Improving Visual Question Answering with Transfer Learning for Question Encoding

Stephen Carrow swc419@nyu.edu
Chris Rogers cdr380@nyu.edu
Isaac Haberman ijh216@nyu.edu
Hollis Nymark hln240@nyu.edu

Abstract

Transfer learning using very large pre-trained models such as GPT and BERT has recently been shown to achieve state-of-the-art results on many NLP tasks with minimal fine-tuning. However, these methods have not yet been applied to multi-modal tasks like visual question answering (VQA). We hypothesize that using BERT as a question encoder in the well-known Multimodal Compact Bilinear Pooling (MCB) architecture can improve performance by supplying better question representations that can be used to extract higher quality visual information. Additionally, we investigate using ELMo word embeddings in the MCB architecture as another form of transfer learning and compare these methods on the VQA 2.0 dataset.

1 Introduction

Visual question answering (VQA) is a uniquely positioned task within natural language processing research that joins computer vision to natural language question answering. Recent work in VQA has been significantly advanced through the use of neural networks for learning representations of both images and questions that are combined to form a single representation that is used for answering questions. Pre-trained computer vision neural networks are an essential component of neural VQA models. Such systems are used to extract visual features from images that are used as inputs to VQA architectures. A similar approach is applied to the question text by using pre-trained word embeddings such as GLoVE (Pennington et al., 2014) as input to a neural language model that encodes a question.

Two recent developments in transfer learning for natural language processing offer potential improvements over the use of pre-trained word embeddings in VQA systems. Contextualized word embeddings, the most popular being ELMo (Peters et al., 2018), generate context dependent embeddings that were shown to improve performance on several NLP tasks. Additionally, sizeable pre-trained language models such as BERT (Devlin et al., 2018) provide a task agnostic architecture that can be fine-tuned for downstream tasks and demonstrates impressive empirical gains over previous approaches for other natural language understanding tasks. We hypothesize that ELMo and BERT can be used to learn higher quality question representations than those learned using word embeddings and that those representations can enable better extraction of visual features leading to improved performance of existing VQA architectures such as MCB (Fukui et al., 2016).

The specific contributions of this work are as follows:

- We extend the MCB architecture to incorporate both ELMo contextualized word embeddings and the BERT pre-trained language model.
- We compare the extended architectures to the original MCB architecture on the VQA 2.0 (Goyal et al., 2017) dataset.

2 Related Work

Word embeddings have become a ubiquitous component in NLP systems and are used in the MCB architecture (Fukui et al., 2016). Mikolov et al. (2013) trained embeddings via a skip-gram model to predict context words given a word in a document. The model uses negative sampling with cross entropy loss to maximize log probabilities of in-context words and minimize log probabilities of out-of-context words. With GLoVE, Pennington et al. (2014) train word embeddings using a weighted least squares model on the word-

word co-occurrence matrix. Such embeddings exhibit useful properties for language modeling, for instance grouping words in a vector space that appear in a similar context in natural language, and showing additive compositionality that makes them useful for analogical reasoning tasks.

While those properties of word embeddings are desirable, constraining a word to a single representation limits its ability to model complex syntax, semantics and polysemy. To overcome this limitation, Peters et al. (2018) develop ELMo to generate contextualized word representations. ELMo embeddings consist of the hidden states of a 2-layer bidirectional language model using a token representation generated by a character n-gram convolutional layer. The pre-training consists of jointly training forward and backward language models that share token representations and softmax layers. The hidden states of ELMo depend on the entire input sequence and can therefore capture multiple uses of a single word.

Contextualized word embeddings have clear advantages over static embeddings as a feature-based approach to transfer learning. However, GPT (Radford, 2018), GPT-2 (Radford et al., 2019) and BERT show that pre-training very large transformers (Vaswani et al., 2017) that are then fine-tuned for specific NLU tasks leads to impressive empirical results on a range of NLU tasks such as those found in GLUE (Wang et al., 2018). While BERT is similar to GPT, the authors develop a novel pre-training strategy based on the Cloze task (Taylor, 1953), as well as a following sentence classification that predicts if two sentences are related. The authors of BERT demonstrate that these pre-training strategies are an important innovation enabling improved performance over GPT on downstream tasks.

