Challenges in Procedural Multimodal Machine Comprehension: A Novel Way To Benchmark

Pritish Sahu[1,2] *† Karan Sikka[1†] Ajay Divakaran[1]
[1]SRI International
[2]Rutgers University
pritish.sahu@rutgers.edu, {karan.sikka, ajay.divakaran}@sri.com

A. Implementation Details for HTRN

In our experiments, we experiment using Word2Vec [4] (trained on the step description of the recipes) and BERT model– "bert-basenli-mean-tokens" trained using Sentence-Transformers [6]. For visual encoder, we use the outputs of the final activation layer for ResNet50 trained on ImageNet dataset and ViT model [5] pretained with CLIP [5]. In our implementation, we used two single layer Bi-Directional LSTMs (size of the hidden layer is 256) to model the temporal information within a recipe steps and across the steps. The transformer based scoring function in HTRN was trained from scratch. This transformer uses 4 hidden layers with a size of 512 and 8 attention heads. We used CrossEntropy loss to train our model. We use the Adam [2] optimizer with the fixed learning rate of 5×10^{-4} including an early stopping criteria with a patience set to 20 for all of our experiments. We performed our experiments on a single Nvidia GTX 1080Ti and that takes about $8-10$ hours to train for a single experiment. We did not perform any hyperparameter tuning and used the same hyperparameters for all the models trained.

B. Control-Knobs Design Process

As described in Section 2.2, question preparation begins by sampling four random locations (increasing order) in the recipe. We give details of the Control-Knobs that are unique to each of the remaining visual tasks i.e. visual coherence and ordering. Control-Knob-1 is applied to all the visual tasks.

B.1. Visual Coherence

Control-Knob-2: In contrast to visual cloze where three negative choices are also provided along with the correct answer in the choice list, visual coherence on the other hand has only one incoherent image (correct answer) among the four images. For visual cloze, the selection of a negative choice was determined by it's metric distance from the correct choice whereas in coherence there are not multiple negative choices i.e. the odd the image in the set is the correct answer. Hence, we alter Control-Knob-2 process to select the incoherent image from the set of union/intersection of K nearest neighbor (KNNs) of these three coherent images. We use two Control-Knob setting and use the union set of the KNNs where the sampling space of negative choice is the Euclidean ball (0, $m_d - s_d$) or ($m_d - s_d$, $m_d + s_d$). However, the m_d and s_d are computed on the samples from the union set.

Control-Knob-3 As described in the main paper, the aim of this Control-Knob is to control the distance of negative choice from the question. In case of visual coherence, we use the three coherent images as the question and the incoherent image as the negative choice. We take the min of all pairwise distance for the coherent images. To make the negative choice closer, the distance of a randomly picked sample from the mean of question should be smaller than the min distance of all pairwise computed earlier. This forces sample of a negative choice with similar features as the question.

B.2. Visual Ordering

Control-Knob-2 For a sequence of four images randomly selected (in increasing order), excluding the correct order, there is 23 other ordering present. The altering process for Control-Knob-2 involves devising a metric to control the sampling of three negative choices out of 23 possible choices. The metric devised is as follows, for each wrong sequence, we get a score by computing the pairwise distance of the consecutive images in the sequence and adding them i.e. $\sum_{i=1}^{i=3} dist(\phi_V(x_i), \phi_V(x_{i+1}))$, where x_i represents the image at i^{th} index in the wrong sequence. We obtain 23 such scores which we refer to as weights to compute weighted probability distribution. We follow by associating each wrong sequence with their corresponding probability.

*Work done while interning at SRI International.
†These two authors contributed equally.

We finally sample three negative choice out of 23 options based on the probability associated with them. We have two setting for this Control-Knob, case-1 when the probability distribution is uniform, case-2 as discussed above. As the visual ordering task evaluates the memorization and recall ability of the model , we do not have Control-Knob-3.

C. Hierarchical Transformer based Reasoning Network (HTRN)

In this section, we briefly describe the process of preparing query vectors for visual coherence and ordering tasks.

C.1. Visual Coherence

For Visual Coherence, the question $Q = \{q_i\}_{i=1}^{N}$ consists of N_q images, including an incoherent image in between. The answer A is a scalar pointing to the location of the incoherent image. As we need N_A scores for each candidate answer a_j, we create N_A query vectors, where each query vector is prepared by removing one element from Q. Finally we obtain N_A query vectors each of size $N_A - 1$ and only one vector contains all the coherent images.

