FashionModel: Mapping Images of Clothes to an Embedding Space

Sarah Wooders and Ryan Senanayake
MIT
77 Massachusetts Avenue, Cambridge, MA
{swooders, rsen}@mit.edu

Abstract

We create a new dataset of 373,521 images of dresses and tops, and their associated titles, colors, and descriptions. We use the dataset to train a Show and Tell model [5] to caption clothing images, which performs substantially better than models trained on the MS COCO dataset [3] for this problem. We then use this model to get the image embedding vector for all images in our dataset. We can now manipulate the vectors in this space to find clothing with specific characteristics. Finally, we present two methods for finding the image embedding vector for a given caption, which allows us to manipulate these image embedding vectors using text captions.

1. Introduction

There are thousands of retail clothing store online, offering millions of items which are listed with their images and descriptions. Searching through these items can be a challenge, as many listed items have descriptions with very little information about the item. In order to improve descriptions of online clothing items, we train a Show and Tell model for image captioning on a clothing dataset we created through scraping for clothes online. In addition to this, using our image captions, we create a system for finding images similar to a given image, with an added or subtracted feature. For example if someone is shopping online and finds the perfect dress, except it has stripes instead of a floral pattern, they can query the image - ”striped” + ”floral” to find what they are looking for.

2. Related Work

The Deepfashion project has trained models to label clothing characteristics using image segmentation and classification. [7] The Deepfashion dataset has over has over 800,000 richly annotated images, many of them scraped from online retail store.

Image captioning is also a very well researched area. Show and Tell model, a neural image captioner, is one of the state of the art implementations of image captioning nueral network [6]. COCO is a large scale captioning dataset with over 200,000 labeled images.

3. Methodology

3.1. Web Scraping

We scraped the images, descriptions, titles, and colors for clothing items listed on ShopStyle.com, which aggregates retail items for over a thousand different shopping websites. In total we collected 373,521 images with 40,081 words in vocab. The images consisted of 20% mens shirts, 6% mens sweaters, 37% dresses, and 37% womens tops. We also built a web scraper for Amazon.com, but it was much slower so we ended up just using ShopStyle.

To create the captions that we trained on, we concatenated the text for the description, title, and color for a given item, then removed any HTML tags, capitalization, punctuation, and single letter words. We also scaled down the images to 25% of the original image, so the images would be closer to the size of images in the COCO dataset, and training would be faster.

3.2. Training Show and Tell Model

The Show and Tell Model [5] is built to learn a human-readable caption from an image. This model consists of an Inception v3 encoder [4] that produces an image embedding vector and an LSTM to turn this vector into a human-readable description. We modified this model slightly to use our dataset with an increased vocab size and then trained it on hyperparamaters based on the Show and Tell Model paper. We also trained this same model on the MS COCO dataset [3] as a reference.

We used the following hyper-parameters for training: batch size = 32, lstm dropout = .7, optimizer = SGD, learning rate = 2.0, learning rate decay factor = .5 every 8.0 epochs, inception learning rate = .0005, clip gradients = 5.0. We trained the LSTM for 500k steps holding the Inception weights constant. We then spent 100k steps where we also trained the Inception encoder so that our embedding space would reflect the way the captions describe the images. We reached a similar convergence to when this model was trained on MS COCO. You can also see in Figure 5 that very little overfitting was observed. This was something we worried about as the original Show and Tell paper described this a significant challenge for them [5].

3.3. Image Embedding Vector Space

As the LSTM is able to take an image embedding vector and produce a human-readable caption, we hypothesized that this vector must encapsulate the aspects of the image that we care about. We started by producing embedding vectors for every image in our training set. This is easily accomplished by running each image through the Inception v3 encoder. Once we had these vectors, it was possible to find similar images for a given image in the validation set via cosine similarity, as seen in Figure 3. However to accomplish our goal of subtracting English descriptions of properties, we also need to find an image embedding vector for a given caption. We attempted two approaches to this problem, which we describe below.

