Procedural Reasoning Networks for Understanding Multimodal Procedures

Mustafa Sercan Amac  Semih Yagcioglu  Aykut Erdem  Erkut Erdem
Hacettepe University Computer Vision Lab
Dept. of Computer Engineering, Hacettepe University, Ankara, TURKEY
{b21626915,n13242994,aykut,erkut}@cs.hacettepe.edu.tr

Abstract
This paper addresses the problem of comprehending procedural commonsense knowledge. This is a challenging task as it requires identifying key entities, keeping track of their state changes, and understanding temporal and causal relations. Contrary to most of the previous work, in this study, we do not rely on strong inductive bias and explore the question of how multimodality can be exploited to provide a complementary semantic signal. Towards this end, we introduce a new entity-aware neural comprehension model augmented with external relational memory units. Our model learns to dynamically update entity states in relation to each other while reading the text instructions. Our experimental analysis on the visual reasoning tasks in the recently proposed RecipeQA dataset reveals that our approach improves the accuracy of the previously reported models by a large margin. Moreover, we find that our model learns effective dynamic representations of entities even though we do not use any supervision at the level of entity states. \footnote{The project website with code and demo is available at https://hucvl.github.io/prn/}

1 Introduction
A great deal of commonsense knowledge about the world we live is procedural in nature and involves steps that show ways to achieve specific goals. Understanding and reasoning about procedural texts (e.g. cooking recipes, how-to guides, scientific processes) are very hard for machines as it demands modeling the intrinsic dynamics of the procedures (Bosselut et al., 2018; Dalvi et al., 2018; Yagcioglu et al., 2018). That is, one must be aware of the entities present in the text, infer relations among them and even anticipate changes in the states of the entities after each action. For example, consider the cheeseburger recipe presented in Fig. 1. The instruction “salt and pepper each patty and cook for 2 to 3 minutes on the first side” in Step 5 entails mixing three basic ingredients, the ground beef, salt and pepper, together and then applying heat to the mix, which in turn causes chemical changes that alter both the appearance and the taste. From a natural language understanding perspective, the main difficulty arises when a model sees the word patty again at a later stage of the recipe. It still corresponds to the same entity, but its form is totally different.

Over the past few years, many new datasets and approaches have been proposed that address this inherently hard problem (Bosselut et al., 2018; Dalvi et al., 2018; Tandon et al., 2018; Du et al., 2019). To mitigate the aforementioned challenges, the existing works rely mostly on heavy supervision and focus on predicting the individual state changes of entities at each step. Although these models can accurately learn to make local predictions, they may lack global consistency (Tandon et al., 2018; Du et al., 2019), not to mention that building such annotated corpora is very labor-intensive. In this work, we take a different direction and explore the problem from a multimodal standpoint. Our basic motivation, as illustrated in Fig. 1, is that accompanying images provide complementary cues about causal effects and state changes. For instance, it is quite easy to distinguish raw meat from cooked one in visual domain.

In particular, we take advantage of recently proposed RecipeQA dataset (Yagcioglu et al., 2018), a dataset for multimodal comprehension of cooking recipes, and ask whether it is possible to have a model which employs dynamic representations of entities in answering questions that require multimodal understanding of procedures. To this end, inspired from (Santoro et al., 2018), we propose Procedural Reasoning Networks (PRN) that incorporates entities into the comprehension process and al-
Step 1: Ingredients and Tools

- 4 oz. seasoned ground beef (25-30% fat)
- 1 hamburger bun
- 1 large onion
- 2 slices of American cheese
- Thousand Island dressing

Step 2: Form Patties

Begin by preheating a cast iron skillet over medium heat. Make four patties by placing 2-ounce portions of seasoned ground beef in the pan. Cook each patty for 2-3 minutes, flip, and cook for another 2-3 minutes on the other side.

Step 3: Season

Lightly salt the bottom sides of the patties. Once the other side is done, set aside.

Step 4: Toast Buns

Toast the bottom half of the hamburger buns in the pan. Let cool.

Step 5: Cook

Set the patties aside and toast the bottom half of the hamburger buns in the pan. Let cool.

Step 6: Chop Onions & Tomatoes

For the "authentic" feel you want to get a large handful of fresh lettuce. Slice it into leaves, and place it in the pan. Set aside.

Step 7: Chap Onion & Tomato

Assemble the burger in the following order from the bottom up: bottom bun, lettuce, cheese, tomato, onion, top bun. Place on a platter, cover with foil, and serve.