C.2. Visual Ordering

As the structure of ordering task provides N_A choice vectors each of size N_A. We do not make any changes here.

D. Meta Dataset of Meta-RecipeQA

The preparation of metadata is broken into two folds: 1) description of each recipe which is compiled from [8] and `instructables.com`, 2) control panels that moderates the scale of the question and answer during the dataset generation process. Below, we describe the process involved in the creation of Meta Dataset.

Each recipe in the metadata stage contains the following information a) name of the recipe, b) all steps required for completing the recipe, where each step includes multi-modal (text, image) information describing that step. In order to prepare the meta recipe content, we begin by further cleaning the RecipeQA [8] as we observe noise in each modality i.e. text and image. The missing texts and images from recipe steps were scrapped again from `instructables.com`. Next, we remove the noise in the textual side by processing the data again. Few of the persisting noise removed by our algorithm are "HTML/CSS" tags, Unicode, few data entries that were not food recipes ("nutrient-calculator", "cnc–nyancat–food-mold–nyancake"). After removing the above-mentioned aberrations, in the next step, we used NLTK toolkit[3] to process the out-of-dictionary vocabulary. Most of the out-of-dictionary vocabulary was found to be some form of a composite of in-dictionary vocabulary or vocabulary with numbers or measurement units. The next step of the preprocessing algorithm separates the in-dictionary words wrapped as out-of-dictionary words. This process provided us with a much cleaner version of the data.

E. Prior Scoring Functions

Impatient Reader[1] is an attention-based model that recurrently uses attention over the context for each question except the location containing the "@ placeholder". This attention allows the model to accumulate information recurrently from the context as it sees each question embeddings. It outputs a final joint embedding for the answer prediction. This embedding is used to compute a compatibility score for each choice using a cosine similarity function in the feature space. The attention over context and question is computed on the output of an LSTM. The answer choices are also encoded using an LSTM with a similar architecture.

BiDAF [7] is abbreviated for "Bi-Directional Attentional Flow", as the name suggests, it employs a bi-directional attention flow mechanism between the context, representation of the question images, and each candidate choice representation to learn temporal matching. We base our prediction on the best-matched candidate. Originally it was proposed as a span-selection model from the input context. Here, we adapt it to work in for visual tasks in multimodal setting.

F. Experimental Results

The additional visual cloze results on the remaining three pairwise combination of LM and VM hold true to our analysis in the main paper. The three tuple shown in the plots are: (Word2Vec, ViT), (BERT, ResNet–50), (BERT, ViT). We even see the impact of Control-Knob-3 in all three plots on the meta dataset as compared to it effect in the case of (Word2Vec, ResNet–50). In the case of coherence, we study the impacts using Word2Vec and ResNet-50, where Control-Knob-1 and Control-Knob-3 clearly have larger impact on model performance compared to Control-Knob-2.

References

[1] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *Advances in neural information processing systems*, pages 1693–1701, 2015.

[2] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[3] Edward Loper and Steven Bird. Nltk: The natural language toolkit. *arXiv preprint cs/0205028*, 2002.

