Stacking With Auxiliary Features for Visual Question Answering

Nazneen Fatema Rajani
Department of Computer Science
University of Texas at Austin
nrajani@cs.utexas.edu

Raymond J. Mooney
Department of Computer Science
University of Texas at Austin
mooney@cs.utexas.edu

Abstract

Visual Question Answering (VQA) is a well-known and challenging task that requires systems to jointly reason about natural language and vision. Deep learning models in various forms have been the standard for solving VQA. However, some of these VQA models are better at certain types of image-question pairs than other models. Ensembling VQA models intelligently to leverage their diverse expertise is, therefore, advantageous.

Stacking With Auxiliary Features (SWAF) is an intelligent ensembling technique which learns to combine the results of multiple models using features of the current problem as context. We propose four categories of auxiliary features for ensembling for VQA. Three out of the four categories of features can be inferred from an image-question pair and do not require querying the component models. The fourth category of auxiliary features uses model-specific explanations. In this paper, we describe how we use these various categories of auxiliary features to improve performance for VQA. Using SWAF to effectively ensemble three recent systems, we obtain a new state-of-the-art. Our work also highlights the advantages of explainable AI models.

1 Introduction

Visual Question Answering (VQA), the task of addressing open-ended questions about images (Malinowski and Fritz, 2014; Antol et al., 2015), has attracted significant attention in recent years (Andreas et al., 2016a; Goyal et al., 2016; Agrawal et al., 2016; Teney et al., 2017). Given an image and a natural language question about the image, the task is to provide an accurate natural language answer. VQA requires visual and linguistic comprehension, language grounding as well as common-sense knowledge. A variety of methods to address these challenges have been developed in recent years (Fukui et al., 2016; Xu and Saenko, 2016; Lu et al., 2016; Chen et al., 2015). The vision component of a typical VQA system extracts visual features using a deep convolutional neural network (CNN), and the linguistic component encodes the question into a semantic vector using a recurrent neural network (RNN). An answer is then generated conditioned on the visual features and the question vector.

Most VQA systems have a single underlying method that optimizes a specific loss function and do not leverage the advantage of using multiple diverse models. One recent ensembling approach to VQA (Fukui et al., 2016) combined multiple models that use multimodal compact bilinear pooling with attention and achieved state-of-the-art accuracy on the VQA 2016 challenge. However, their ensemble uses simple softmax averaging to combine outputs from multiple systems. Also, their model is pre-trained on the Visual Genome dataset (Krishna et al., 2017) and they concatenate learned word embeddings with pre-trained GloVe vectors (Pennington et al., 2014). Several other deep and non-deep learning approaches for solving VQA have also been proposed (Lu et al., 2016; Zhou et al., 2015; Noh et al., 2016). Although these models perform fairly well on certain image-question (IQ) pairs, they fail spectacularly on certain other IQ pairs. This led us to conclude that the various VQA models have learned to perform well on specific types of questions and images. Therefore, there is an opportunity to combine these models intelligently so as to leverage their diverse strengths.

Ensembling multiple systems is a well known standard approach to improving accuracy in machine learning (Dietterich, 2000). Stacking with Auxiliary Features (SWAF) (Rajani and Mooney, 2017) is a recent ensembling algorithm that learns to combine outputs of multiple systems using fea-

Proceedings of NAACL-HLT 2018, pages 2217–2226
New Orleans, Louisiana, June 1 - 6, 2018.

Q. Is that a frisbee?
A. Yes
Q. Is this a man or a woman?
A. Woman
Q. What color is the frisbee?
A. Red

Q. Is this a romantic spot that couples would like to go?
A. Yes
Q. What time of day is it?
A. Night
Q. How many spires below big ben's clock?
A. 10

Figure 1: Random sample of images with questions and ground truth answers taken from the VQA dataset.

tures of the current problem as context. In this paper, we use SWAF to more effectively combine several VQA models. Traditional stacking (Wolpert, 1992) trains a supervised meta-classifier to appropriately combine multiple system outputs. SWAF further enables the stacker to exploit additional relevant knowledge of both the component systems and the problem by providing *auxiliary features* to the meta-classifier. Our approach extracts features from the IQ pair under consideration, as well as the component models and provides this information to the classifier. The meta-classifier then learns to predict whether a specific generated answer is correct or not.