3.3.1 Backpropagation of Caption into Image Embedding Vector

Normally we back-propagate the error with respect to the weights of a neural networks, instead we calculated the error with respect to the image embedding vector and performed updates with respect to this input. We modified the Tensorflow model to replace the start of the model with a variable and instructed the optimizer to only modify this vector. We also had to use a different set of hyper parameters to learn image embedding vectors. We started with the same set of hyper parameters as used in captioning model (described in Section 4.1), but used the Adam Optimizer [1] with a learning rate of .1 and a decay factor of .9 every 500 steps. We also made sure to remove the LSTM dropout. To verify that the resulting vector really represented the caption, we ran inference on these vectors and obtained the same caption that we trained the vector from.

3.3.2 Doc2Vec Model

We also experimented with a doc2vec model to translate the output captions into vectors. We used the English Wikipidia DBOW pretrained model and implementation to create vectors for image captions and features [2]. We generated similar images using the cosine similarity of caption vectors, as seen in Figure 2.

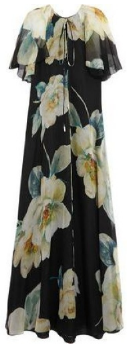

Figure 1. Image from the validation set of our dataset. Captions for this image are provided in Section 4.2

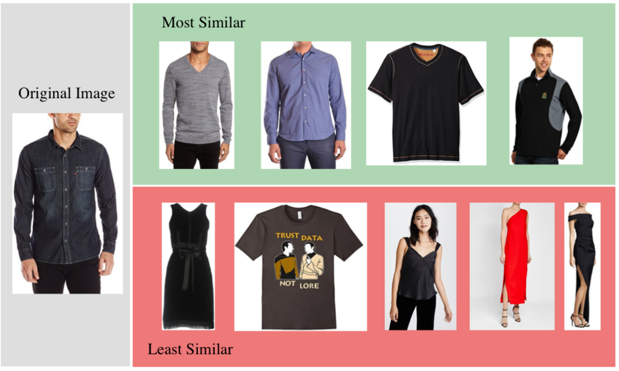

Figure 2. Results for most similar and least similar images using the cosine distance between the doc2vec vectors of the captions.

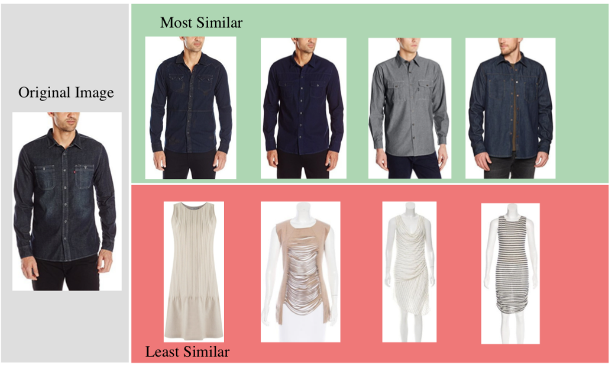

Figure 3. Results for most similar and least similar images using the distances between the image encodings in the Show and Tell Model LSTM.

4. Results

4.1. Clothing Image Captioning

We compare the result of our captioning model to the Show and Tell model trained on the MC COCO dataset, the model trained on our own dataset, and the original caption

