Integrating Visual and Linguistic Information to Describe Properties of Objects

Calvin MacKenzie
The University of Texas at Austin

Abstract

Generating sentences from images has historically been performed with standalone Computer Vision systems. The idea of combining visual and linguistic information has been gaining traction in the Computer Vision and Natural Language Processing communities over the past several years. The motivation for a combined system is to generate richer linguistic descriptions of images. Standalone vision systems are typically unable to generate linguistically rich descriptions. This approach combines abundant available language data to clean up noisy results from standalone vision systems.

This thesis investigates the performance of several models which integrate information from language and vision systems in order to describe certain attributes of objects. The attributes used were split into two categories: color attributes and other attributes. Our proposed model was found to be statistically significantly more accurate than the vision system alone for both sets of attributes.

1. Introduction

Language is used as a medium for communication, and this communication is performed in various ways. Being in such a digital age, much of this communication happens online. This abundance of language on the web allows complex language models to be built which represent natural language very well. Additionally, many images and videos exist which serve as datasets for experiments in computer vision. This volume of data makes it possible for robust systems to be built and enhanced.

One import research area in natural language processing (NLP) that these language models assist in is natural language generation. Natural language generation is the task of generating natural language sentences - typically using some sort of corpus to learn about language.

The ability to describe objects using their properties dramatically increases the visual stimulation of a sentence. Authors use as much detail as possible to immerse the reader in the story being told. The same can be said when providing a caption or description for an image or video. Whenever more detail is used, the viewer can understand more about what is captured.

Thus, an important component of natural language generation is the ability to use modifiers, such as adjectives, to describe the properties of objects in generated sentences. These modifiers enhance the generated sentences and provide more detail and context.

Another important research problem tangential to this topic is the task of feature or attribute detection within computer vision. This task involves identifying features of objects within images or videos. Models can be trained on datasets annotated with these features and then tested on new images which also contain these features. These models have proven to work well [1, 3].

The intersection of these two fields, systems which use both vision and language, is an active research area [6, 7]. Combined models not only use vision techniques to determine features of objects, but confirm their existence in language. Without the assistance of language, vision systems can produce erroneous features which are typically not attributed to certain objects.

This paper attempts to bridge the gap between results in computer vision and linguistic knowledge from a variety of corpora by creating a new model which leverages the two. By utilizing linguistic information, we can aid the vision system. We propose a model which uses a linear interpolation of vision and language in order to predict attributes of objects.

The vision model consists of probability distributions taken from the image which determine the probability of a specific attribute in the image. The language model consists of probability distributions taken from the corpora which determine the probability of a specific attribute modifying an object in language. The vision model does not always perform perfectly, and is often noisy due to lack of using advanced computer vision techniques, such as image segmentation. Whenever these circumstances arise, the language model helps to determine whether the attributes found by the vision system do in fact appear in language. For example, the red bus shown below in Figure 2 is misclassified as a black bus by the color vision model. However, the language model knows that the probability of a black bus is much less than that of a red bus, so the bus being red is more likely.

With this new model, we have shown an increase in accuracy over a generic, out-of-the-box object and attribute detector and a system that only uses linguistic information. The final accuracy of our color attribute model is 62%, a 6% increase from the pure vision system and a 15% increase from the pure language system. The final accuracy of our other attribute model is 62%, an 6% increase from the pure vision system and a 35% increase from the pure language system.

Our model was designed to aid other language-vision systems which only produce a sentence containing a subject, verb, and object. By adding adjectives which describe the object being shown, the generated sentences will contain more vivid imagery and help describe the scene in greater detail, providing a richer experience for the reader.

2. Related Work

The task of generating natural language descriptions for images and videos has become popular recently. Most of this work has been performed on static images. Farhadi et al. [4] describe a system which can produce a sentence given an image. Similarly, Yao et al. [12] describe a system that takes an image and is able to produce a text description by using finding the semantic

representation of the image and combining that with a knowledge base. Laptev et al. [8] describe a system that is able to learn human actions from movie clips and to label other clips with the appropriate action.

There is a growing body of recent work in the field of combining vision and language. None of the aforementioned systems use linguistic knowledge to aid interpretation. Researchers have realized that the intersection of vision and language is a fertile ground for new development. The most comparable work comes from Kulkarni et al. [7]. The system created by Kulkarni et al. generates descriptions for images using a combination of language and vision. The corpora used for creating their language model comes from a combination of image descriptions from Flickr and search results from Google Search. By using a conditional random field, a labeling for the image is predicted and a sentence is generated which also includes attributes about the object. The performance of their system was not compared against a pure vision or pure language model. The system was also designed to include prepositions in order to determine the relationship between the objects found in the image, but did not always describe the attributes of objects in the generated sentence.

Krishnamoorthy et al. [6] describe a system that produces a subject, verb, and object (SVO) triplet given a YouTube video. For the subject and object specifically, the vision model is able to analyze individual static frames and use an object detector to correctly identify what is in the video. Their system combines a language model with results from the vision model in order to generate this triplet. Finally, using another language model, the SVO triplet is converted into an actual sentence which describes the video. By utilizing language, the model was able to better predict the SVO triplet for describing the video.

This paper builds on the work of Krishnamoorthy et al. By adding adjectives to the existing SVO triplet, the sentences generated will be more complex and provide greater detail. Since adjectives of objects are generally not dependent on the entire video, the problem can be reduced to analyzing a single image rather than a video. This simplification allows existing systems to be used and applied to a more complex problem.