Explanations attempt to justify a system's predicted output and provide context for their decision that may also help SWAF. We extract *visual* explanations from various deep learning models and use those as auxiliary features for SWAF. Our contributions can be summarized as follows: (a) developing novel auxiliary features that can be inferred from VQA questions and images; (b) extracting visual explanations from several component models for each IQ pair and using those to also generate auxiliary features; and (c) using SWAF to ensemble various VQA models and evaluating ablations of features while comparing our approach extensively to several individual as well as ensemble systems. By effectively ensembling three leading VQA systems with SWAF, we demonstrate state-of-the-art performance.

2 Background and Related Work

VQA is the task of answering a natural language question about the content of an image by returning an appropriate word or phrase. Figure 1 shows a sample of images and questions from the VQA 2016 challenge. The dataset consists of images taken from the MS COCO dataset (Lin et al., 2014) with three questions and answers per image obtained through Mechanical Turk (Antol et al., 2015). Table 1 summarizes the splits in the VQA dataset. Several deep learning models have been developed that combine a computer vision component with a linguistic component in order to solve the VQA challenge. Some of these models also use data-augmentation for pre-training. We discuss the VQA models we use in Section 5.

	Images	Questions
Training	82,783	248,349
Validation	40,504	121,512
Test	81,434	244,302

Table 1: VQA dataset splits.

Stacking With Auxiliary Features (SWAF) is an ensembling technique that combines outputs from multiple systems using their confidence scores and task-relevant features. It has previously been applied effectively to information extraction (Viswanathan et al., 2015), entity linking (Rajani and Mooney, 2016) and ImageNet object detection (Rajani and Mooney, 2017). To the best of our knowledge, there has been no prior work on stacking for VQA, and we are the first to show how model-specific explanations can serve as an auxiliary feature. The auxiliary features that we use are motivated by an analysis of the VQA dataset and also inspired by related work, such as using a Bayesian framework to predict the form of the answer from the question (Kafle and Kanan, 2016).

Deep learning models have been used widely on several vision and language problems. However, they frequently lack transparency and are unable to explain their decisions (Selvaraju et al., 2017). On the other hand, humans can justify their decisions with natural language as well as point to the visual evidence that supports their decision. There are several advantages of having AI systems that can generate explanations that support their predictions (Johns et al., 2015; Agrawal et al., 2016). These advantages have motivated recent work on explainable AI systems, particularly in computer vision (Antol et al., 2015; Goyal et al., 2016; Hendricks et al., 2016; Park et al., 2016). However, there has been no prior work on using explanations for ensembling multiple models or improving performance on a challenging task. In this

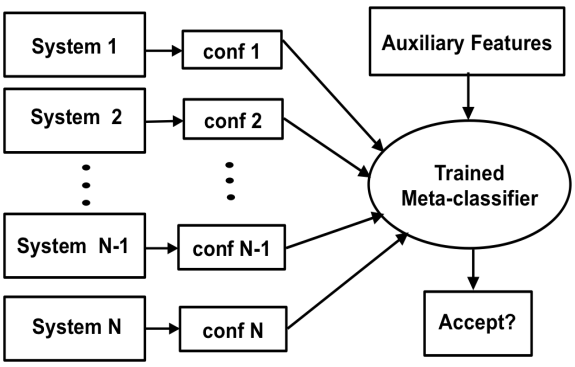

Figure 2: Ensemble Architecture using Stacking with Auxiliary Features. Given an input, the ensemble judges every possible question-answer pair produced by the component systems and determines the final output answer.

paper, we generate visual explanations for three different VQA models and use these explanations to develop auxiliary features that aid in effectively ensembling VQA systems.

3 Stacking With Auxiliary Features (SWAF) for VQA

In stacking, a meta-classifier is learned to combine the outputs of multiple underlying systems (Wolpert, 1992). The stacker learns a classification boundary based on the confidence scores provided by individual systems for each possible output. However, many times the scores produced by systems are not probabilities or not well calibrated and cannot be meaningfully compared. In such circumstances, it is beneficial to also have other reliable auxiliary features, as in the SWAF approach. SWAF provides the meta-classifier additional information, such as features of the current problem and provenance or explanation information about the output from individual systems. This allows SWAF to *learn* which systems do well on which types of problems and when to trust agreements between specific systems. The learned meta-classifier makes a binary decision whether or not to accept a particular output. Figure 2 gives an overview of the SWAF approach.

For stacking VQA systems, we first form unique question-answer pairs across all of the systems' outputs before passing them through the stacker. If a system generates a given output, then we use its probability estimate for that output, otherwise, the confidence is considered zero. If a question-answer pair is classified as correct by the stacker, and if there are other answers that are also classified as correct for the same question, the output with the highest meta-classifier confidence is chosen. For questions that do not have any answer classified as correct by the stacker, we choose the answer with lowest classifier confidence, which means it is least likely to be incorrect. The reason we do this is that the online VQA scorer expects an answer for each question in the test set and penalizes the model for every unanswered question.