Experiments with BERT have not yet explored its application to multimodal tasks like visual question answering. While approaches to VQA such as Multimodal Compact Bi-linear Pooling (Fukui et al., 2016, MCB) and many others (Pandhre and Sodhani, 2017) have shown improved performance, the language models used in those architectures are significantly smaller than BERT. In MCB, the authors extract visual features using Resnet-152 (He et al., 2016) and use learned and GLoVE embeddings as input to a two-layer randomly initialized LSTM that encodes a question. They extract a visual representation from the Resnet-152 features with an attention mechanism using the learned question representation as a query vector. Then, they combine the visual features and question representation using the Multimodal Compact Bi-linear Pooling operation. Their research shows that this composition function outperforms traditional approaches such as concatenation and use it to produce state-of-the-art results for VQA at the time of publication.

Recent state of the art results on the VQA 2.0 (Goyal et al., 2017) dataset have come from focusing on better extracting visual information using an "up-down" approach to attention. This was first shown by Anderson et al. (2018) and more recently by Jiang et al. (2018). In the "up-down" approach a bottom-up network is used to generate the visual regions that the task specific "top-down" model attends over. In these implementations a GRU (Cho et al., 2014) is used to encode a question. Because of the increased complexity of the visual model and our resource constraints, we chose to focus on the computationally more efficient MCB model for our experiments. We leave investigation of using a BERT question encoder with a more powerful visual model to future work.

3 Architecture

Our model architecture is primarily based on MCB (Fukui et al., 2016). We use the MCB attention and pooling operations and experiment with the question encoder to introduce both the MC-ELMo and MC-BERT models. Figure 1 depicts the two proposed models.

MCB The MCB architecture takes as input an image and a natural language question. The image is pre-processed and passed through a pre-trained Resnet-152 (He et al., 2016) model to generate a $2048 \times 14 \times 14$ feature tensor. The question is stripped of punctuation and split on words. Each token is embedded into a vector $\mathbf{h}_{MCB}^{(t)} \in R^{300}$ and optionally concatenated with pre-trained GLoVE embeddings $\mathbf{g}^{(t)} \in R^{300}$. Questions are encoded to a vector $\mathbf{q}_{MCB} \in R^{2048}$ by concatenating the final hidden states of a two-layer LSTM with dropout of 0.3 after each layer. The vector $\mathbf{q}_{MCB}$ is tiled and combined with the extracted visual features through the Multimodal Compact Bi-linear Pooling method. An attention mechanism consisting of two convolutional layers and a softmax attends over the spatial dimensions of the combined question and visual features to extract

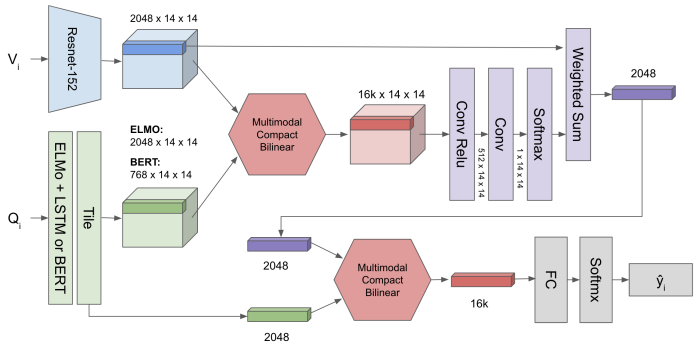

Figure 1: MC-ELMo and MC-BERT architectures. Resnet-152 takes in an image V_i. A question Q_i is encoded using either the ELMo word embeddings as input into an LSTM or a pre-trained BERT model. The output, $\hat{y}_i$, is a probability distribution over the answer vocabulary.

a single visual feature vector $\mathbf{v} \in R^{2048}$ from the Resnet-152 features. The resulting visual feature is combined again with the original LSTM output in another Mutltimodal Compact Bi-linear pooling operation. The result is passed to a fully connected layer followed by a log-softmax to predict log probabilities over the vocabulary of answers.