Step 8: Enjoy

At that’s left to do is enjoy the copious double decker. To be honest, this was impressively close to the real taste. I would definitely make this one again.

For the "authentic" feel you want to get a large handful of fresh lettuce. Slice into leaves, and place it in the pan. Set aside.

Assemble the burger in the following order from the bottom up: bottom bun, lettuce, cheese, tomato, onion, top bun. Place on a platter, cover with foil, and serve.

Figure 1: A recipe for preparing a cheeseburger (adapted from the cooking instructions available at https://www.instructables.com/id/In-N-Out-Double-Double-Cheesburger-Copycat/). Each basic ingredient (entity) is highlighted by a different color in the text and with bounding boxes on the accompanying images. Over the course of the recipe instructions, ingredients interact with each other, change their states by each cooking action (underlined in the text), which in turn alter the visual and physical properties of entities. For instance, the tomato changes its form by being sliced up and then stacked on a hamburger bun.

allows to keep track of entities, understand their interactions and accordingly update their states across time. We report that our proposed approach significantly improves upon previously published results on visual reasoning tasks in RecipeQA, which test understanding causal and temporal relations from images and text. We further show that the dynamic entity representations can capture semantics of the state information in the corresponding steps.

2 Visual Reasoning in RecipeQA

In our study, we particularly focus on the visual reasoning tasks of RecipeQA, namely visual cloze, visual coherence, and visual ordering tasks, each of which examines a different reasoning skill. We briefly describe these tasks below.

**Visual Cloze.** In the visual cloze task, the question is formed by a sequence of four images from consecutive steps of a recipe where one of them is replaced by a placeholder. A model should select the correct one from a multiple-choice list of four answer candidates to fill in the missing piece. In that regard, the task inherently requires aligning visual and textual information and understanding temporal relationships between the cooking actions and the entities.

**Visual Coherence.** The visual coherence task tests the ability to identify the image within a sequence of four images that is inconsistent with the text instructions of a cooking recipe. To succeed in this task, a model should have a clear understanding of the procedure described in the recipe and at the same time connect language and vision.

**Visual Ordering.** The visual ordering task is about grasping the temporal flow of visual events with the help of the given recipe text. The questions show a set of four images from the recipe and the task is to sort jumbled images into the correct order. Here, a model needs to infer the temporal relations between the images and align them with the recipe steps.

3 Procedural Reasoning Networks

In the following, we explain our Procedural Reasoning Networks model. Its architecture is based on a bi-directional attention flow (BiDAF) model (Gardner et al., 2018), but also equipped with an explicit reasoning module that acts on entity-specific rela-

---

2We intentionally leave the textual cloze task out from our experiments as the questions in this task do not necessarily need multimodality.

3Our implementation is based on the implementation publicly available in AllenNLP (Gardner et al., 2018).
Step 1: Ingredients

- Pumpkin puree
- Cream cheese
- Sugar
- Eggs
- Vanilla extract
- Nutmeg
- Ginger
- Cinnamon
- Flour
- Salt
- Freshly ground pepper

Step 2: The Crust

- Preheat the oven to 350F (180C).
- Place the gingersnap cookies in a food processor or a mallet (or liner bag).
- Crush into crumbs! Add butter to crumbs and process until well incorporated.

Step 3: The Filling

- In a mixer or food processor, combine the cream cheese, pumpkin puree, sugar, eggs, vanilla extract until well incorporated. (If you're using the mallet method, you can use a mallet and a baggie - go for it!), turn your gingersnaps into crumbs! Add butter to crumbs and process until well incorporated. (If you're using the mallet method, you can use a mallet and a baggie - go for it!)

Step 4: Bake

- Bake the pumpkin cheesecake for 80-90 minutes, until the center is almost set, and loosen the crust from the edge of the pan. Allow to cool before removing the rim of the pan.

At a high level, as the model is reading the cooking recipe, it continually updates the internal memory representations of the entities (ingredients) based on the content of each step – it keeps track of changes in the states of the entities, providing an entity-centric summary of the recipe. The response to a question and a possible answer depends on the representation of the recipe text as well as the last states of the entities. All this happens in a series of implicit relational reasoning steps and there is no need for explicitly encoding the state in terms of a predefined vocabulary.

3.1 Input Module

Let the triple \((R, Q, A)\) be a sample input. Here, \(R\) denotes the input recipe which contains textual instructions composed of \(N\) words in total. \(Q\) represents the question that consists of a sequence of \(M\) images. \(A\) denotes an answer that is either a single image or a series of \(L\) images depending on the reasoning task. In particular, for the visual cloze and the visual coherence type questions, the answer contains a single image \((L = 1)\) and for the visual ordering task, it includes a sequence.