The confidence scores along with other auxiliary features form the complete set of features used by the stacker. The auxiliary features are the backbone of the SWAF approach, enabling the stacker to intelligently learn to rely on systems' outputs conditioned on the supporting evidence. We use a total of four different categories of auxiliary features for VQA. Three of these types can be inferred directly from the image-question (IQ) pair and do not require querying the individual models. For the fourth category of auxiliary features, we generate *visual explanations* for the component models and use these to create the explanation auxiliary features. The first three categories of features are discussed below and the fourth category is discussed in the next section.

3.1 Question and Answer Types

Antol et al. (2015) analyzed the VQA data and found that most questions fall into several types based on the first few words (e.g. questions beginning with "What is...", "Is there...", "How many...", or "Does the..."). Using the validation data, we discover such lexical patterns to define a set of question types. The questions were tokenized and a question type was formed by adding one token at a time, up to a maximum of five, to the current substring. The question "What is the color of the vase?" has the following types: "What", "What is", "What is the", "What is the color", "What is the color of". The prefixes that contain at least 500 questions were then retained as types. We added a final type "other" for questions that do not fall into any of the predefined types, resulting in a total of 70 question types. A 70-bit vector is used to encode the question type as a set of auxiliary features.

The original analysis of VQA answers found that they are 38% "yes/no" type and 12% numbers.

There is clearly a pattern in the VQA answers as well and we use the questions to infer some of these patterns. We considered three answer types – "yes/no", "number", and "other". The answer-type auxiliary features are encoded using a one-hot vector. We classify all questions beginning with "Does","Is","Was","Are", and "Has" as "yes/no". Ones beginning with "How many", "What time", "What number" are assigned "number" type. These inferred answer types are not exhaustive but have good coverage. The intuition behind using the question and answer types as auxiliary features is that some VQA models are better than others at handling certain types of questions and/or answers. Making this information available at the time of classification aids the stacker in making a better decision.

3.2 Question Features

We also use a bag-of-words (BOW) representation of the question as auxiliary features. Words that occur at least five times in the validation set were included. The final sparse vector representing a question was normalized by the number of unique words in the question. In this way, we are able to embed the question into a single vector. Goyal et al. (2016) showed that attending to specific words in the question is important in VQA. Including a BOW for the question as auxiliary features equip the stacker to efficiently learn which words are important and can aid in classifying answers.

3.3 Image Features

We also used "deep visual features" of the image as additional auxiliary features. Specifically, we use the $4,096$ features from VGGNet's (Simonyan and Zisserman, 2015) $fc7$ layer . This creates an embedding of the image in a single vector which is then used by the stacker. Using such image features enables the stacker to learn to rely on systems that are good at identifying answers for particular types of images. Recall that the individual VQA models fuse an embedding of the image along with an embedding of the question. By using the question and image embeddings at the meta-classifier level, the stacker learns to discriminate between the component models based on a deeper representation of the IQ pair.

4 Using Explanations

Recently, there has been work on analyzing regions of an image that deep-learning models focus on when making decisions (Goyal et al., 2016; Hendricks et al., 2016; Park et al., 2016). This work shows that deep-learning models attend to relevant parts of the image when making a decision. For VQA, the parts of images that the models focus on can be thought of as *visual* explanations for answering the question. We use these visual explanations to construct auxiliary features for SWAF. The idea behind using explanation features is that they enable the stacker to learn to trust the agreement between systems when they also agree on the heat-map explanation by "looking" at the right region of the image when generating an answer.

4.1 Generating Explanations

We use the GradCAM algorithm (Selvaraju et al., 2017) to generate model-specific explanatory heat-maps for each IQ pair. This approach generates a class-discriminative localization-map for a given model based on its respective predicted output class in the following way. First, the gradient of the score y^c for the predicted class c is computed before the softmax layer with respect to the feature maps A^k of a convolutional layer. Then, the gradients flowing back are global average pooled to obtain the neuron importance weights.

$$w_k^c = \overbrace{\frac{1}{Z}\sum_i\sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial y^c}{\partial A_{ij}^k}}_{\text{backprop gradients}}$$

The above weights capture the importance of a convolutional feature map k for the output class c, where Z is the total number of pixels in the feature map. A ReLU over the weighted combination of the feature maps results in the required localization-map for the output class as follows:

$$H^c = ReLU(\sum_k w_k^c A^k)$$

For each of the component VQA models, we generate the localization-map to be used as auxiliary features for ensembling. Figure 3 shows a sample of IQ pairs from the VQA dataset and their respective heat-maps generated for three VQA models.