The MCB architecture maximizes the log probability of the correct answer token using cross entropy loss. MCB also limits the possible answer classes to the top 3000 answers found by combining the train and validation splits of VQA 2.0.

MC-ELMo MC-ELMo modifies the MCB architecture by replacing the tokenization and word embedding layers. Questions are tokenized via ELMo's word tokenizer and passed to a pre-trained ELMo model to produce a sequence of word embeddings. The ELMo embeddings, $\mathbf{h}_{ELMo} \in R^{1024}$, are used in place of the learned $\mathbf{h}_{MCB} \in R^{300}$ and GLoVE embeddings. These are fed into the MCB question encoder as before with the remainder of the model functioning as described in section 3.

MC-BERT MC-BERT also uses the MCB architecture but replaces the two-layer LSTM question encoder with a pre-trained BERT language model [1]. We also use the word piece (Wu et al., 2016) tokenizer rather than the custom tokenizer described by Fukui et al. (2016). BERT uses a special [CLS] token during pre-training to learn a next sentence classifier that predicts if two sentences are related. The hidden state of this token is therefore used as a sentence representation for downstream tasks such as classification. MC-BERT uses the hidden state of the [CLS] token as the question representation with the remainder of the MCB architecture unchanged. The resulting sentence representation is $\mathbf{q}_{BERT} \in R^{768}$ rather than $\mathbf{q}_{MCB} \in R^{2048}$ as used in MCB.

We also implement a fine-tuning strategy for MC-BERT in which the BERT language model parameters are tuned for n epochs, at which point the parameters are frozen. The number of fine-tuning epochs is set to be significantly less than the number of total epochs, $n \ll N$.

4 Experiments

We perform our experiments using the VQA 2.0 dataset. All models[2] are implemented in PyTorch (Paszke et al., 2017) and use the ADAM (Kingma and Ba, 2014) optimizer.

Data The Visual Question Answering 2.0 (VQA 2.0) data set consists of over 250k images and 1.1 million associated natural language questions. The authors of VQA 2.0 attempt to resolve several issues with the VQA 1.0 (Antol et al., 2015) dataset. As described by Goyal et al. (2017) many models were able to achieve decent accuracy without utilizing any visual information.

Goyal et al. (2017) suggest that language priors allow models to rely on prior knowledge (e.g. some answers are just more common than others) to correctly answer. Additionally, as seen by Zhang et al. (2016), visual priming bias occurs in question answering where a question is only asked about an image if the image contains the object of the question (e.g. Is there a clock tower in the image is a question for an image with a clock tower). The VQA 2.0 dataset aims to mitigate these biases by pairing different images with the same ques-

[1] All experiments use $BERT_{base}$ https://github.com/huggingface

[2] The code to replicate our experiments is available at https://github.com/estebandito22/MC-BERT.

tion, resulting in different answers that depend on the image.

We follow the convention used by Fukui et al. (2016) of using both train and validation sets to train the model. In addition, we randomly sample a small portion of validation to use as a held out set for hyper-parameter tuning.

Results We compare the proposed models, MC-ELMo and MC-BERT, to the MCB architecutre on the *test-dev* set of VQA 2.0. We make use of the VQA 2019 Challenge testing server to generate our final results. We also compare our MCB re-implementation to the published result to account for any affects due to implementation differences. Table 1 gives the overall accuracy of the four methods we investigate.

Method	Overall Accuracy
MCB - Baseline	61.96
MCB	59.52
MCB + GLoVE	60.56
MC-ELMo	59.89
MC-BERT	59.45

Table 1: Comparison of VQA architectures using different question encoders. MCB - Baseline is the published result while MCB is our re-implementation.

We find that concatenating GLoVE vectors to learned embeddings performs better than using more powerful language models like ELMO and BERT. Furthermore, using BERT performs similarly to the baseline MCB approach.

5 Analysis

We study two aspects of our model behavior using the small hold out validation set to better understand the affect of using ELMo word embeddings or a pre-trained BERT model in place of the original MCB question encoder. First, we compare the performance of the models when they are restricted to using only the question encoder path and when they also use visual information. This test should isolate if differences in overall accuracy are due to the increased capacity of the question encoder. Second, we examine accuracy by question length, which we use as a proxy for question complexity.

