Do You See the Pink Dog? Image-Tagging with Multiple Adjective-Noun Pairs

Calvin Kullvén Liao
University of Gothenburg
LT2318 Artificial Intelligence: Cognitive Systems HT21

Abstract

The Flickr8k captions were parsed by scaCy and filtered to generate {image, adjective-noun-pairs (ANPs)} tuples. The tuples were used to fine-tune pretrained ResNet to train models that can predict multiple ANP labels of an image. Three simple models (treating ANPs, adjectives, and nouns as separate sets of classes) and three factorisation models (training the models to separately classify adjectives and nouns then multiplying the outputs of the two sets of classes) were compared. The result showed that the factorisation models were able to output ANPs that did not appear in the training data, by combining the separate adjective output and noun output, although the correlation of the two was overlooked by the factorisation models.

1. Introduction

This project report describes building and training convolutional neutral networks (CNN) that can predict multiple adjective-noun pair (ANP) labels appropriate to an input image, including an ANP that is not present in the training data. §2 discusses the relevant previous works and the problem of labelling images with unseen ANP labels. §3 describes the dataset and the software tools that were used in the project. The data pre-processing and the architectures of the neural networks are discussed in §4. The example outputs of the trained models are shown and discussed in §5. Finally, §6 concludes with the limitations of this project and the possible future improvements.

2. Previous Efforts and the Research Question

This project is inspired by previous works of [1], [2] and [9] that trained CNNs to predict the ANP label of an input image, provided that the image is associated with one of the adjectives from an adjective-classes set and one of the nouns from a noun-classes set. This prediction can essentially be seen as finding the most appropriate ANP combination from an |A| by |N| matrix, where |A| is the number of adjectives and |N| is the number of nouns. For example, as shown in Table 1, given 3 adjectives and 3 nouns, we have 3*3 ANP combinations:

Table 1: Sets of adjectives and nouns. The possible ANP combinations are inside the orange box.

adj noun	white	black	pink
cat	white cat	black cat	pink cat
dog	white dog	black dog	pink dog
rat	white rat	black rat	pink rat

A simple approach (A) is treating each ANP as a class, but the problem is that the ANP labels in the data cannot cover all possible combinations. An image labelled with ‘pink dog’, for example, is likely rare or absent in the dataset, thus the model cannot predict a ‘pink dog’ label because it was never seen during training.

Another approach (B) is considering the adjectives and nouns as two different sets of classes. In this way, we can train separate models to predict adjective-classes and noun-classes, and then combine the two to predict the ANP combination. The problem to solve would thus be: given an ANP, e.g., ‘pink dog’ that is absent in the training data, how can a model correctly label an image of pink dog?

[9] experimented with the ANP classification problem mentioned above. The experiment used the SentiBank dataset to train and compare five CNNs: ANP-Net, Fork-Net, Fact-Net, and two LSTM-Nets (which predict the adjective preceding a noun, or vice versa). The ANP-Net is the previously mentioned approach A, while Fork-Net extracts image representations with a VGG network and then feeding the VGG output through two separate fully-connected (FC) layers that predict the respective adjective and noun; Fact-Net is similar the Fork-Net but factorises the two respective outputs of adjective-FC and noun-FC to get a matrix multiplication which is used to predict the final output. The evaluation found that while the ANP-Net had the highest accuracy when predicting seen ANPs (i.e., those in the dataset), it cannot predict unseen ones, whereas Fact-Net was better than Fork-Net at predicting unseen ANPs.

[1] adopted the similar approach by using an MVSO subset and training two separate AdjNet and NounNet, each of which is a classifier that predicts one label out of a fixed set of adjectives or nouns. The two networks then became the blocks of a larger network model that fuses the features extractions (either the “semantic features”: the final logits-and-Softmax outputs, or the “visual features”: the intermediate representations from the pre-logits layer) of AdjNet and NounNet with added downstream FC and Softmax to predict ANP classification. The result showed that fusing the “visual features” instead of the “semantic features” achieved superior performance; the model using “semantic features” fusion though, was even worse than the baseline ANP classifier model. (Note that the models in [1] only need to predict the number of seen ANP classes, instead of |A| * |N| classes.)