3. Method Overview

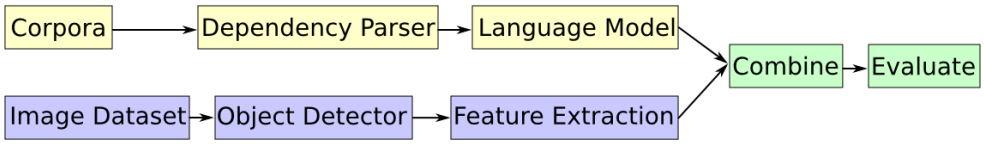

Figure 1: Pipeline for system

Figure 1 shows the pipeline for the system. Since the goal of the model is to bridge information from vision and language, the results for the vision and language systems must be independently calculated before being combined. In sequential order, the tasks are **1)** using object detectors to detect the object within an image, **2)** extracting information about these objects using attribute detectors, **3)** extracting information about these objects from previously mined and parsed corpora, and finally **4)** combining this information to produce attributes. The attributes used in this experiment are separated into two categories: the color attributes and the other attributes. The following sections describe this process in more detail and showcase intermediate results.

3.1. Image Dataset

The PASCAL Visual Object Classes Challenge is a popular image dataset built for object recognition and related tasks. The image dataset in this experiment came from the 2008 PASCAL images and was curated by Rashtchian et al. [2, 10]. The objects represented in the PASCAL set are: person, bird, cat, cow, dog, horse, sheep, aeroplane, bicycle, boat, bus, car, motorbike, train, bottle, chair, dining table, potted plant, sofa, and tv. This dataset contained fifty images per object from the PASCAL set, for a total of one thousand images. The 'person' object was excluded due to clothing obfuscating actual attributes, leading to nine hundred and fifty images and nineteen objects or categories. Figure 2 shows an unmodified image used for the bus category.

Figure 2: Initial image of bus

Figure 3: Bounding box of bus

Next, object detectors from Girshick, Felzenszwalb, & McAllester [5] were used. These detectors were pre-trained for use on the PASCAL dataset, so a detector existed for each of the PASCAL objects. Since the system is built on prior work where the objects have already been detected, only the specific object detectors were run on the images containing that particular object. Since the object in the image was already known, the specific object detectors were run on the images containing that particular object. Gold standard bounding boxes were not used. The object detector bounding box for each image was marked and also cropped for use in the pipeline. Since the images were already separated into categories, determining the most likely object in each image was not necessary. The bounding box found from the object detector is shown in Figure 3.

After finding the bounding box, the colors of the image were annotated by a single person and each annotation was a subset of all possible attributes used in this experiment. These color annotations were used as the gold standard set of color attributes for later evaluating this experiment. The gold standard other attributes came from the dataset curated by Rashtchian et al [10]. It was found that none of the selected attributes were included in the gold standard for the plant category, so it was excluded from all other attribute evaluation. For example, to describe the image shown in Figure 2, the color attributes 'red' and 'blue' were chosen and the other attributes were 'metal,' 'glass,' and 'shiny.'

Features were then extracted from the cropped image. First, the colors of the cropped image were reduced to one of the following eleven: black, blue, brown, grey, green, orange, pink, purple, red, white, and yellow. This process used a function described by Weijer et al. [11] which mapped each input pixel of an image to one of the eleven, reducing the color space of the image. Afterwards, the out-of-the-box other attribute detectors [3] were run on each image and the resulting scores for each attribute was recorded. Of these, the only attributes used were round, metal, plastic, wood, cloth, furry, glass, feather, wool, clear, shiny, and leather.

The reduced color image is shown in Figure 4. The last step was to extract the colors and their occurrence from the new images and create a color histogram. An example histogram is shown in Figure 5.