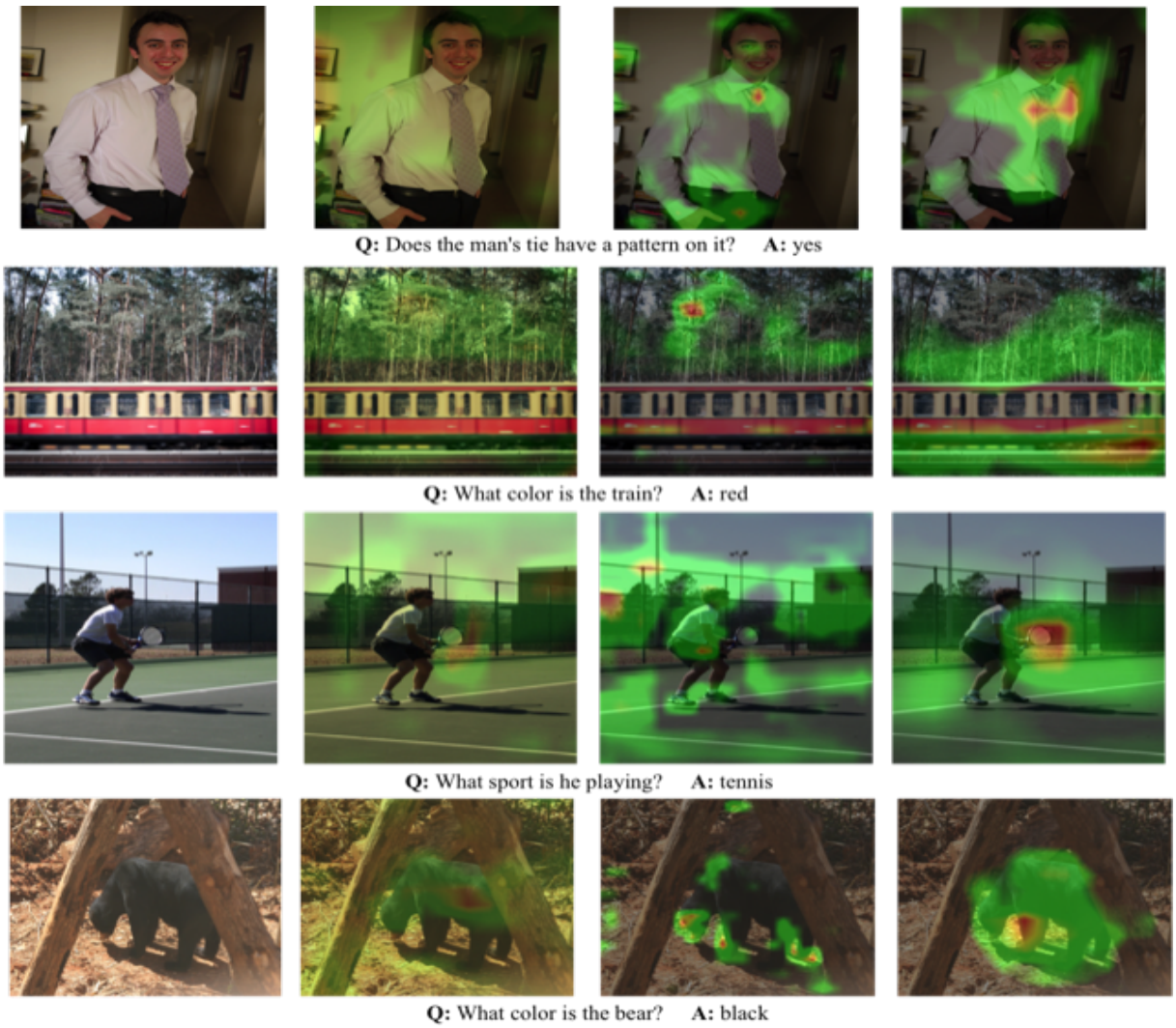

Figure 3: Each row from left to right shows an image-question pair from the VQA dataset along with localization-maps overlaid on the image generated by the LSTM, HieCoAtt and MCB models respectively. The answers shown are those predicted by our ensemble.

