredients:
  - 2 lb large shrimp, peeled and deveined
  - 10 large garlic cloves, peeled and thinly sliced
  - 1 cup fresh chile pepper flakes
  - 2 teaspoons fine sea salt
  - 1 cup extra virgin olive oil
  - 1 tablespoon fresh lemon juice

*Instructions:
  1. Heat oil in a large heavy skillet over moderately low heat and add garlic, red pepper flakes and sea salt.
  2. Cook until garlic is pale golden, about 4 to 5 minutes. Increase heat to moderately high and add chilies.
  3. Simmer, stirring occasionally, about 3 to 4 minutes.
  4. Remove from heat and stir in local rice and transfer to a serving bowl.

**Without semantic regularization**

*Southeast Chicken Panini*
- ingredients:
  - 1 lb thin slice tortilla
  - 1 tablespoon chopped curry
  - 2 cloves garlic, minced
  - 1/2 teaspoon salt
  - 1/4 teaspoon black pepper
  - 1/2 teaspoon ground cumin
  - 1/2 teaspoon paprika
  - 1/4 teaspoon cumin
  - 1/4 teaspoon salt
  - 1/4 teaspoon pepper

*Instructions:
  1. Heat the skillet on medium heat.
  2. Cook sandwich 5 minutes on each side or until cheese is melted and sandwich is golden brown on both sides, gently pressing down top of sandwich with spatula to flatten slightly, as in a pan.

**Real image**

*Banh Mi Pesto*
- ingredients:
  - 8 oz Linguine Noodles
  - 1/4 cup Thai Chili Sauce
  - 1/2 cup Mayonnaise
  - 1/2 cup Lime Juice
  - 1/2 cup Hot Sauce

*Instructions:
  1. Cook noodles as directed.
  2. Whisk all the ingredients together in a bowl.
  3. Mix together and serve.

**With semantic regularization**

*Steak Soup*
- ingredients:
  - 1 cup chopped tomatoes
  - 1/2 cup chopped onions
  - 1/2 cup chopped celery
  - 1/2 cup chopped green peppers
  - 1/2 cup chopped green beans
  - 1/2 cup chopped carrots
  - 1/2 cup chopped potatoes
  - 1/2 cup chopped tomatoes
  - 1/2 cup chopped celery
  - 1/2 cup chopped green beans
  - 1/2 cup chopped carrots
  - 1/2 cup chopped green peppers
  - 1/2 cup chopped onions
  - 1/2 cup chopped tomatoes
  - 1/2 cup chopped celery
  - 1/2 cup chopped green beans
  - 1/2 cup chopped carrots
  - 1/2 cup chopped green peppers
  - 1/2 cup chopped onions
  - 1/2 cup chopped tomatoes
  - 1/2 cup chopped celery
  - 1/2 cup chopped green beans
  - 1/2 cup chopped carrots
  - 1/2 cup chopped green peppers
  - 1/2 cup chopped onions
  - 1/2 cup chopped tomatoes
  - 1/2 cup chopped celery
  - 1/2 cup chopped green beans
  - 1/2 cup chopped carrots
  - 1/2 cup chopped green peppers
  - 1/2 cup chopped onions
  - 1/2 cup chopped tomatoes
  - 1/2 cup chopped celery
  - 1/2 cup chopped green beans
  - 1/2 cup chopped carrots
  - 1/2 cup chopped green peppers
  - 1/2 cup chopped onions

*Instructions:
  1. Heat the skillet on medium heat.
  2. Cook sandwich 5 minutes on each side or until cheese is melted and sandwich is golden brown on both sides, gently pressing down top of sandwich with spatula to flatten slightly, as in a pan.

**Without semantic regularization**

*Sugar Free Granola*
- ingredients:
  - 1 cup chopped dates
  - 1 cup raisins
  - 1 cup chopped walnuts
  - 1 cup rolled oats
  - 1 cup honey
  - 1 cup peanut butter
  - 1/2 cup sunflower seeds
  - 1/2 cup almonds
  - 1/2 cup dried apricots
  - 1/2 cup dried cranberries
  - 1/2 cup dried raisins
  - 1/2 cup dried apricots
  - 1/2 cup dried cranberries
  - 1/2 cup dried raisins
  - 1/2 cup honey
  - 1/2 cup peanut butter
  - 1/2 cup sunflower seeds
  - 1/2 cup almonds
  - 1/2 cup dried apricots
  - 1/2 cup dried cranberries
  - 1/2 cup dried raisins

*Instructions:
  1. Heat the skillet on medium heat.
  2. Cook sandwich 5 minutes on each side or until cheese is melted and sandwich is golden brown on both sides, gently pressing down top of sandwich with spatula to flatten slightly, as in a pan.

*Steak Soup*
Real image

With semantic regularization

Without semantic regularization

Real image

With semantic regularization

Without semantic regularization

Real image

With semantic regularization

Without semantic regularization

Real image

With semantic regularization

Without semantic regularization

Real image

With semantic regularization

Without semantic regularization

Real image

With semantic regularization

Without semantic regularization