Figure 4: Reduced color image of bus

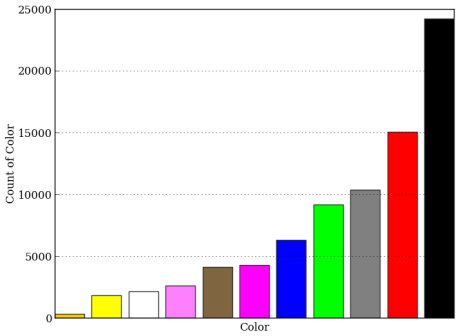

Figure 5: Color histogram of bus image

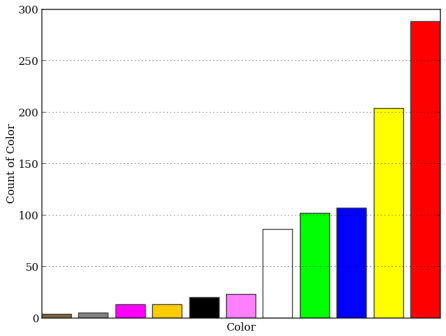

Figure 6: Color histogram of bus in language

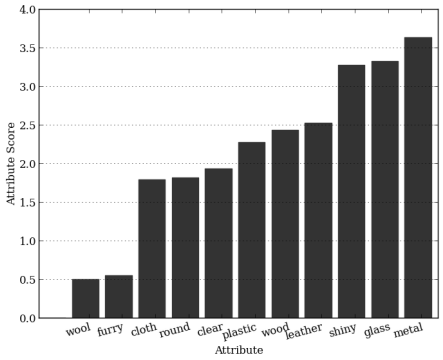

Figure 7: Other attribute scores of bus image

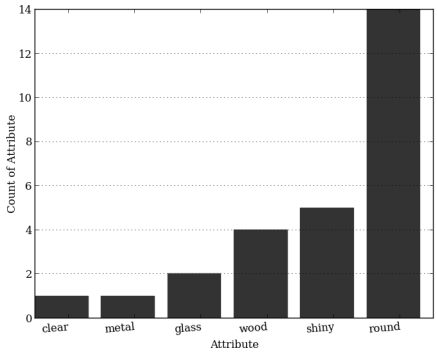

Figure 8: Other attribute scores of bus in language

3.2. Corpora

The corpora for this experiment included the British National Corpus, Gigaword, ukWaC, and WaCkypedia_EN. The sizes of these corpora (after preprocessing) are shown in Table 1. The corpora were previously parsed by the Stanford dependency parser [9]. This created a Stanford typed dependencies representation [1] for each sentence, which highlights relationships between words in a sentence. Next, the parsed corpora was searched for adjectival modifiers. If the modified object belonged to the set of PASCAL objects and the adjective matched one of the attributes used to describe objects in this experiment, it was extracted. A bigram language model was created from these pairs, which estimated the probability of an adjective given the noun. In our case, the bigram model estimated the probability of an attribute given an object. This language model consisted of probability distributions which were used later to combine with the results from the image dataset. A histogram showing the extracted color bigrams for the bus category is shown in Figure 6 and the other attribute bigrams histogram for the bus category is shown in Figure 8.

Corpora	Size of text
British National Corpus (BNC)	1.5 GB
WaCkypedia_EN	2.6 GB
ukWaC	5.5 GB
Gigaword	26 GB

Table 1: Corpora used to Mine Bigrams

3.3. Combination

As seen in Figures 4 and 5, some of the images were a little noisy. Since no image segmentation was performed, the image background and other parts which did not include the object were

[1] http://nlp.stanford.edu/software/dependencies_manual.pdf

present and contributed heavily to the color space. Image segmentation is a difficult problem and is imperfect in many cases. Most object detectors also do not perfect image segmentation and rather just output the bounding box of the detected object. By itself, the results from the vision, shown in Figure 5, would conclude that this bus was black. However, in Figure 6 we see that black buses are rare. What the language shows is that a red bus has the highest probability. For the other attributes, the results from the vision system in Figure 7 seem to be more accurate than the histogram of the language in Figure 8, because the language model believes that buses are 'round' rather than 'metal.'

By combining these two models, we harness the knowledge from the language while still relying on the vision to provide general information about what is present in the image. In this experiment, the two models are combined by using a linear interpolation of the two.

We created two linear interpolation models, one uses a global lambda which is used for all categories, and the second uses a per-category lambda, λ_c. The per-category lambda model was used to gauge the respective gains and losses from using a combination of vision and language at the category level.

In both models, the constants used for linear interpolation were determined by the training set. For the global lambda model, the value of λ was chosen to maximize the accuracy on the training set. For the per-category lambda model, the value of each λ_c was chosen to maximize the accuracy of category c in the training set. Finally, the n best colors were selected from each linear interpolated model.

The function used for linear interpolation that used a global lambda is shown in Equation 1 and the function which used a per-category lambda is shown in Equation 2

$$P_{vision}(attribute\ set|object) \times \lambda + P_{language}(attribute\ set|object) \times (1 - \lambda) \quad (1)$$

$$P_{vision}(attribute\ set|object) \times \lambda_{category} + P_{language}(attribute\ set|object) \times (1 - \lambda_{category}) \quad (2)$$

Another model was created that used the training set to learn the probability distribution of attributes for each category. This model, named the prior, was considered the top-line model for the language in an in-domain experiment. Similar to the models shown above, the prior was also combined with the vision system with a linear interpolation with a global lambda.

4. Experimental Evaluation

4.1. Methodology

To evaluate the interpolated models, accuracy, precision, and recall were measured from the test set. The formulas for calculating these values are shown in the equations below. Because each image could be annotated by a set of attributes, each guess was considered correct if the attribute it guessed belonged to the gold standard set. Precision and recall were measured by varying the

number of attributes the model output and calculating standard precision and recall between this guessed set of attributes and the gold standard set.

$$accuracy(r) = \frac{\sum_{i \in r} \begin{cases} 1 & \text{if } predicted_i \subseteq gold\ standard_i \\ 0 & \text{if } predicted_i \not\subseteq gold\ standard_i \end{cases}}{|r|} \tag{3}$$

$$precision(r) = \frac{\sum_{i \in r} |predicted_i \cap gold\ standard_i|}{\sum_{i \in r} |predicted_i|} \tag{4}$$

$$recall(r) = \frac{\sum_{i \in r} |predicted_i \cap gold\ standard_i|}{\sum_{i \in r} |gold\ standard_i|} \tag{5}$$

For example, if the color attribute model guessed 'red' for the image in Figure 2, the accuracy and precision would equal 1, while the recall would equal $\frac{1}{2}$. If the other attribute model guessed 'metal' for this image, the accuracy and precision would equal 1, while the recall would equal $\frac{1}{3}$. We hypothesized that a model using a combination of language and vision would produce a higher accuracy than a model using pure vision or pure language.

The PASCAL image dataset mentioned previously was used for both training and testing of the model. Rather than separating the dataset into two distinct categories of training and testing data, a stratified 10-fold cross validation was performed. Cross validation was chosen because it would help eliminate the issue of overfitting the model on the training data. This was accomplished by first randomizing the order of images in each category and splitting them into n distinct buckets, where $n = 10$. It was ensured that each bucket contained an equal amount of images per-category.