[4] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

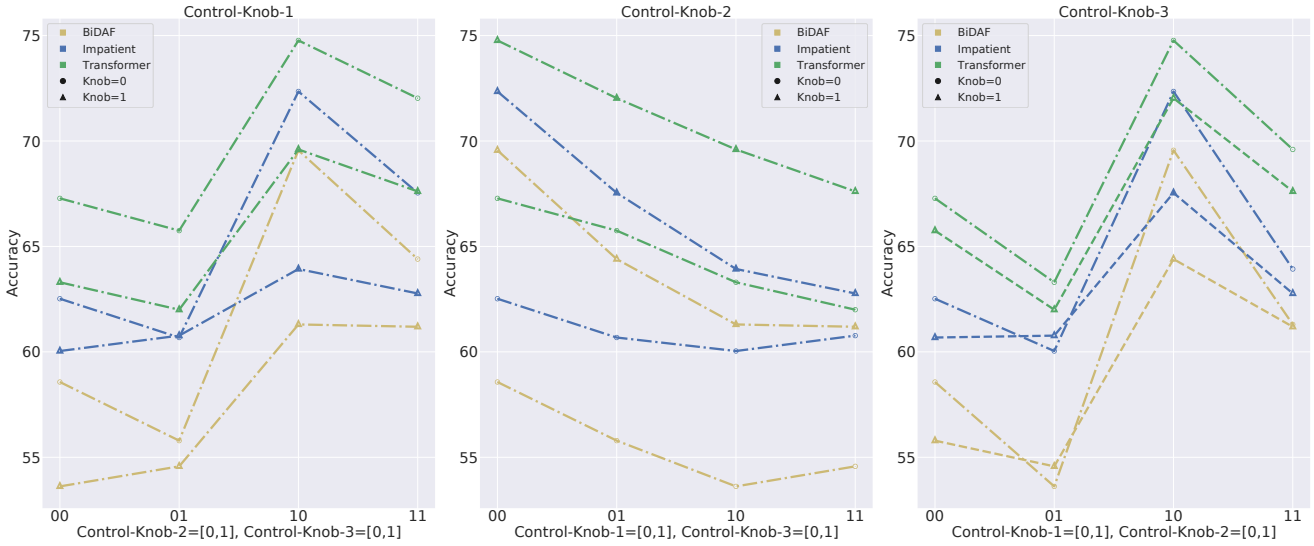

Figure 1. **Visual Cloze:** Impact of Control-Knobs by comparing the performance of different scoring functions (adapted baseline and transformer) for each knob setting. LM is set to Word2Vec and VM is set to ViT. Starting from left we plot performance of Control-Knob-2 and Control-Knob-3 for all combination of control setting by fixing Control-Knob-1. We do the same for Control-Knob-2 and Control-Knob-3 in the center and right figure respectively.

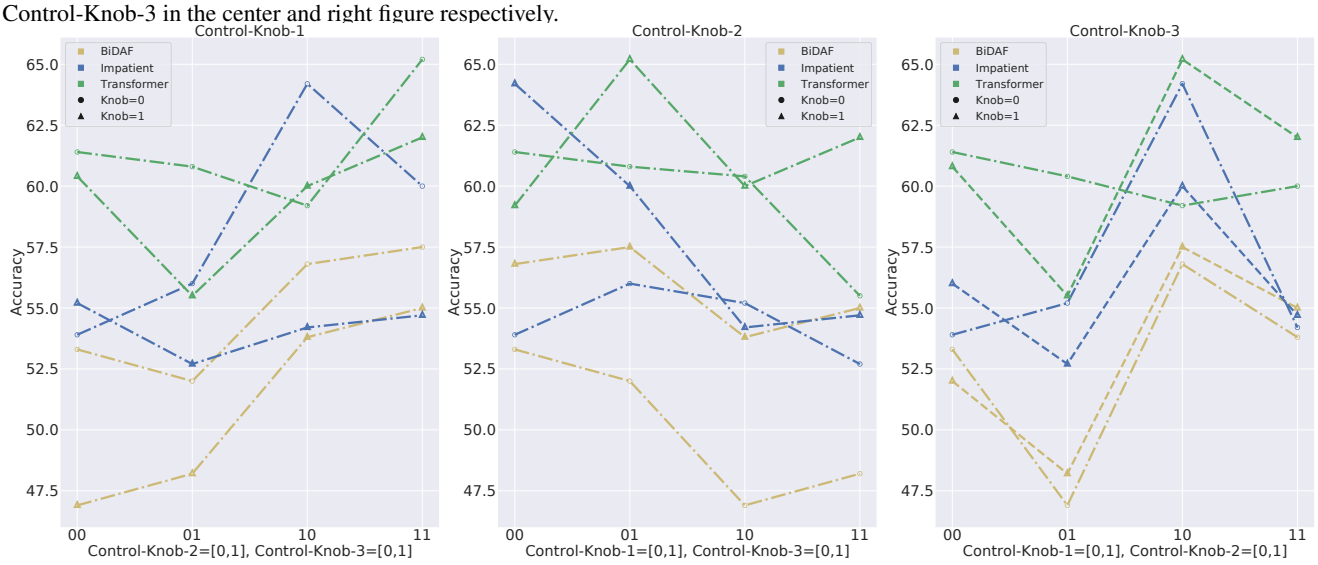

Figure 2. **Visual Cloze:** Impact of Control-Knobs by comparing the performance of different scoring functions (adapted baseline and transformer) for each knob setting. LM is set to BERT and VM is set to ResNet–50. Starting from left we plot performance of Control-Knob-2 and Control-Knob-3 for all combination of control setting by fixing Control-Knob-1. We do the same for Control-Knob-2 and Control-Knob-3 in the center and right figure respectively.