[2]presented a continuation of [1], and compared an "ANPNet" (which fused the Softmax outputs of independently trained AdjNet and NounNet, each of which is a fine-tuned ResNet model that predicts a label out of |A| or |N| classes) and a "Non-interpretable" model which fused the respective intermediate representations of AdjNet and NounNet. The setup of ANPNet vs. Non-interpretable is similar to the "semantic" vs. "visual" models in [1] which also aimed to observe the difference between learning from Softmax probabilities and learning from intermediate image representations. The result showed marginally better accuracy of the Non-interpretable model.

A difference between the previous works and this project, however, is that the models in [1], [2] and [9] predict only one ANP label, whereas the models in this project predict multiple labels that are relevant to the input image. In other words, an image can be assigned more than one labels by the models in this project. This multi-label classification approach is further discussed in §4.

3. Dataset and tools

Flickr8k: The papers reviewed in §2 used Visual Sentiment Ontology (VSO) dataset or its variants. However, the dataset is over 50GB. For efficiency's sake, this project used a modified version of the Flickr8k dataset introduced in [5]. The modified dataset contains 8091 images, each of which has 5 captions. Hence there are a total of 40455 <image, caption> pairs associated with 8091 images. The ANPs were extracted by parsing the caption texts, as further described in §4.1.

spaCy: To extract ANPs from the caption texts, the spaCy parser [4] was used. It is an open-source Python library for natural language processing. One of the features used in this project is the part-of-speech tagging of word-tokens of a given text. The details about the parsing for ANPs is elaborated in §4.1-4.2.

Pytorch, Torchvision, and ResNet: The PyTorch library presented in [7], [8] is a power for deep learning and especially useful for computer vision and natural language processing tasks. Torchvision is an open-source machine vision package presented in [6] and includes image-processing algorithms and also CNN models available for initialising from the scratch or using pre-trained models. This project used the pre-trained ResNet101 model which is a variant of the architecture presented in [3] with 101 layers. The model was originally trained to predict a label out of a set of 1000 classes. This project fine-tuned the model by replacing the last layer to predict the customised number of classes. The fine-tuning is detailed in §4.

4. Methodology

4.1 Parsing caption to get ANPs

Since the original Flickr8k dataset only contains caption texts for images, the spaCy parser was used to parse the captions for ANPs by looking for tokens that is tagged as a noun. A noun is paired with its preceding token if the latter is tagged as an adjective, otherwise the noun is paired with a "-" token to denote empty preceding adjective. An example is shown below in Table 2.

Table 2: Example of parsing a caption text for ANPs, either including or excluding nouns without a preceding adjective.

Caption text	Three black dogs and a tri-colored dog playing with each other on the road.
Parsed ANPs (with empty adjective class)	('black', 'dog'), ('colored', 'dog'), ('-', 'road')
Parsed ANPs (no empty adjective class)	('black', 'dog'), ('colored', 'dog')

For simplicity and to avoid having the non-differentiable "-" adjective class (since it could be annotated in any image and thus pollute the sense of "-"), nouns that lack a preceding adjective in the captions were ignored.

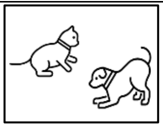
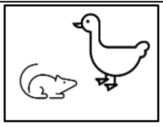
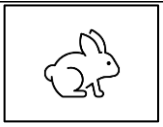
4.2 Filtering adjectives and nouns by frequency

Applying the above-mentioned spaCy parsing on Flickr8k dataset extracted 1061 adjectives and 3915 nouns. To get the model to generalise during training and to limit the number of classes, the parsed ANPs were filtered by only keeping those that contain adjectives and nouns that appear at least 150 times, therefore reducing the number of classes to 26 adjectives and 101 nouns (see Appendix A). The images were also filtered to exclude those annotated with adjectives or nouns appearing fewer than 150 times. The filtered adjectives and nouns make up 704 ANPs in the filtered data, which is far fewer than the 2026 possible combinations. After this filtering, the pre-processed dataset contained 6057 ANP-annotated images, which was further shuffled and split to 4846 training and 1211 validating images.