4.2 Explanation as Auxiliary Features

The localization-map generated by each VQA model serves as a visual explanation for the predicted output of that model. We compare agreement between the localization-maps of the individual models to generate auxiliary features for SWAF. We take the absolute gray-scale value of the localization-maps in of each model and compute their mean rank-correlation with the localization-map of every other model. We rank the pixels according to their spatial attention and then compute the correlation between the two ranked lists. The rank correlation protocol has been used in the past to compare machine-generated and human attention-maps as described by Das et al. (2016). We also experimented with using the Earth Mover's Distance (EMD) in place of the rank-order correlation metric, as discussed in Section 6. We compare the localization-maps of each pair of VQA models, generating $\binom{n}{2}$ "explanation agreement" auxiliary features for SWAF, where n is the total number of models.

5 Component VQA Systems

We use SWAF to combine three diverse VQA systems such that the final ensemble performs better than any individual component model even on questions with a low agreement. The three component models are trained on the VQA training set. Each of the three models is described below.

5.1 Long Short-Term Memory (LSTM)

The LSTM model (Antol et al., 2015) is one of the original baseline models used to establish a

benchmark for the VQA dataset. A VGGNet (Simonyan and Zisserman, 2015) is used to obtain embeddings for the image which is combined with an LSTM (Hochreiter and Schmidhuber, 1997) embedding of each question. An LSTM with two hidden layers is used to obtain a $2,048$-dimensional embedding of the question, followed by a fully-connected layer with *tanh* non-linearity to transform the embedding to $1,024$ dimensions. The l_2 normalized activations from the last hidden layer of VGGNet are used as a $4,096$ dimensional image embedding. The image embedding is first transformed to $1,024$ dimensions by a fully-connected layer with *tanh* nonlinearity to match the dimensionality of the LSTM embedding of the question. The transformed image and LSTM embeddings are then fused via element-wise multiplication.

5.2 Hierarchical Question-Image Co-Attention (HieCoAtt)

The idea behind the HieCoAtt model is that in addition to using visual attention to focus on where to look, it is equally important to model what words to attend to in the question (question-attention) (Lu et al., 2016). This model jointly reasons about the visual and language components using "co-attention". Question attention is modeled using a hierarchical architecture at word, phrase, and question levels.

HieCoAtt uses two types of co-attention – parallel and alternating. Parallel co-attention attends to the image and question simultaneously by calculating the similarity between image and question features at all pairs of image-locations and question-locations. Alternating co-attention sequentially alternates between generating image and question attention by attending to the image based on the question summary vector and then attending to the question based on the attended image features.

5.3 Multimodal Compact Bilinear pooling (MCB)

The MCB model combines the vision and language vector representations using an outer product instead of the traditional approach of using concatenation or element-wise product or sum of the two vectors (Fukui et al., 2016). Bilinear pooling computes the outer product between two vectors which, in contrast to the element-wise product, allows a multiplicative interaction between all elements of both vectors. To overcome the challenge of high dimensionality due to the outer product, the authors adopt the idea of using Multimodal Compact Bilinear pooling (MCB) (Gao et al., 2016) to efficiently and expressively combine multimodal features.

The MCB model extracts representations for the image using the 152-layer Residual Network (He et al., 2016) and an LSTM (Hochreiter and Schmidhuber, 1997) embedding of the question. The two vector are pooled using MCB and the answer is obtained by treating the problem as a multi-class classification problem with $3,000$ possible classes. The best MCB model is an ensemble of seven attention models and uses data-augmentation for pre-training along with pre-trained GloVe word embeddings. The best MCB model won the VQA 2016 challenge by obtaining the best performance on the test set.

6 Experimental Results and Discussion

We present experimental results on the VQA challenge using the SWAF approach and compare it to various baselines, individual and ensemble VQA models, as well as ablations of our SWAF algorithm on the standard VQA test set. In addition to the three data splits given in Table 1, the VQA challenge divides the test set into *test-dev* and *test-standard*. Evaluation on either split requires submitting the output to the competition's online server.[1] However, there are fewer restrictions on the number of submissions that can be made to the *test-dev* compared to the *test-standard*. The *test-dev* is a subset of the standard test set consisting of randomly selected $60,864$ (25%) questions. We use the *test-dev* set to tune the parameters of the meta-classifier. All the individual VQA models that we ensemble are trained only on the VQA training set and the SWAF meta-classifier is trained on the VQA validation set.