[5] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

[6] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

[7] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. *International Conference on Learning Representations (ICLR)*, 2017a.

[8] Semih Yagcioglu, Aykut Erdem, Erkut Erdem, and Nazli Ikizler-Cinbis. Recipeqa: A challenge dataset for multimodal comprehension of cooking recipes. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1358–1368, 2018.

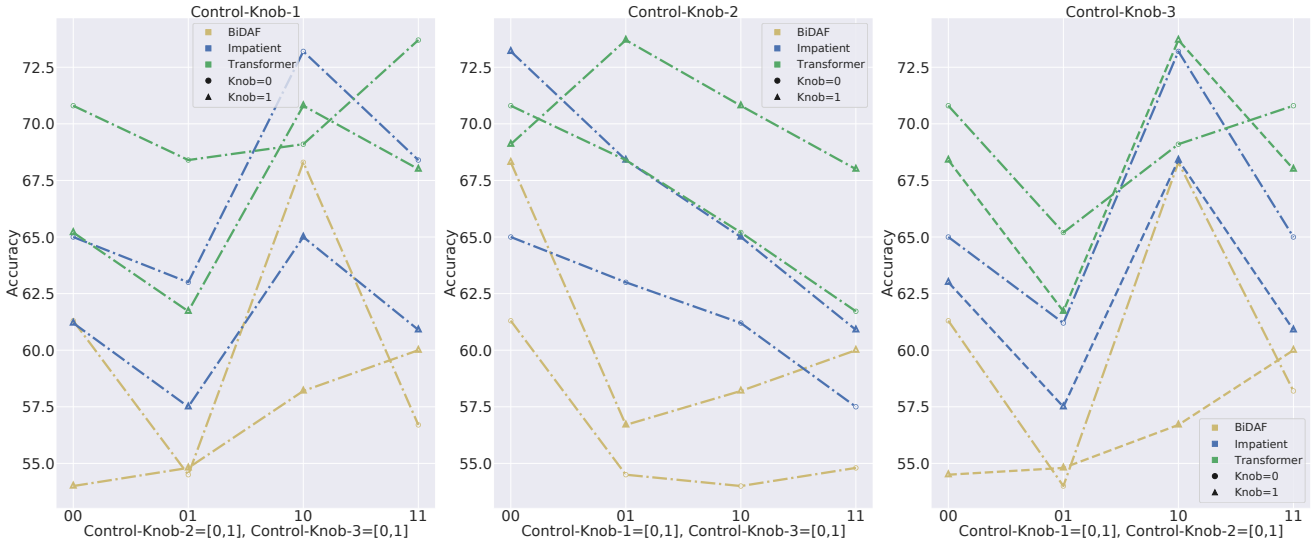

Figure 3. **Visual Cloze:** Impact of Control-Knobs by comparing the performance of different scoring functions (adapted baseline and transformer) for each knob setting. LM is set to Bert and VM is set to ViT. Starting from left we plot performance of Control-Knob-2 and Control-Knob-3 for all combination of control setting by fixing Control-Knob-1. We do the same for Control-Knob-2 and Control-Knob-3 in the center and right figure respectively.