The model was tested on bucket i and trained on the other $n - 1$ buckets. During training, λ was varied between 0 and 1 in increments of 0.001. The value of λ was chosen to maximize accuracy of the training set. The resulting value of λ was used during testing. For a global λ, the accuracy was maximized across all categories, and when using a category specific λ_c, the accuracy was maximized for each category specifically.

4.2. Results and Discussion

The results from this experiment are shown in the following Figures and Tables. Figure 9 shows the accuracy of the training data as λ was varied between 0 and 1 for the color attributes. It can be seen here that the optimal trade-off of language and vision found during training is evenly split. Similarly, Figure 10 shows accuracy versus λ for other attributes where the optimal trade-off of language and vision is very close to pure vision.

The results in Table 2 show the accuracy of all six models tested with color attributes, the interpolated global lambda model, the interpolated per-category lambda model, the pure vision model, the pure language model, the prior model, and the interpolated global lambda with prior model. This table shows that the two interpolated models have a higher accuracy than both the pure

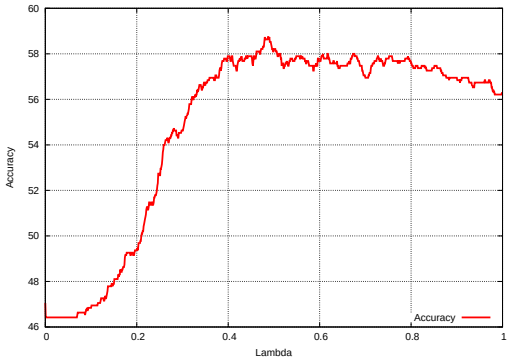

Figure 9: Graph of accuracy versus lambda for training data for color attributes

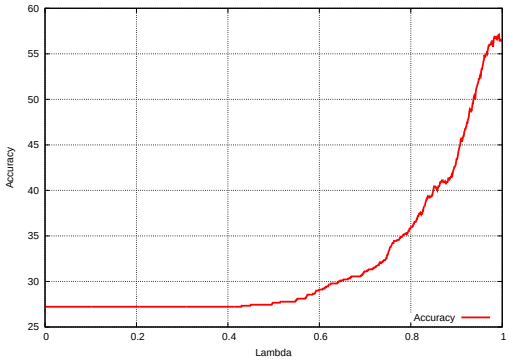

Figure 10: Graph of accuracy versus lambda for training data for other attributes

vision and pure language models, which agrees with our hypothesis. However, it also shows that the global and per-category lambda models do not perform as well as the top-line models, prior and interpolated global lambda with prior.

For the global lambda model, it was shown that the ideal value of λ leans toward a 50/50 split of vision to language, leading to a final accuracy of 57% on the PASCAL dataset. For the per-category lambda model, the final accuracy found was 62% on the PASCAL dataset. The reason for this split is that the vision model is not always able to precisely determine the correct attributes and the extra support from the language directs the model towards commonly used attributes found in language.

Model	Accuracy	Lambda
Interpolated Global Lambda	57.47%	0.52
Interpolated Per-Category Lambda	**62.21%**	-
Pure Vision	56.32%	1.00
Pure Language	47.05%	0.00
Prior	71.15%	0.00
Interpolated Global Lambda with Prior	74.31%	0.15

Table 2: Testing accuracy of each model for color attributes. Bold results were statistically significant.

Similarly, the results in Table 3 show the accuracy of all six models tested with other attributes, the interpolated global lambda model, the interpolated per-category lambda model, the pure vision model, and the pure language model. This table also shows that the two interpolated models perform better than the pure vision and pure language models. Once again, the table shows that the global and per-category lambda models do not perform as well as the top-line models, prior and interpolated global lambda with prior.

In this case, the global lambda model chooses an optimal of λ which makes it reliant on the vision model, leading to a final accuracy of 56% on the PASCAL dataset. The per-category lambda

model performs much better, with an accuracy of 62% on the PASCAL dataset. While the language model does not perform well on its own, it does boost the accuracy of the pure vision system in certain categories.

Model	Accuracy	Lambda
Interpolated Global Lambda	56.33%	0.99
Interpolated Per-Category Lambda	**62.22%**	-
Pure Vision	56.22%	1.00
Pure Language	27.22%	0.00
Prior	84.00%	0.00
Interpolated Global Lambda with Prior	84.44%	0.61

Table 3: Testing accuracy of each model for other attributes. Bold results were statistically significant.

The graph shown in Figure 11 shows an 11-point interpolated precision recall curve for the testing data for color attributes. The interpolated global lambda model is only marginally better than the pure vision model. As shown in this graph, the best curve comes from the interpolated per-category lambda model. The same conclusions can be drawn from Figure 12, which also shows an 11-point interpolated precision recall curve for the testing data for other attributes.