We encode the input recipe \(R\) at character, word, and step levels. Character-level embedding layer uses a convolutional neural network, namely Char-CNN model by Kim (2014), which outputs character level embeddings for each word and alleviates the issue of out-of-vocabulary (OOV) words. In word embedding layer, we use a pretrained GloVe model (Pennington et al., 2014) and extract word-level embeddings\(^4\). The concatenation of the character and the word embeddings are then fed to a two-layer highway network (Srivastava et al., 2015) to obtain a contextual embedding for each word in the recipe. This results in the matrix \(R' \in \mathbb{R}^{2d \times N}\).

On top of these layers, we have another layer that encodes the steps of the recipe in an individual manner. Specifically, we obtain a step-level con-

\(^4\)We also consider pretrained ELMo embeddings (Peters et al., 2018) in our experiments but found out that the performance gain does not justify the computational overhead.
textual embedding of the input recipe containing $T$ steps as $S = (s_1, s_2, \ldots, s_T)$ where $s_i$ represents the final state of a BiLSTM encoding the $i$-th step of the recipe obtained from the character and word-level embeddings of the tokens exist in the corresponding step.

We represent both the question $Q$ and the answer $A$ in terms of visual embeddings. Here, we employ a pretrained ResNet-50 model (He et al., 2016) trained on ImageNet dataset (Deng et al., 2009) and represent each image as a real-valued 2048-d vector using features from the penultimate average-pool layer. Then these embeddings are passed first to a multilayer perceptron (MLP) and then its outputs are fed to a BiLSTM. We then form a matrix $Q' \in \mathbb{R}^{2d \times M}$ for the question by concatenating the cell states of the BiLSTM. For the visual ordering task, to represent the sequence of images in the answer with a single vector, we additionally use a BiLSTM and define the answering embedding by the summation of the cell states of the BiLSTM. Finally, for all tasks, these computations produce answer embeddings denoted by $a \in \mathbb{R}^{2d \times 1}$.

3.2 Reasoning Module

As mentioned before, comprehending a cooking recipe is mostly about entities (basic ingredients) and actions (cooking activities) described in the recipe instructions. Each action leads to changes in the states of the entities, which usually affects their visual characteristics. A change rarely occurs in isolation; in most cases, the action affects multiple entities at once. Hence, in our reasoning module, we have an explicit memory component implemented with relational memory units (Santoro et al., 2018). This helps us to keep track of the entities, their state changes and their relations in relation to each other over the course of the recipe (see Fig. 3). As we will examine in more detail in Section 4, it also greatly improves the interpretability of model outputs.

Specifically, we set up the memory with a memory matrix $E \in \mathbb{R}^{d_e \times K}$ by extracting $K$ entities (ingredients) from the first step of the recipe\(^5\). We initialize each memory cell $e_i$ representing a specific entity by its CharCNN and pre-trained GloVe embeddings\(^6\). From now on, we will use the terms memory cells and entities interchangeably throughout the paper. Since the input recipe is given in the form of a procedural text decomposed into a number of steps, we update the memory cells after each step, reflecting the state changes happened on the entities. This update procedure is modelled via a relational recurrent neural network (R-RNN), recently proposed by Santoro et al. (2018). It is built on a 2-dimensional LSTM model whose matrix of cell states represent our memory matrix $E$. Here, each row $i$ of the matrix $E$ refers to a specific entity $e_i$ and is updated after each recipe step $t$ as follows:

$$\phi_{i,t} = \text{R-RNN}(\phi_{i,t-1}, s_t)$$  \hspace{1cm} (1)

where $s_t$ denotes the embedding of recipe step $t$ and $\phi_{i,t} = (h_{i,t}, e_{i,t})$ is the cell state of the R-RNN at step $t$ with $h_{i,t}$ and $e_{i,t}$ being the $i$-th row of the hidden state of the R-RNN and the dynamic representation of entity $e_i$ at the step $t$, respectively.

The R-RNN model exploits a multi-headed self-attention mechanism (Vaswani et al., 2017) that allows memory cells to interact with each other and attend multiple locations simultaneously during the update phase.