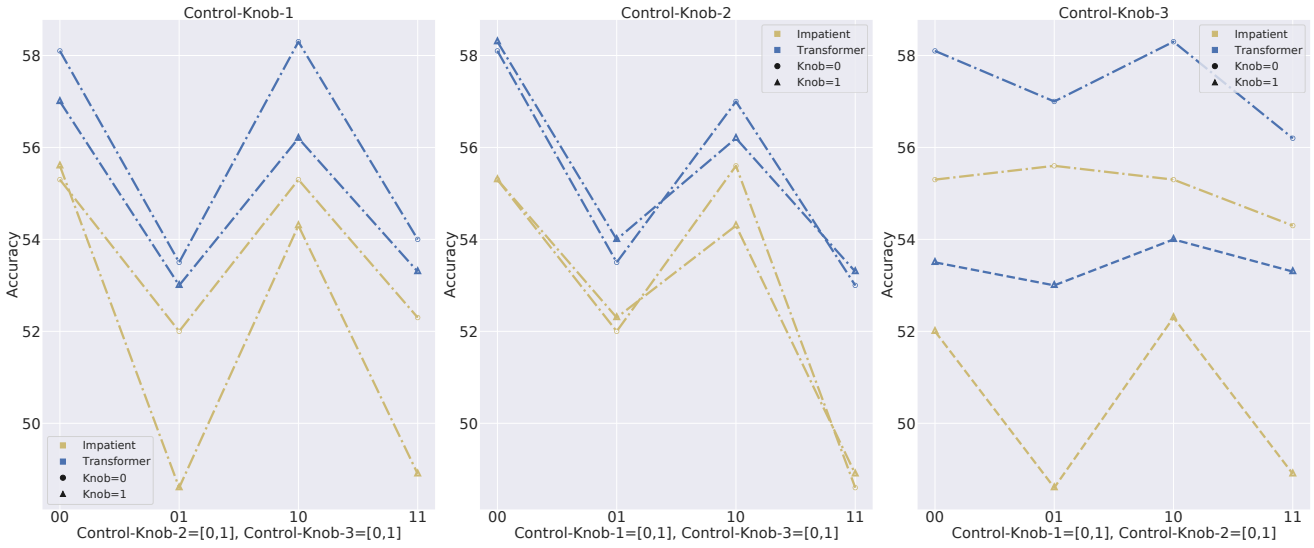

Figure 4. **Visual Coherence:** Impact of Control-Knobs by comparing the performance of different scoring functions (adapted baseline and transformer) for each knob setting. LM is set to Word2Vec and VM is set to ResNet–50. Starting from left we plot performance of Control-Knob-2 and Control-Knob-3 for all combination of control setting by fixing Control-Knob-1. We do the same for Control-Knob-2 and Control-Knob-3 in the center and right figure respectively.

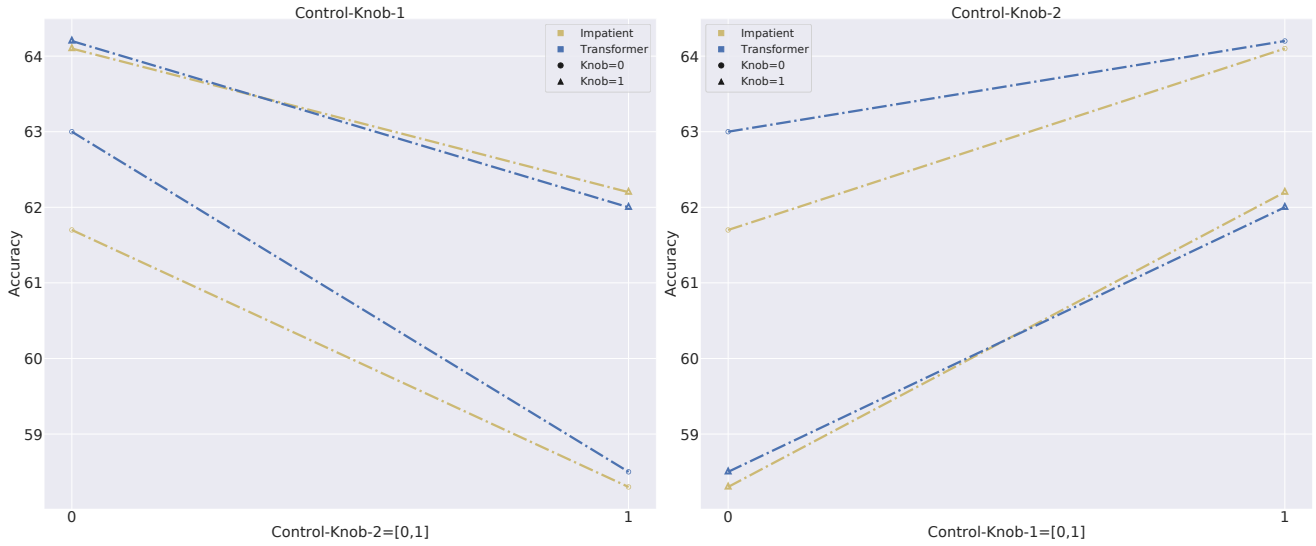

Figure 5. **Visual Ordering:** Impact of Control-Knobs by comparing the performance of different scoring functions (adapted baseline and transformer) for each knob setting. LM is set to Word2Vec and VM is set to ResNet–50. For each Control-Knob we fix one and plot the other, as show in the figure side by side.