For the meta-classifier, we use a $L1$-regularized SVM classifier for generic stacking and stacking with only question/answer types as auxiliary features. For the question, image, and explanation features, we found that a neural network with two hidden layers works best. The first hidden layer is fully connected and the second has approximately half the number of neurons as the first layer. The question and image features are high-dimensional and therefore a neural network classifier worked

[1] www.visualqa.org/challenge.html

Method	All	Yes/No	Number	Other
DPPNet (Noh et al., 2016)	57.36	80.28	36.92	42.24
iBOWIMG (Zhou et al., 2015)	55.72	76.55	35.03	42.62
NMNs (Andreas et al., 2016b)	58.70	81.20	37.70	44.00
LSTM (Antol et al., 2015)	58.20	80.60	36.50	43.70
HieCoAtt (Lu et al., 2016)	61.80	79.70	38.70	51.70
MCB (Single system) (Fukui et al., 2016)	62.56	80.68	35.59	52.93
MCB (Ensemble) (Fukui et al., 2016)	66.50	**83.20**	39.50	58.00
Voting (MCB + HieCoAtt + LSTM)	60.31	80.22	34.92	48.83
Stacking	63.12	81.61	36.07	53.77
+ Q/A type features	65.25	82.01	36.50	57.15
+ Question features	65.50	82.26	38.21	57.35
+ Image features	65.54	82.28	38.63	57.32
+ Explanation features	**67.26**	82.62	**39.50**	**58.34**

Table 2: Accuracy results on the VQA *test-standard* set. The first block shows performance of a VQA model that use external data for pre-training, the second block shows single system VQA models, the third block shows an ensemble VQA model that also uses external data for pre-training, and the fourth block shows ensemble VQA models.

well. We found that using late fusion (Karpathy et al., 2014) to combine the auxiliary features for the neural network classifier worked slightly better. We used Keras with Tensorflow back-end (Chollet, 2015) for implementing the network. We compare our approach to a voting baseline that returns the answer with maximum agreement, with ties broken in the favor of systems with higher confidence scores. We also compare against other state-of-the-art VQA systems not used in our ensemble: iBowIMG (Zhou et al., 2015), DPPNet (Noh et al., 2016) and the Neural Module Networks (NMNs) (Andreas et al., 2016b).

The iBowIMG concatenates the image features with the bag-of-word question embedding and feeds them into a softmax classifier to predict the answer, resulting in performance comparable to other models that use deep or recursive neural networks. The iBowIMG beats most VQA models considered in their paper. The DPPNet, on the other hand, learns a CNN with some parameters predicted from a separate parameter prediction network. Their parameter prediction network uses a Gated Recurrent Unit (GRU) to generate a question representation and maps the predicted weights to a CNN via hashing. The DPPNet uses external data (data-augmentation) in addition to the VQA dataset to pre-train the GRU. Another well-known VQA model is the Neural Module Network (NMN) that generates a neural network on the fly for each individual image and question. This is done through choosing from various sub-modules based on the question and composing these to generate the neural network, *e.g.*, the `find[x]` module outputs an attention map for detecting `x`. To arrange the modules, the question is first parsed into a symbolic expression and using these expressions, modules are composed into a sequence to answer the query. The whole system is trained end-to-end through backpropagation.

The VQA evaluation server, along with reporting accuracies on the full question set, also reports a break-down of accuracy across three answer categories. The image-question (IQ) pairs that have answer type as "yes/no", those that have "number" as their answer type and finally those that do not belong to either of the first two categories are classified as "other". Table 2 shows the full and category-wise accuracies. All scores for the stacking models were obtained using the VQA *test-standard* server. The table shows results for both single system and ensemble MCB models. We used the single system MCB model as a component in our ensemble. The ensemble MCB system, however, was the top-ranked system in the VQA 2016 challenge and it is pre-trained on the Visual Genome dataset (Krishna et al., 2017) as well as uses pre-trained GloVe vectors (Pennington et al., 2014). On the other hand, our ensemble system does not use any external data and consists

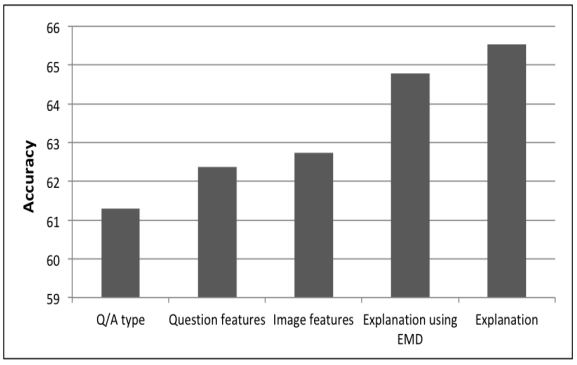

Figure 4: Results for auxiliary feature ablations on the VQA *test-dev* set. The `x-axis` indicates the feature set that was ablated from the final ensemble.

of only three component models.