In Fig. 3, we illustrate how this interaction takes place in our relational memory module by considering a sample cooking recipe and by presenting how the attention matrix changes throughout the recipe. In particular, the attention matrix at a specific time shows the attention flow from one entity (memory cell) to another along with the attention weights to the corresponding recipe step (offset column). The color intensity shows the magnitude of the attention weights. As can be seen from the figure, the internal representations of the entities are actively updated at each step. Moreover, as argued in (Santoro et al., 2018), this can be interpreted as a form of relational reasoning as each update on a specific memory cell is operated in relation to others. Here, we should note that it is often difficult to make sense of these attention weights. However, we observe that the attention matrix changes very gradually near the completion of the recipe.

3.3 Attention Module

Attention module is in charge of linking the question with the recipe text and the entities present in the recipe. It takes the matrices $Q'$ and $R'$ from the input module, and $E$ from the reasoning module vector of all the words.
The attention weight is calculated by \( a \) where the attention weight is computed by \( \phi \).

The question-to-recipe matrix is then obtained by replicating \( \text{softmax}(\max \text{col}(S^R)) \) across the column, giving \( R^r \in \mathbb{R}^{2d \times N} \).

Then, we construct the question aware representation of the input recipe, \( G \), with its \( i \)-th column \( G_i \in \mathbb{R}^{d \times N} \) denoting the final embedding of \( i \)-th word given by

\[
G_i = [R'_i; \hat{Q}_i; R'_i \circ \hat{Q}_i; R'_i \circ \hat{R}_i] .
\]  

Attentions from question to entities, and from entities to question are computed in a way similar to the ones described above. The only difference is that it uses a different shared affinity matrix to be computed between the memory encoding entities \( E \) and the question \( Q \). These attentions are then used to construct the question aware representation of entities, denoted by \( Y \), that links and integrates the images in the question and the entities in the input recipe.

### 3.4 Modeling Module

Modeling module takes the question-aware representations of the recipe \( G \) and the entities \( Y \), and forms their combined vector representation. For this purpose, we first use a two-layer BiLSTM to read the question-aware recipe \( G \) and to encode the interactions among the words conditioned on the question. For each direction of BiLSTM, we use its hidden state after reading the last token as its output. In the end, we obtain a vector embedding \( e \in \mathbb{R}^{2d \times 1} \). Similarly, we employ a second BiLSTM, this time, over the entities \( Y \), that links and integrates the images in the question and the entities in the input recipe.
3.5 Output Module

The output module takes the output of the modeling module, encoding vector embeddings of the question-aware recipe and the entities $Y$, and the embedding of the answer $A$, and returns a similarity score which is used while determining the correct answer. Among all the candidate answers, the one having the highest similarity score is chosen as the correct answer. To train our proposed procedural reasoning network, we employ a hinge ranking loss (Collobert et al., 2011), similar to the one used in (Yagcioglu et al., 2018), given below.

$$L = \max\{0, \gamma - \cos(o, a_+) + \cos(o, a_-)\} \quad (4)$$

where $\gamma$ is the margin parameter, $a_+$ and $a_-$ are the correct and the incorrect answers, respectively.

4 Experiments

In this section, we describe our experimental setup and then analyze the results of the proposed Procedural Reasoning Networks (PRN) model.

4.1 Entity Extraction

Given a recipe, we automatically extract the entities from the initial step of a recipe by using a dictionary of ingredients. While determining the ingredients, we exploit Recipe1M (Marin et al., 2018) and KaggleWhats Cooking Recipes (Yummly, 2015) datasets, and form our dictionary using the most commonly used ingredients in the training set of RecipeQA. For the cases when no entity can be extracted from the recipe automatically (20 recipes in total), we manually annotate those recipes with the related entities.

4.2 Training Details

In our experiments, we separately trained models on each task, as well as we investigated multi-task learning where a single model is trained to solve all these tasks at once. In total, the PRN architecture consists of ~12M trainable parameters. We implemented our models in PyTorch (Paszke et al., 2017) using AllenNLP library (Gardner et al., 2018). We used Adam optimizer with a learning rate of 1e-4 with an early stopping criteria with the patience set to 10 indicating that the training procedure ends after 10 iterations if the performance would not improve. We considered a batch size of 32 due to our hardware constraints. In the multi-task setting, batches are sampled round-robin from all tasks, where each batch is solely composed of examples from one task. We performed our experiments on a system containing four NVIDIA GTX-1080Ti GPUs, and training a single model took around 2 hours. We employed the same hyperparameters for all the baseline systems. We plan to share our code and model implementation after the review process.