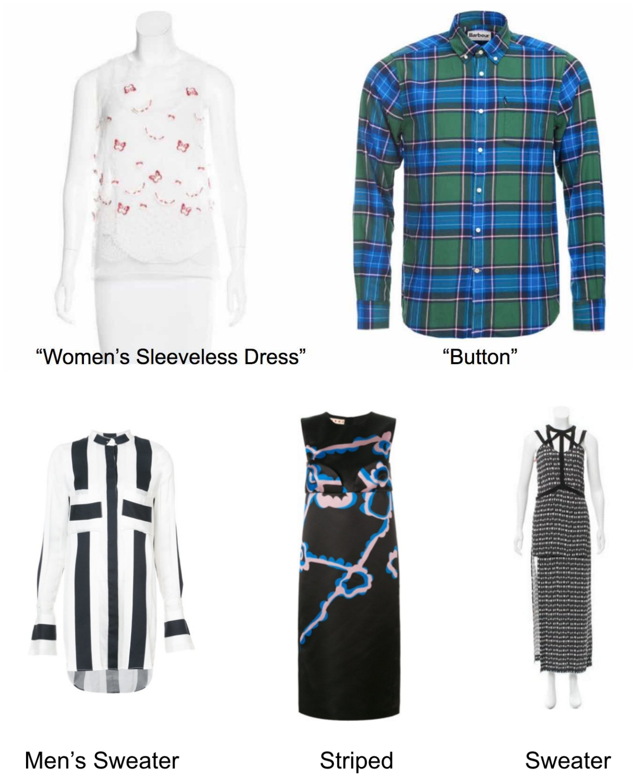

Figure 4. Images for given features.

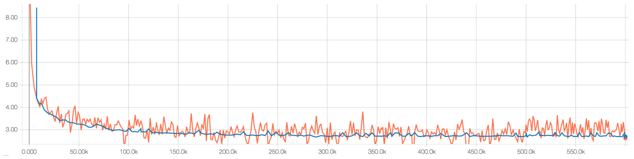

Figure 5. Cost vs step for training (orange) and validation (blue). Produced by running procedure discussed in Section 4.1

for the image shows in Figure 1:
FashionModel: floral print maxi dress multicolour multicoloured silk from emilio pucci
MS COCO: dress
ShopStyle: Floral-print silk-chiffon gown
Our caption was generated by using a beam search of beam width 3 where each iteration of running the LSTM generates a new word. Right now we are not penalizing longer length captions, which is why ”from emilio pucci” appears in the caption. In the future, we plan to experiment with adding a length penalty. Otherwise, our model's caption identifies all the characteristics in the original caption, such as ”floral” and ”silk”, and even identifies an additional characteristic, ”maxi dress” (describing the length of the dress), which was not contained in the original caption. Hence our model is able to provide additional information about a given clothing image, beyond its original description.

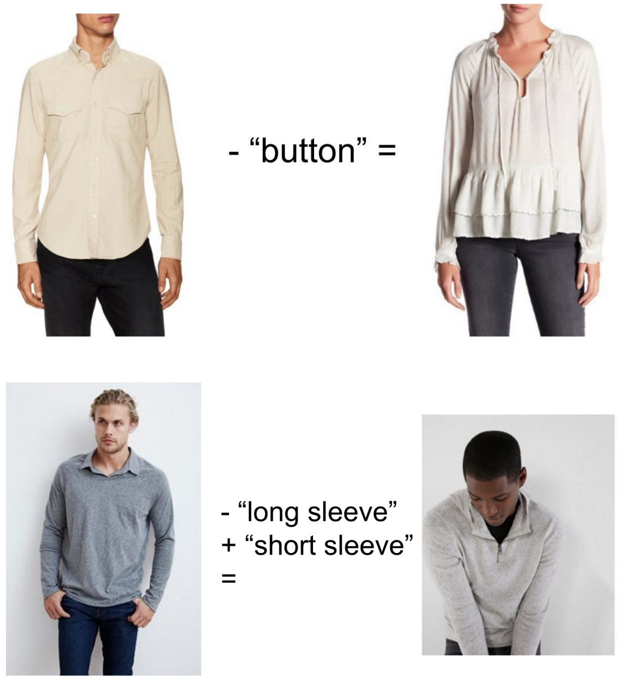

Figure 6. Resulting image nearest to the result of subtracting vector for ”button” from embedding vector of left image.