The SWAF approach obtains a new state-of-the-art result on the VQA task. The vanilla stacking approach itself beats the best *individual* model and adding the auxiliary features further boosts the performance. Our SWAF model that uses all three sets of auxiliary features related to IQ pairs does particularly well on the more difficult "other" answer category, indicating that the auxiliary features provide crucial information at classification time. To further analyze the SWAF results, we performed experiments with ablations of the auxiliary features. Figure 4 shows the results on the *test-dev* set obtained when ablating each of the auxiliary feature sets. We observe that deleting the Q/A type decreased performance the most and deleting the explanation features decreased performance the least. This indicates that the Q/A type features are the most informative and the explanation features are the least informative for deciding the correct answer.

The voting baseline does not perform very well even though it is able to beat one of the component models. The SWAF ablation results clearly indicate that there is an advantage to using each type of auxiliary feature. Each of the auxiliary feature sets contributes to the final ensemble's performance, which is clear from Table 2. The voting and the "vanilla stacking" ensembles do not perform as well as SWAF. This leads us to conclude that the performance gain is actually obtained from using the auxiliary features.

In particular, using explanations generated by various deep learning models as auxiliary features improved performance. We observed that the localization-maps generated were fairly noisy, as is evident from Figure 3. Although the individual component systems agreed on an answer for many of the IQ pairs, the regions of the image they attend to varied significantly. However, the rank correlation metric in the auxiliary features made the localization-maps useful for ensembling. This is because, when training on the validation set, the stacker *learns* how to weight the auxiliary features, including those obtained using localization-maps. In this way, it learns to trust only the localization-maps that are actually useful. We also observed that there was a high positive correlation between the localization-maps generated by the HieCoAtt and MCB models, followed by the LSTM and MCB models, and then the LSTM and HieCoAtt models with several of the maps even negatively correlated between the last two models.

We also experimented with using Earth Mover's Distance (EMD) to compare heat-maps and found that it worked even better than rank-order correlation; however, it came at a cost of high computational complexity ($\mathcal{O}(n^3)$ vs. $\mathcal{O}(n)$). Figure 4 shows the difference in performance obtained when explanation features calculated using either EMD or rank-order correlation are ablated from the final ensemble. Clearly, using EMD to compare explanation maps has more impact on the system's accuracy. Consistent with previous findings (Bylinskii et al., 2018), our results confirm that EMD provides a finer-grained comparison between localization maps. Overall, our work shows that the utility of explanations is not limited to just developing human trust and making models more transparent. Explanations can also be used to improve performance on a challenging task.

7 Conclusions and Future Work

We have presented results for using stacking with auxiliary features (SWAF) to ensemble VQA systems. We proposed four different categories of auxiliary features, three of which can be inferred from an image-question pair. We showed that our model trained on these auxiliary features outperforms the individual component systems as well as other baselines to obtain a new state-of-the-art for VQA. For the fourth category of features, we have proposed and evaluated the novel idea of using explanations to improve ensembling of multiple systems. We demonstrated how visual explanations for VQA (represented as localization-maps) can be used to aid stacking with auxiliary

features. This approach effectively utilizes information on the degree to which systems agree on the *explanation* of their answers. We showed that the combination of all of these categories of auxiliary features, including explanation, gives the best results.

We believe that integrating explanation with ensembling has a two-fold advantage. First, as discussed in this paper, explanations can be used to improve the accuracy of an ensemble. Second, explanations from the component systems could be used to build an explanation for the overall ensemble. That is, by combining multiple component explanations, SWAF could also produce more comprehensible results. Therefore, in the future, we would like to focus on explaining the results of an ensemble. Another issue we plan to explore is using textual explanations (Park et al., 2016) for VQA. We believe that the words in the question to which a system attends can also be used to improve ensembling. Finally, we hope to apply our approach to additional problems beyond VQA.

Acknowledgement

This research was supported by the DARPA XAI program under the AFRL grant.

References

Aishwarya Agrawal, Dhruv Batra, and Devi Parikh. 2016. Analyzing the Behavior of Visual Question Answering Models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP2016)*.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016a. Learning to compose neural networks for question answering. In *Proceedings of the Conference on Natural language learning (NAACL2016)*. pages 1545–1554.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016b. Neural module networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR2016)*. pages 39–48.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. VQA: Visual Question Answering. In *The IEEE International Conference on Computer Vision (ICCV2015)*.