Question Encoder Only vs Full Model Table 2 shows the performance when using only the question encoder compared to using the full model. ELMo and BERT do perform slightly better as question encoder only models, but the improvement is even greater when using visual information. The differences are small and provide weak evidence for improved use of image features.

Method	QE Only Acc.	Overall Acc.
MCB	43.11 $\pm$0.10	59.81 $\pm$0.08
MCB+GLoVE	42.84 $\pm$0.10	60.36 $\pm$0.12
MC-ELMo	43.39 $\pm$0.08	61.47 $\pm$0.09
MC-BERT	43.77 $\pm$0.11	60.77 $\pm$0.08

Table 2: Comparison of accuracy and bootstrap estimated std. for question encoder only and full models.

Question Length Analysis Examining accuracy by question length in Figure 2 shows that the models perform similarly for shorter sentences, but MC-BERT and MC-ELMo perform slightly better for long questions. This suggest that for datasets with more complex questions a more sophisticated question encoder may in fact improve performance.

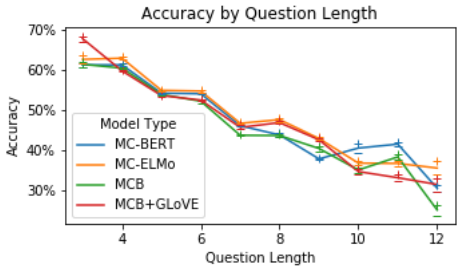

Figure 2: Comparison of accuracy $\pm$ estimated std. by sentence length on different architectures.

6 Conclusion

Our baseline model was unable to completely match the reported baseline. This could be due to differences in training or implementation of the model. By comparing internally we still draw several conclusions from our results.

We find that using pre-trained GLoVE embeddings performs better than using more sophisticated question encoders. This has some precedent. Tanti et al. (2019) show that for image captioning, using better pre-trained language models did not necessarily show better results. It may be that these specific multi-modal tasks lean more heavily on the visual information, and smaller language models are more than sufficient. Indeed as noted eariler, further improvements over the MCB-Baseline have come from improved visual processing.

Contribution Statement

Chris Rogers built the replicated MCB model and worked on debugging and tuning the framework and running trials for all models as well as building the evaluation and submission process; on the report wrote the MCB related sections and worked on the Abstract, Introduction, Results and Conclusion, and some overall editing. Stephen Carrow built the general training framework and the MC-BERT model, worked on debugging and running experiments for all models, wrote the Introduction and BERT related sections, contributed to the results, analysis and conclusion sections of the paper and edited and contributed to all sections of the paper. Hollis Nymark prepared data from the MS COCO data set, explored caption generation as next steps, and contributed to the Data and Related Work sections of the paper. Isaac Haberman obtained the VQA data, working with Chris on building the dataloader and built the ELMo components of MC-ELMo. To that end, he wrote the ELMo sections of the paper, and edited the paper. Chris and Steve built the slides, everyone contributed to preparing the presentation.

References

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. pages 6077–6086.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. VQA: visual question answering. *CoRR*, abs/1505.00468.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

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*, pages 457–468, Austin, Texas. Association for Computational Linguistics.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

K. He, X. Zhang, S. Ren, and J. Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Yu Jiang, Vivek Natarajan, Xinlei Chen, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. 2018. Pythia v0.1: the winning entry to the vqa challenge 2018. *CoRR*, abs/1807.09956.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 3111–3119, USA. Curran Associates Inc.

Supriya Pandhre and Shagun Sodhani. 2017. Survey of recent advances in visual question answering. *CoRR*, abs/1709.08203.

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. In *NIPS-W*.

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 (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Alec Radford. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Marc Tanti, Albert Gatt, and Kenneth P. Camilleri. 2019. Transfer learning from language models to image caption generators: Better models may not transfer better. *CoRR*, abs/1901.01216.

Wilson L. Taylor. 1953. cloze procedure: A new tool for measuring readability. *Journalism Bulletin*, 30(4):415–433.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Peng Zhang, Yash Goyal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2016. Yin and Yang: Balancing and answering binary visual questions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.