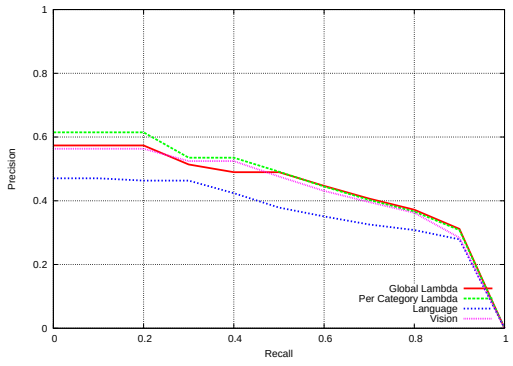

Figure 11: 11-point interpolated precision recall curve for testing data using color attributes

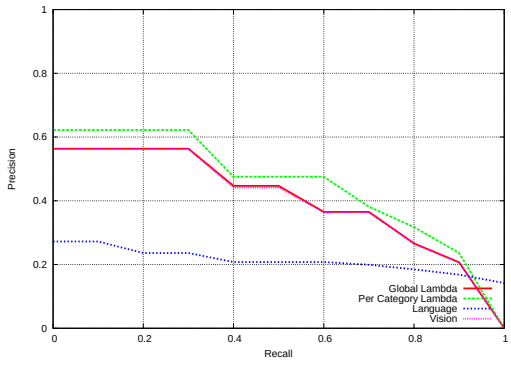

Figure 12: 11-point interpolated precision recall curve for testing data using other attributes

In order to determine how well each model performed with a varying amount of training data, a 10-fold learning curve was created. The method is very similar to the 10-fold cross validation, but rather than training on all $n - 1$ buckets, the size of the training set was varied and the accuracy was measured. The learning curve for the color attributes is shown in Figure 13. This curve shows that the models typically perform better with more training data and with lots of training data, the preferred models are those that utilize the prior. However, in the zero-shot case, where we have no training data, it can be seen that the interpolated global and per-category lambda models (using

$\lambda = 0.5$) perform better than the prior models. With no training data, the prior models rely entirely on the vision model.

The learning curve for the other attributes is shown in Figure 14. Similarly, this curve shows that the models typically perform better with more training data and with lots of training data, the preferred models are those that utilize the prior. It also shows that in the zero-shot case, where we have no training data, the interpolated global lambda with prior model performs the best. In the zero-shot case, the interpolated global and per-category lambda model use $\lambda = 0.5$ and this drags down performance since the optimal value of λ tends to be closer to 1.

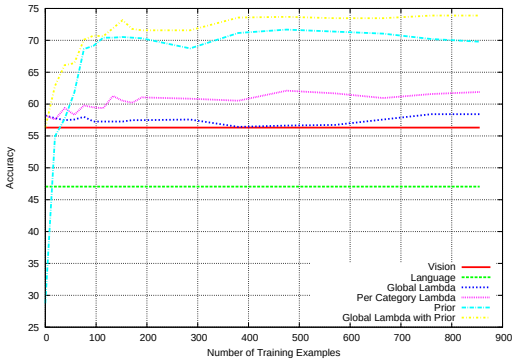

Figure 13: 10-fold learning curve using color attributes

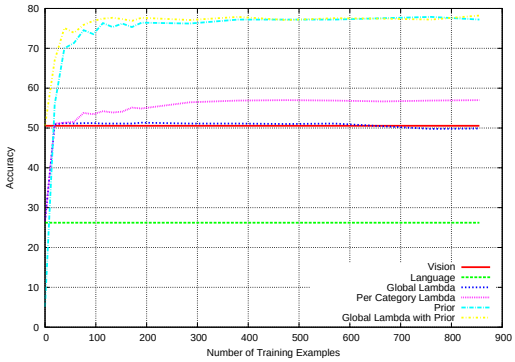

Figure 14: 10-fold learning curve using other attributes

We noticed that categories with obvious or typical colors are generally not described in language. Writers and speakers tend to omit these evident attributes because they are common knowledge and are unnecessary in conversation. Objects which have a wide variety of attributes associated with them, such as cars, tend to be described in greater detail in language. On the other hand, we prefer describing objects which have an interesting or unexpected attribute. For example, when describing a plant, one is more inclined to mention its color if it is purple rather than green. Only in specific mediums, such as detailed descriptions of images and vivid imagery used by authors, will objects be described by their obvious attributes. This was also true for the other attributes that were used. For these features, we observed that the language is not providing much help for the attributes that describe materials and textures. In many cases, the language model did not highlight attributes normally found in the real world. For example, the top attribute reported by the language model for the train category is 'wood,' which is more representative of a toy train than an actual train.

These insights are reflected in Tables 4 and 5. In Table 4, the plant category is shown to use more language than vision, perhaps because the main attribute used to describe a plant is green. Some categories, such as airplanes, have a variety of colors in images, which lead to more reliance on the vision system. The results in Table 4 seem to indicate that objects which can be described by many colors do not benefit from language as much as objects that have a limited set of typical

attribute values. This is because objects which can be described by many colors will have more of a uniform distribution from the language, negating the effectiveness of the language model. On the other hand, objects which have one major color, such as sheep, also do not benefit greatly from language. The distribution of colors in language does not always match the distributions in the real world, and we tend to omit colors that are very common. The objects which benefit most from language tend to have a limited set of colors, and are well reflected in language.

In Table 5, it is shown that all of the objects make greater use of the vision model than the language model for other attributes. As noted above, it was seen that the language model did not provide a good representation of the attributes that appear in the real world.

Object	Accuracy	Lambda
bicycle	94.0%	0.19
cow	64.0%	0.22
plant	94.0%	0.37
train	66.0%	0.37
cat	58.0%	0.41
bird	52.0%	0.46
bus	62.0%	0.47
motorbike	76.0%	0.48
aeroplane	50.0%	0.64
tv	54.0%	0.66
horse	50.0%	0.76
table	66.0%	0.80
car	46.0%	0.82
boat	58.0%	0.85
sheep	32.0%	0.96
bottle	62.0%	0.98
chair	48.0%	0.99
sofa	66.0%	0.99
dog	72.0%	0.99

Table 4: Object specific lambda and the respective accuracy for color attributes

Figures 15 through 20 show sample results from the color attribute model. The first image in the row contains the original image, the second contains the reduced color space image, the third is the histogram from the image, and the last is the histogram from the language.