Zoya Bylinskii, Tilke Judd, Aude Oliva, Antonio Torralba, and Frédo Durand. 2018. What do different evaluation metrics tell us about saliency models? *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI2018)* .

Kan Chen, Jiang Wang, Liang-Chieh Chen, Haoyuan Gao, Wei Xu, and Ram Nevatia. 2015. ABC-CNN: An attention based convolutional neural network for Visual Question Answering. *arXiv preprint arXiv:1511.05960* .

Franois Chollet. 2015. Keras. https://github.com/fchollet/keras.

Abhishek Das, Harsh Agrawal, C. Lawrence Zitnick, Devi Parikh, and Dhruv Batra. 2016. Human Attention in Visual Question Answering: Do Humans and Deep Networks Look at the Same Regions? In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP2016)*.

T. Dietterich. 2000. Ensemble methods in machine learning. In J. Kittler and F. Roli, editors, *First International Workshop on Multiple Classifier Systems, Lecture Notes in Computer Science*. Springer-Verlag, pages 1–15.

Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. 2016. Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP2016)*.

Yang Gao, Oscar Beijbom, Ning Zhang, and Trevor Darrell. 2016. Compact bilinear pooling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR2016)*. pages 317–326.

Yash Goyal, Akrit Mohapatra, Devi Parikh, and Dhruv Batra. 2016. Towards Transparent AI Systems: Interpreting Visual Question Answering Models. In *International Conference on Machine Learning (ICML) Workshop on Visualization for Deep Learning, 2016*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR2016)*. pages 770–778.

Lisa Anne Hendricks, Zeynep Akata, Marcus Rohrbach, Jeff Donahue, Bernt Schiele, and Trevor Darrell. 2016. Generating visual explanations. In *Proceedings of the European Conference on Computer Vision (ECCV2016)*. Springer, pages 3–19.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Edward Johns, Oisin Mac Aodha, and Gabriel J Brostow. 2015. Becoming the expert-interactive multi-class machine teaching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR2015)*. pages 2616–2624.

Kushal Kafle and Christopher Kanan. 2016. Answer-type prediction for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR2016)*.

Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. 2014. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR2014)*. pages 1725–1732.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. Visual Genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision (IJCV)* 123(1):32–73.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision (ECCV2014)*. Springer, pages 740–755.

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. 2016. Hierarchical Question-Image Co-Attention for Visual Question Answering. In *Advances in Neural Information Processing Systems (NIPS2016)*. pages 289–297.

Mateusz Malinowski and Mario Fritz. 2014. A multi-world approach to question answering about real-world scenes based on uncertain input. In *Advances in Neural Information Processing Systems (NIPS2014)*. pages 1682–1690.

Hyeonwoo Noh, Paul Hongsuck Seo, and Bohyung Han. 2016. Image question answering using convolutional neural network with dynamic parameter prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR2016)*. pages 30–38.

Dong Huk Park, Lisa Anne Hendricks, Zeynep Akata, Bernt Schiele, Trevor Darrell, and Marcus Rohrbach. 2016. Attentive explanations: Justifying decisions and pointing to the evidence. *arXiv preprint arXiv:1612.04757* .

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP2014)*. pages 1532–1543.

Nazneen Fatema Rajani and Raymond J. Mooney. 2016. Combining Supervised and Unsupervised Ensembles for Knowledge Base Population. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP2016)*.

Nazneen Fatema Rajani and Raymond J. Mooney. 2017. Stacking With Auxiliary Features. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI2017)*. Melbourne, Australia.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *The IEEE International Conference on Computer Vision (ICCV2017)*.

Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations (ICLR2015)*.

Damien Teney, Peter Anderson, Xiaodong He, and Anton van den Hengel. 2017. Tips and tricks for visual question answering: Learnings from the 2017 challenge. *arXiv preprint arXiv:1708.02711* .

Vidhoon Viswanathan, Nazneen Fatema Rajani, Yinon Bentor, and Raymond J. Mooney. 2015. Stacked Ensembles of Information Extractors for Knowledge-Base Population. In *Association for Computational Linguistics (ACL2015)*. Beijing, China, pages 177–187.

David H. Wolpert. 1992. Stacked Generalization. *Neural Networks* 5:241–259.

Huijuan Xu and Kate Saenko. 2016. Ask, Attend and Answer: Exploring question-guided spatial attention for Visual Question Answering. In *European Conference on Computer Vision (ECCV2016)*. Springer, pages 451–466.

Bolei Zhou, Yuandong Tian, Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. 2015. Simple baseline for Visual Question Answering. *arXiv preprint arXiv:1512.02167* .