4.3 Fine-tuning ResNet to predict multiple labels

To enable the model to predict multiple labels, the ground truth labels of an image were encoded as a multi-hot vector: a 1D vector where the length = the number of classes and populated with 1 or 0 to denote whether a class is annotated. Table 3 shows some examples of multi-hot encoding.

Table 3: Examples of labels and multi-hot encoding. (The illustrations are not from the dataset images.)

Set of 5 classes: {cat, dog, rat, rabbit, duck}		
 {cat, dog} [1, 1, 0, 0, 0]	 {rat, duck} [0, 0, 1, 0, 1]	 {rabbit} [0, 0, 0, 1, 0]

The pretrained ResNet101 model is fine-tuned by replacing the original final fc layer (linear transformation to 1000 features) with a linear transformation to the number of classes and a Sigmoid function. The output of the model and the multi-hot encoding are given as the input-and-target of Binary Cross Entropy (BCE) loss function to compute the loss and adjust the model parameters. §4.4 further discusses the models and training.

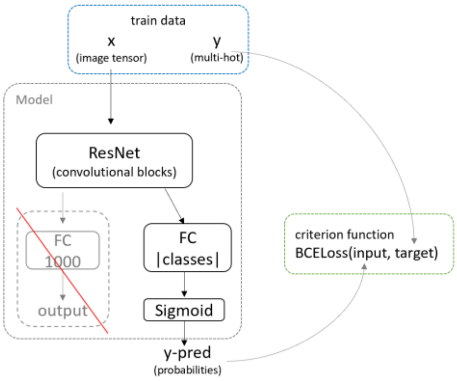
4.4 Model architectures

This section briefly describes the different model architectures that were experimented in this project. The outputs of the models are discussed in §5.

4.4.1 Simple multilabel-tagger models (ANP-tagger, Adj-tagger, Noun-tagger):

These models treat the ANPs, adjectives, or nouns as sets of independent classes, and predicts the probability of classes given an image input. Figure 1 illustrates the architecture and how the loss is computed for backpropagation. In the case of the ANP-tagger model, each ANP is treated as a class and the relationship between the ANP and the adjective/noun it contains is not considered. Problems with this ANP model are that there will be unnecessary combinations. It also cannot predict an ANP unseen in the training data, e.g., “pink dog”.

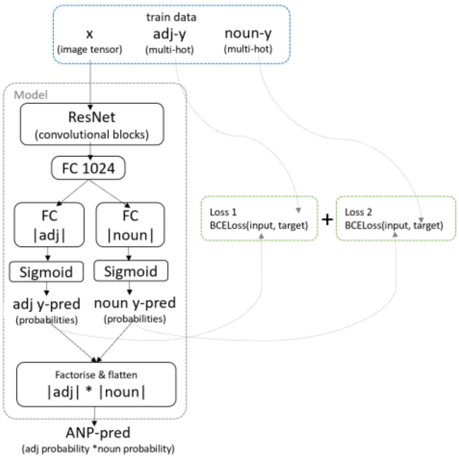
Figure 1: Architecture of simple multi-label classification. The FC-1000 layer in the pre-trained ResNet was replace with FC-number-of-classes.



4.4.2 Factorise- adj-and-noun model:

This model trains to predict the respective probabilities of adjectives and nouns, then cross-multiplies these two sets of probabilities to get the probability of each ANP combination, as shown in Figure 2. This model only learns from the separate adjective- and noun-labels and does not consider the ANP labels. The probability of each ANP in the model output is the product of the adjective probability and noun probability.