Figure 15 shows an example from the cat category. In this case, the vision incorrectly identifies brown as the top color, and the language assists to show that brown is not likely, and black is the more likely color. Similarly, Figure 16 is incorrectly identifies as grey from the vision system, and the language helps to identify white as the correct color. This is also true for Figure 17. The vision system believes the bicycle is grey due to the road beneath the bicycle, but the language model correctly identifies the bicycle as black.

Figure 18 shows an example from the sheep category. Unfortunately, the sheep category does not benefit much from the language. The idiom ‘black sheep’ is very prevalent in language, as

Object	Accuracy	Lambda
sheep	92.0%	0.82
chair	50.0%	0.87
dog	48.0%	0.89
cat	72.0%	0.90
boat	54.0%	0.95
table	54.0%	0.97
bottle	54.0%	0.97
motorbike	70.0%	0.98
sofa	40.0%	0.99
bird	36.0%	0.99
cow	56.0%	0.99
bicycle	56.0%	0.99
tv	64.0%	0.99
train	70.0%	0.99
aeroplane	72.0%	0.99
horse	74.0%	0.99
car	76.0%	0.99
bus	82.0%	0.99

Table 5: Object specific lambda and the respective accuracy for other attributes

shown in the figure. Unfortunately, black sheep are not common in images and this idiom does not contribute positively to the interpolated model.

Figure 19 shows an example from the plant category. In this case, the language and vision are both correct and agree that the top color for the plant is green. This is also true in Figure 20, where the top color, black, is correctly chosen by both models.

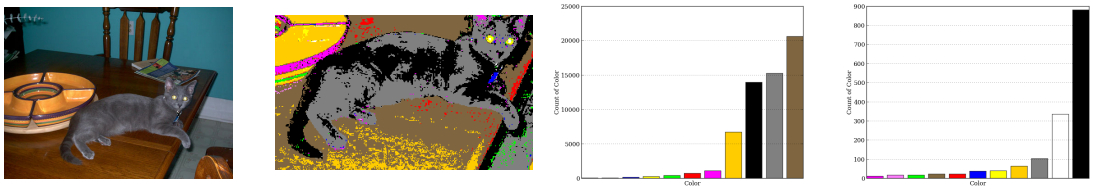

Figure 15: Example of cat category where the language helps correctly identify that the cat is black.

Figures 21 through 24 show sample results from the other attribute model. The first image in the row contains the cropped image, the second shows the other attribute scores from the image, and the last is the other attribute histogram from the language.

Figure 21 shows an example from the car category where the gold standard label is shiny. In this case, the vision model incorrectly identifies metal as the top attribute, but the language assists to show that metal is not as likely as shiny. Likewise, Figure 22 is incorrectly labeled as metal when

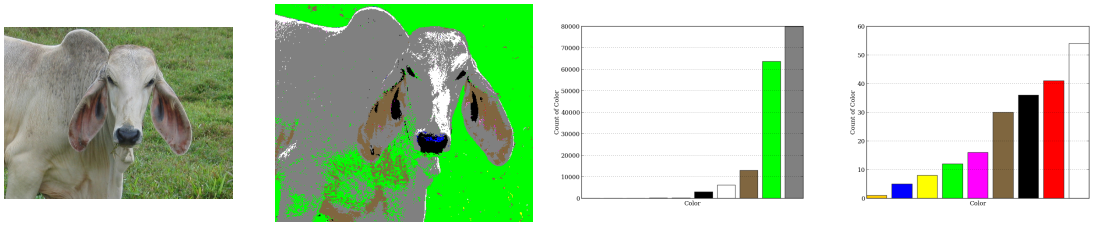

Figure 16: Example of cow category where the language helps correctly identify that the cow is white.

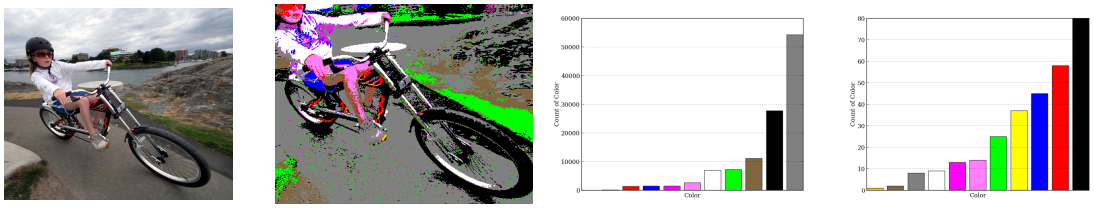

Figure 17: Example of bicycle category where the language helps correctly identify that the bicycle is black.

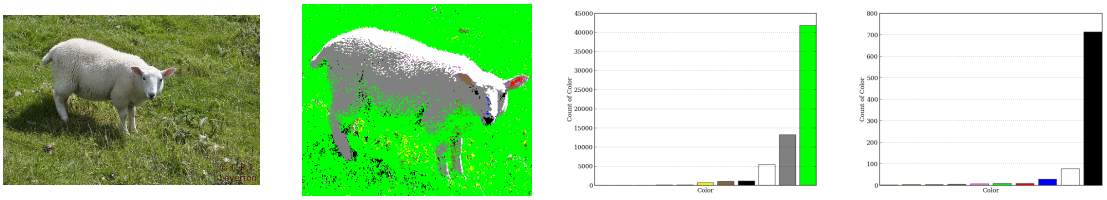

Figure 18: Example of sheep category where the language does not help the vision.