4.2. Embedding Vectors for Captions

We found that the representations of the images in the image embedding vector space were much better for determining if images were similar or not than the doc2vec vectors of the image captions. We can observe this by comparing Figures 2 and 3. Figure 2 shows the result of finding similar images by measuring cosine distance between the doc2vec of the image captions. Although there are clearly more similarities between the top images and the original image than with the bottom images, overall the results are not very consistent. However in Figure 3, we see that the image embedding space vectors work very well for finding similar images, with the top result being near identical to the original image. The most similar images are all dark colored, grey or blue, buttoned, collared, long-sleeved, men's shirts. The least similar images are all white, women's dresses, which would match what we would expect the ”opposite” of the original image would be. Hence we conclude the image embeddings are better representations.

4.3. Manipulating Vectors in Embedding Space

As the embedding vector space was much better for detecting image similarity than the doc2vec method, we focused on the image embedding vector space for adding and subtracting vectors representing features from images. Figure 4 shows the result of generating images with embeddings closes to that of a given feature. We also tried generating images that were similar to a given image ”minus” some feature. Figure 6 shows one of our results for this. Although there were some examples where this mechanism works very well, our results are very inconsistent.

5. Evaluation

This was a difficult project to evaluate due to the lack of quantitative metrics and also given that we created our own dataset so we could not compare our model against other models that were trained on the dataset.

5.1. Dataset Quality

We were able to scrape a dataset larger than MS COCO in a 40 minutes. MS COCO is one of the most prevalent datasets used for image captioning and so we feel like this is was an important achievement. Even though the MS COCO captions were better formatted, they were also much simpler and didn’t describe complexities in the images that we needed. Given similar levels of convergence and no overfitting, we feel that dataset was of a sufficient quality to be used for training captioning networks.

5.2. Captions

We had some errors in captions, but they typically were at the end of the caption, which suggests they originated from not penalizing captions of longer length. Other than errors in the last few words, our captions were very accurate and provided additional information beyond what the original descriptions did.

Show and Tell presents their evaluation in terms of unbiased humans who classified how relevant each caption is to an image. This was something that we could not hope to accomplish in such a short time period, but it is something that we would hope to do in the future to evaluate the quality of our captions. In terms of finding images similar to a given caption or manipulating vectors in an embedding space, we feel that these images were not sufficient and more work would be needed to improve the quality of these results.

5.3. Feature Vectors

Our results for finding images similar to a given caption or manipulating vectors in an embedding space were very poor, as seen in Figure 6.

6. Conclusion

Our project was successful in generating detailed, and fairly accurate captions for clothing images using the Show and Tell model on our scraped data. In addition, the image embedding vector our trained model returned for images is very good for determining how similar different images are. However we were not able to obtain strong results for adding and subtracting features. We suspect this may be because of problems with how we are adding or subtracting the vectors from the embedding vectors, or how we are generating the vectors for a given feature with the LSTM back-propagation.

7. Individual Contribution

I (Ryan Senanayake) wrote the Amazon.com scraper, which we ended up not using. I also worked on the ShopStyle scraper and performed preprocessing on the resulting dataset so that we could use it just like MS COCO. I trained both the models both on MS COCO and our own dataset and also modified the Tensorflow model to perform backpropagation through the LSTM. Once I had these vectors, I verified that these vectors were correct by performing inference on these vectors. I also created scripts to generate the image embedding vector space. Once I had this vector space, I also wrote scripts which allowed us to find similar images and used these scripts to produce all of our results.

References

[1] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 2

[2] J. H. Lau and T. Baldwin. An empirical evaluation of doc2vec with practical insights into document embedding generation. *arXiv preprint arXiv:1607.05368*, 2016. 2

[3] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. 1

[4] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015. 1

[5] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. *CoRR*, abs/1411.4555, 2014. 1, 2

[6] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):652–663, 2017. 1

[7] S. Q. X. W. Ziwei Liu, Ping Luo and X. Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1