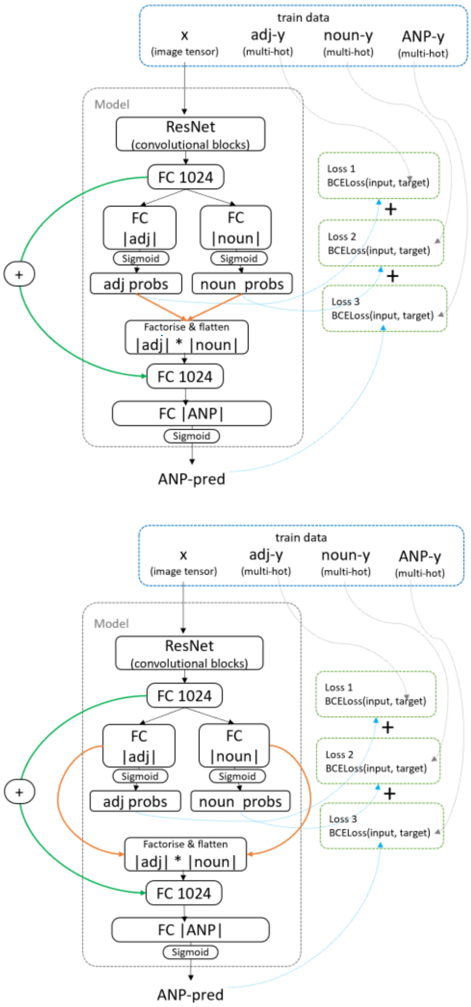
Figure 2: Architecture of cross-multiplying the adjective and noun probabilities.



4.4.3 Factorise-plus-ResNet models:

The models consider the ANP labels for the backpropagation during training, as shown in Figure 3. Another FC layer is added after the adjectives-and-nouns factorisation and added to the representations from ResNet (the green line in Figure 3), then a final FC to the number of ANP classes and Sigmoid function outputs the ANP probabilities. The different between the two models is that one factorises the Sigmoid outputs while the other factorises the intermediate representations (the orange lines in Figure 3).

Figure 3: Architectures of cross-multiplying the adjective and noun representations or probabilities, then adding the ResNet representations.



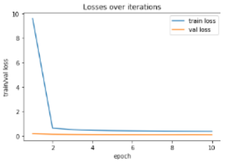
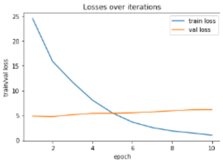
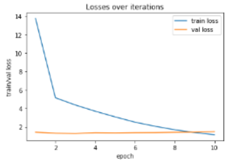
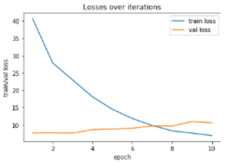
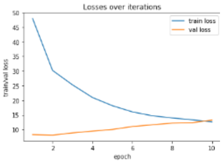
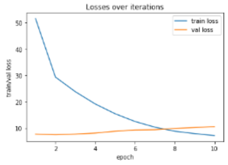
5. Results

5.1 Train and validate losses

The models from §4.4 were trained with the filtered data from §4.2 over 10 epochs. Table 4 shows the train and validate losses. The losses of the simple multi-label classification models from the first row show that despite having more classes (101 nouns vs 26 adjectives) to tell apart, the noun-tagging model had steeper descending train losses and lower validate losses. This could be owing to that ResNet was pre-trained to classify objects, and that adjectives, as quality attributes, were more difficult to generalise than nouns which have distinctive characteristics in images. The second row shows the factorisation models.

The factorise-ANP model is simply the summation of adj- and noun- taggers since it learns two sets of classes simultaneously. Factorise-and-sum models showed similar statistics but factorising the Sigmoid probabilities showed steeper-descending train losses and lower validate losses.

Table 4: The train and validate losses of different models

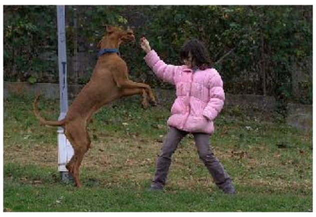
ANP-tagger	Adj-tagger	Noun-tagger
		
Factorise-adj-and-noun	Factorise-reps-plus-ResNet	Factorise-Sigmoids-plus-ResNet
		

5.2 Models' outputs with images containing seen and unseen ANP combinations

Table 5 shows the outputs given an image from the validation data. The ANP-tagger detected the objects, but the modifying adjectives are not all correct. The respective Adj- and Noun- taggers captured additional information that the ANP-tagger did not, such as "pink" and "girl". On the other hand, the factorisation models were able to recognise the two sets of classes and combine them. The Factorise-adj-and-noun model, which outputs the products of probabilities of adjectives and nouns, only outputs all possible combinations. The Factorise-Sigmoids also had the same issue. The output of the Factorise-Sigmoids model is the most descriptive of the image content.