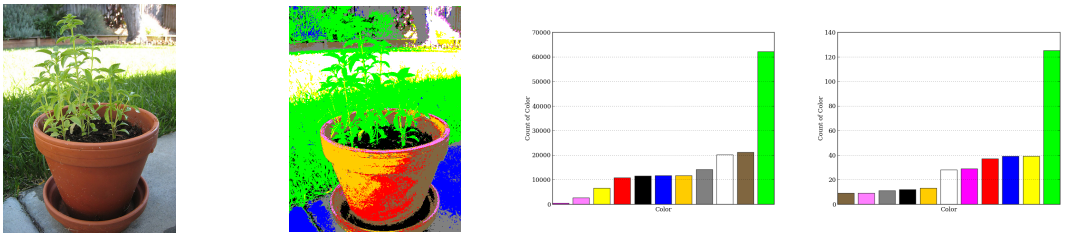

Figure 19: Example of plant category where language agrees with vision.

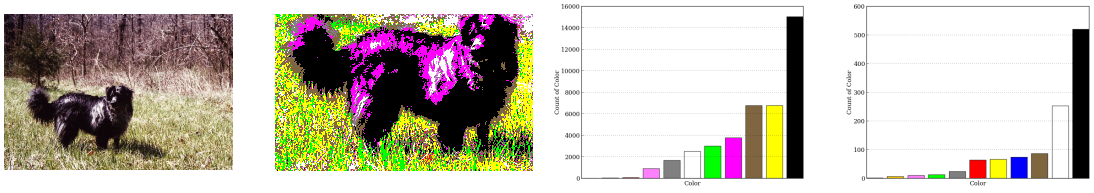

Figure 20: Example of dog category where language agrees with vision.

the gold standard label is wood. The language helps identify that wood is the more likely attribute for boats.

Figure 23 shows an example from airplane category. In this case, the vision system correctly identifies the plane as metal, but the language model offers little assistance. The language model for this category seems to be primarily influenced by toy planes. The same is true for Figure 24, an example from the bird category. Once again, the vision system correctly picks the attribute feather, but the language model incorrectly identifies the bird as furry. Unfortunately, the language model for this category is also not very strong or representative of real birds.

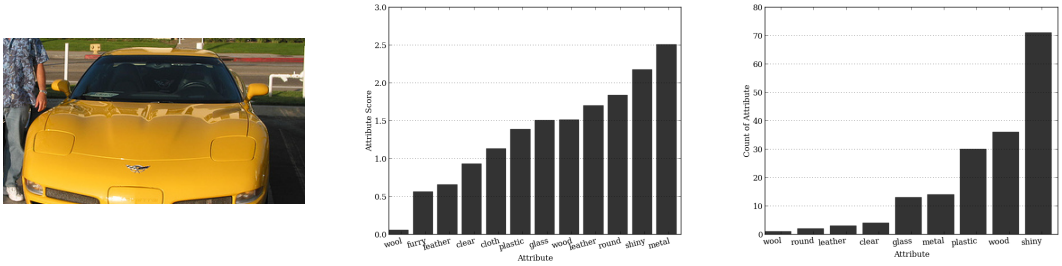

Figure 21: Example of car category where language helps correctly identify that the car is shiny.

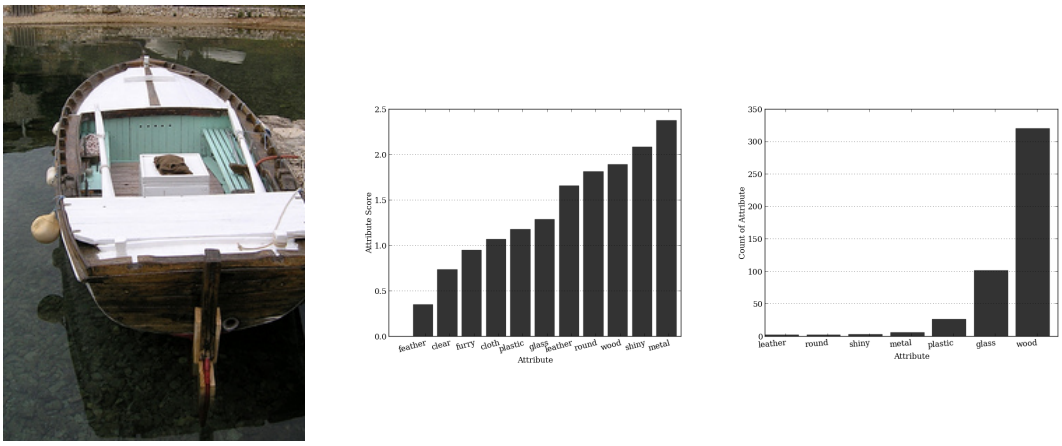

Figure 22: Example of boat category where language helps correctly identify that the boat is wooden.

5. Future Work

Our current models only support a small set of colors. In order to support a larger set, a mapping would need to be created from a large list of commonly used colors to the small group used in this experiment. This would allow us to extract a wider variety of colors from the language which in turn would help strengthen the language model.