More example outputs are shown in Table 6, and it can be observed that the models were better at recognising nouns, and predicted ANPs with the same noun and varying adjectives, emphasising the fact that the ResNet-based models are more adept at classifying objects than at attributes.

Table 5: Models outputs given an image of a girl in pink jacket with a large brown dog.

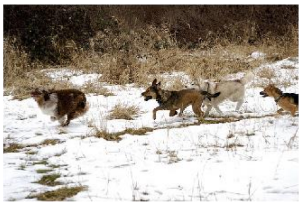


Gold labels: ('brown', 'dog'), ('large', 'dog'), ('pink', 'jacket')

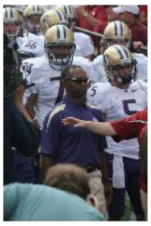
Simple models
ANP-tagger: ('black', 'dog'), ('white', 'dog'), ('brown', 'dog'), ('grassy', 'field')
Adj-tagger: 'pink', 'little', 'brown', 'grassy'
Noun-tagger: 'girl', 'dog'

Factorisation models
Adj & Noun: ('pink', 'girl'), ('pink', 'dog'), ('brown', 'girl'), ('brown', 'dog')
Reps + ResNet: ('little', 'girl'), ('brown', 'dog')
Sigmoids + ResNet: ('brown', 'dog'), ('brown', 'shirt'), ('large', 'dog'), ('large', 'rock')

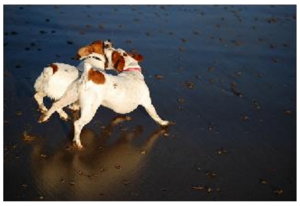
Table 6: Models outputs given images with seen ANPs.



Gold: ('snowy', 'field')
Simple models
ANP: ('black', 'dog'), ('white', 'dog'), ('brown', 'dog'), ('small', 'dog'), ('large', 'dog'), ('tan', 'dog')
Adj: 'black', 'brown', 'snowy'
Noun: 'dog', 'field'
Factorisation models
A*N: ('black', 'dog'), ('brown', 'dog'), ('brown', 'field'), ('young', 'dog'), ('snowy', 'dog')
Reps: []
Sigmoids: ('snowy', 'hill')



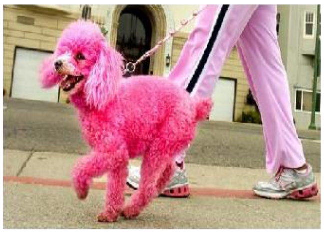
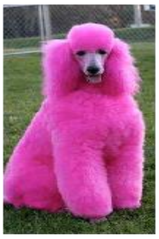
Gold: ('purple', 'shirt'), ('blue', 'shirt')
Simple models
ANP: ('young', 'man')
Adj: 'blue', 'purple'
Noun: 'shirt'
Factorisation models
A*N: ('blue', 'hat'), ('blue', 'shirt'), ('blue', 'jacket'), ('blue', 'helmet'), ('purple', 'hat'), ('purple', 'shirt'), ('purple', 'helmet')
Reps: []
Sigmoids: ('blue', 'shirt'), ('purple', 'shirt')



Gold: ('white', 'dog'), ('brown', 'dog')
Simple models
ANP: ('black', 'dog'), ('white', 'dog'), ('brown', 'dog'), ('small', 'dog'), ('tan', 'dog')
Adj: 'white', 'brown'
Noun: 'dog'
Factorisation models
A*N: ('white', 'dog'), ('brown', 'dog')
Reps: ('white', 'dog')
Sigmoids: ('white', 'dog'), ('brown', 'dog')

When giving images not in the dataset as inputs, as shown in Table 7, the models had to rely on the generalisation learnt from the training data and make predictions. The images are of pink dogs, and the ANP-tagger could not output the label because such combination did not appear in the training data. However, the Adj- and Noun- taggers were able to tell the existence of *something pink* and *something dog-like*. The factorisation models utilised the combination of the separate Adj- and Noun- models and a prediction of "pink dog" was output based on the product of a high probability for "pink" and a high probability for "dog". Nonetheless, the other ANP labels are not applicable to the images since the models only classify two separate things, rather than learning the relations (e.g., something that is dog-like *and* is pink).

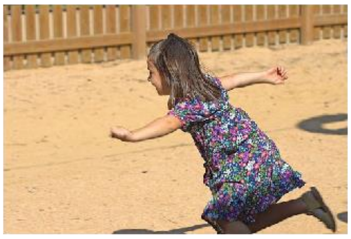
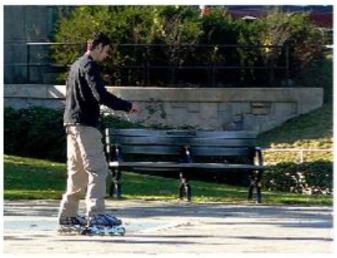
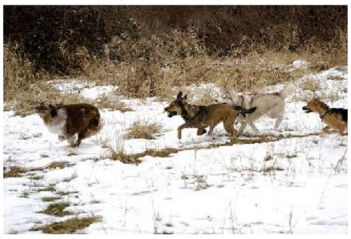
Table 7: Models outputs given images with unseen ANPs

 **Gold**: None **Simple models** ANP: ('little', 'girl'), ('young', 'girl'), ('young', 'boy') Adj: 'pink', 'little', 'young' Noun: 'dog' **Factorisation models** A*N: ('pink', 'dress'), ('pink', 'girl'), ('pink', 'shirt'), ('little', 'dress'), ('little', 'girl'), ('little', 'shirt'), ('young', 'girl'), ('young', 'shirt'), ('red', 'girl'), ('red', 'shirt'), ('purple', 'girl') Reps: ('little', 'girl'), ('young', 'girl') Sigmoids: ('little', 'girl'), ('young', 'girl')
 **Gold**: None **Simple models** ANP: ('black', 'dog'), ('white', 'dog'), ('brown', 'dog') Adj: 'pink', 'little', 'white', 'young', 'purple' Noun: 'dog' **Factorisation models** A*N: [('pink', 'dress'), ('pink', 'girl'), ('pink', 'dog'), ('pink', 'hat'), ('pink', 'shirt'), ('purple', 'girl'), ('purple', 'shirt')] Reps: ('little', 'girl'), ('young', 'girl') Sigmoids: ('pink', 'shirt'), ('little', 'girl'), ('young', 'girl')

5.3 Problem with quantitative evaluation

[1], [2] and [9] used K-best accuracy to evaluation the single-label output of the models by checking whether the gold label is in the model's top-K predictions (in order of highest probability), but the approach is not applicable to the multi-label models in this project. Since the 'right answer' is open-ended, a mismatch between the ground truths and the predicted labels does not necessarily mean an incorrect prediction. Another problem is that many of the classes in the data are semantically and visually similar and even overlap. As seen in Table 7, a "little girl" is synonymous to a child who is young; the middle image got a "young man" output instead of the clothing items and the third image got output for dogs instead of the scene/background. Hence, a different approach must be devised to evaluate the models. A potential evaluation method would be using WordNet dataset to group hypernym and hyponym labels (e.g., "girl" being a hyponym of "child"). Another possible way might be sampling a smaller set of images and predicted labels and ask human evaluators to rate the appropriateness of the predictions.