In addition to increasing the set of colors, increasing the number of other attributes would help increase the usability of this model. To successfully do this, more attributes detectors would need

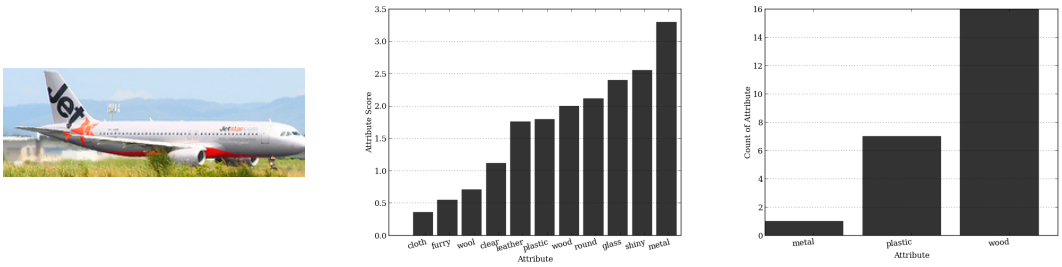

Figure 23: Example of airplane category where the language does not help the vision.

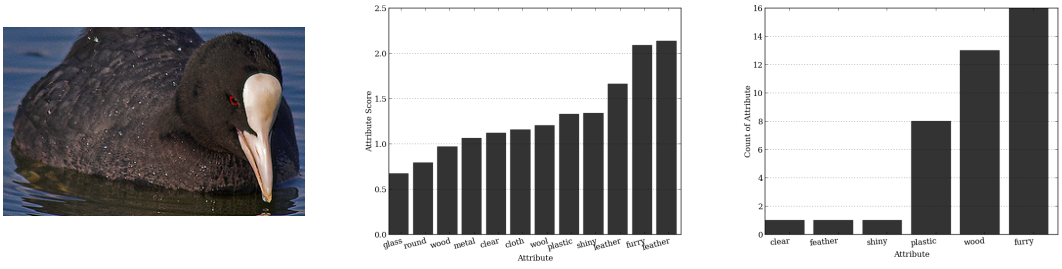

Figure 24: Example of bird category where the language does not help the vision.

to be located and used in conjunction with the current detectors. This would also likely help strengthen the language model as the set of other attributes used were mostly texture or material based.

The current corpora do not include text used to describe images. It would be nice to include image captions from Flickr which would likely assist the language model. Another unused corpus is the Google Ngram Dataset, which would have added valuable information to the language model regarding object attribute pairs.

The model only supports a small set of objects. Adding support for more objects would allow the model to be applied to a wider domain. In order to do this, more object detectors would need to be found and added to our vision model.

Another next step would be to interpolate and combine the results from vision, language, and the prior. This would involve using two constants rather than one and performing a grid search to find the optimal constants to maximize accuracy.

6. Conclusion

We have demonstrated that by using a language model in conjunction with an out-of-the-box vision system, we can describe the properties of objects better than with language or vision alone. Instead of adding complex features to the vision system such as image segmentation, a language model is used. This language model compensates for when the basic vision system fails to clearly identify the object shown. Our interpolated model performs better than both the pure vision and pure language models when tested on the PASCAL dataset using both color and other attributes.

The interpolated per-category lambda model was also found to be statistically significantly more accurate than the vision system alone for both sets of attributes. While our proposed models fell short of the top-line models, we have demonstrated the usefulness of combining visual and linguistic information and shown that in the zero-shot case with color attributes, our proposed models outperform all other models.

Acknowledgment

I would like to thank my wonderful thesis advisor, Dr. Ray Mooney, for mentoring and supporting me during this academically challenging journey. I would also like to thank my Honors Thesis Committee, Dr. Kristen Grauman and Dr. William Press, for their feedback throughout this process. Also, to everyone in the Machine Learning Research Group, thank you for your welcoming environment and continued support. Lastly, I would like to thank my family and friends for the encouragement and support, I would not have made it this far without you.

References

[1] C. Chen and K. Grauman. Inferring Analogous Attributes. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[2] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results, 2008.

[3] A. Farhadi, I. Endres, D. Hoiem, and D.A. Forsyth, Describing Objects by their Attributes. *IEEE Converence on Computer Vision and Pattern Recognition*, 2009.

[4] A. Farhadi, M. Hejrati, M. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. Every Picture Tells a Story: Generating Sentences from Images. *European Conference on Computer Vision*, 2010.

[5] R. Girshick, P. Felzenszwalb, and D. McAllester. Discriminatively Trained Deformable Part Models, Release 5. `http://people.cs.uchicago.edu/~rbg/latent-release5/`, 2012.

[6] N. Krishnamoorthy, G. Malkarnenkar, R. Mooney, K. Saenko, S. Guadarrama. Generating Natural-Language Video Descriptions Using Text-Mined Knowledge. *In Proceedings of the NAACL HLT Workshop on Vision and Language*, 2013.

[7] G. Kulkarni, V. Premraj, S. Dhar, S. Li, Y. Choi, A. Berg, and T. Berg. Baby Talk: Understanding and Generating Image Descriptions. *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[8] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning Realistic Human Actions from Movies. *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[9] M. Marneffe, B. MacCartney, and C. Manning. Generating Typed Dependency Parses from Phrase Structure Parses. *Proceedings of the International Conference on Language Resources and Evaluation*, 2006.

[10] C. Rashtchian, P. Young, M. Hodosh, and J. Hockenmaier. Collecting Image Annotations Using Amazon's Mechanical Turk. *In Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, 2010.

[11] J. Weijer, C. Schmid, J. Verbeek, and D. Larlus. Learning Color Names for Real-World Applications. *IEEEE Transactions on Image Processing*, 2009.

[12] B. Yao, X. Yang, L. Lin, M. Lee, and S. Zhu. I2T: Image Parsing to Text Description. *Proceedings of the IEEE*, 2010.