Table 7: Examples of mismatched gold vs predicted labels


Gold: ('little', 'girl') ANP: ('little', 'girl'), ('young', 'girl'), ('young', 'child')

Gold: ('black', 'shirt'), ('brown', 'pant') ANP: ('young', 'man')

Gold: ('snowy', 'field') ANP: ('black', 'dog'), ('white', 'dog'), ('brown', 'dog'), ('small', 'dog'), ('large', 'dog'), ('tan', 'dog')

6. Conclusion

Future improvements

In the models of this project, the adjectives and nouns were treated as two separate classification tasks, but in rea life, a human observes objects and associates the adjective-noun relation in a different manner. For instance, detecting a dog in a picture, then further analysing the dog's colour, size, and/or other attributes. Alternatively, one may see 'something pink' and then decide whether that pink object is a flower, a sweater, or a dog. Applying attention layers to the NN model will help mimicking such a process of determining the adjective-noun relation.

The current project used pre-trained ResNet to build fine-tuned models, it would be worth experimenting with other state-of-the-art convolutional neutral network models available from Torchvision, e.g., EfficientNet and observe if the performance improves. To help the model to see a wider variety of data and generalise the representations instead of fitting the training data, an augmentation of the training images can be applied by random rotation and flipping.

Summary

This project parsed the Flickr8k image captions and extracted ANPs to fine-tune ResNet-based models in order to make multi-label ANP predictions. Different approaches were applied by treating ANPs as individual classes or training separate classifier models for adjectives and nouns and combining the two by factorisation. The result showed that factorising made labelling an unseen ANP possible, but the overall labelling is far from perfect. For future projects, a further exploration of evaluation methods and better ways of fusing the adjective and noun classifications to predict ANP are recommended.

References

[1] Dèlia Fernández. 2016. *Clustering and Prediction of Adjective-Noun Pairs for Affective Computing*. Master's thesis. Universitat Polit`ecnica de Catalunya, Barcelona, Spain.

[2] Dèlia Fernández, Alejandro Woodward, Víctor Campos, Xavier Giró-i-Nieto, Brendan Jou, and Shih-Fu Chang. 2017. More cat than cute? Interpretable Prediction of Adjective-Noun Pairs. In *Proceedings ofMUSA2'17, Mountain View, CA, USA, October 27, 2017*, 9 pages. DOI: https://doi.org/10.1145/3132515.3132520

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778. DOI: https://doi.org/10.1109/CVPR.2016.90

[4] Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.

[5] Micah Hodosh , Peter Young, and Julia Hockenmaier. 2013. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research 47 (2013)*, 853-899.

[6] Sébastien Marcel and Yann Rodriguez. 2010. Torchvision the machine-vision package of torch. In *Proceedings of the 18th ACM international conference on Multimedia (MM '10). Association for Computing Machinery, New York, NY, USA*, 1485–1488. DOI: https://doi.org/10.1145/1873951.1874254

[7] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. (2017).

[8] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Advances in Neural Information Processing Systems 32. Curran Associates, Inc., 8024–8035. Retrieved from http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[9] Takuya Narihira, Damian Borth, Stella X. Yu, Karl Ni, and Trevor Darrell. 2015. Mapping Images to Sentiment Adjective Noun Pairs with Factorized Neural Nets. arXiv: 1511.06838

Appendix

A. The filtered (appearing at least 150 times in the data) adjective and noun classes

adjectives (26 classes)	pink, wooden, little, black, white, brown, small, young, orange, red, grassy, green, blue, several, yellow, large, tan, snowy, blonde, rocky, purple, colorful, other, old, asian, haired
nouns (101 classes)	child, dress, girl, building, dog, road, street, front, hand, grass, man, bench, park, ground, hat, glass, rope, fence, beach, ball, water, head, side, boy, wall, city, shirt, jean, rock, tree, collar, snow, field, picture, person, group, people, face, car, mouth, toy, air, midair, woman, lake, swimming, couple, baby, body, sand, jacket, pant, stick, kid, hill, camera, top, boat, crowd, guy, jump, skateboard, trick, skateboarder, river, ocean, hair, arm, area, background, pool, short, mountain, yard, tennis, helmet, bike, dirt, bicycle, race, ramp, toddler, lady, sidewalk, outfit, wood, path, coat, suit, rider, soccer, wave, game, team, uniform, swing, adult, sunglass, player, football, track