4.3 Baselines

We compare our model with several baseline models as described below. We note that the results of the first two are previously reported in (Yagcioglu et al., 2018).

Hasty Student (Yagcioglu et al., 2018) is a heuristics-based simple model which ignores the recipe and gives an answer by examining only the question and the answer set using distances in the visual feature space.

Impatient Reader (Hermann et al., 2015) is a simple neural model that takes its name from the fact that it repeatedly computes attention over the recipe after observing each image in the query.

BiDAF (Seo et al., 2017a) is a strong reading comprehension model that employs a bi-directional attention flow mechanism to obtain a question-aware representation and bases its predictions on this representation. Originally, it is a span-selection model from the input context. Here, we adapt it to work in a multimodal context and answer multiple choice questions instead.

BiDAF w/ static memory is an extended version of the BiDAF model which resembles our proposed PRN model in that it includes a memory unit for the entities. However, it does not make any updates on the memory cells. That is, it uses the static entity embeddings initialized with GloVe word vectors. We propose this baseline to test the significance of the use of relational memory updates.

4.4 Results

Table 1 presents the quantitative results for the visual reasoning tasks in RecipeQA. In single-task training setting, PRN gives state-of-the-art results compared to other neural models. Moreover, it achieves the best performance on average. These results demonstrate the importance of having a dynamic memory and keeping track of entities extracted from the recipe. In multi-task training set-
Figure 4: t-SNE visualizations of learned embeddings from each memory snapshot mapping to each entity and their corresponding states from each step for visual cloze task.

| Model                  | Single-task Training | Multi-task Training |
|------------------------|----------------------|---------------------|
|                        | Cloze   | Coherence | Ordering | Average | Cloze   | Coherence | Ordering | All   |
| Human*                 | 77.60  | 81.60     | 64.00     | 74.40   | –      | –        | –        | –     |
| Hasty Student          | 27.35  | **65.80** | 40.88     | 44.68   | –      | –        | –        | –     |
| Impatient Reader       | 27.36  | 28.08     | 26.74     | 27.39   | –      | –        | –        | –     |
| BIDAF                  | 53.95  | 48.82     | 62.42     | 55.06   | 44.62  | 36.00    | 63.93    | 48.67 |
| BIDAF w/ static memory | 51.82  | 45.88     | 60.90     | 52.87   | **47.81** | 40.23    | 62.94    | **50.59** |
| PRN                    | **56.31** | 53.64     | **62.77** | **57.57** | 46.45  | **40.58** | 62.67    | 50.17 |

* Taken from the RecipeQA project website, based on 100 questions sampled randomly from the validation set.

Table 1: Quantitative comparison of the proposed PRN model against the baselines.

In Fig. 4, we illustrate the entity embeddings space by projecting the learned embeddings from the step-by-step memory snapshots through time with t-SNE to 3-d space from 200-d vector space. Color codes denote the categories of the cooking recipes. As can be seen, these step-aware embeddings show clear clustering of these categories. Moreover, within each cluster, the entities are grouped together in terms of their state characteristics. For instance, in the zoomed parts of the figure, chopped and sliced, or stirred and whisked entities are placed close to each other.

Fig. 5 demonstrates the entity arithmetics using the learned embeddings from each entity step. Here, we show that the learned embedding from the memory snapshots can effectively capture the contextual information about the entities at each time point in the corresponding step while taking into account of the recipe data. This basic arithmetic operation suggests that the proposed model can successfully capture the semantics of each entity’s state in the corresponding step.

5 Related Work

In recent years, tracking entities and their state changes have been explored in the literature from a variety of perspectives. In an early work, Henaff et al. (2017) proposed a dynamic memory based network which updates entity states using a gating mechanism while reading the text. Bansal et al. (2017) presented a more structured memory augmented model which employs memory slots for representing both entities and their relations. Pavez et al. (2018) suggested a conceptually similar model in which the pairwise relations between attended memories are utilized to encode the world.
Step 1: Ingredients
Place the tomatoes and onion as thin as is reasonable, slice the garlic as thin as you can. Thoroughly mix the onion and slice the bell pepper. Then add the butter and add 1/3 cup milk. When the onion and bell pepper are well cooked, add the prosciutto and chopped prosciutto in addition to the wine. Be creative/why not, add even more in addition to the wine. Be creative/why not.

Step 1: Gathering Your Ingredients...

Onions (Flowerpot Chicken)

Step 1: Prepare Vegetables.
Chop your vegetables while the pot is cooking. I consider the potatoes the most important part of this recipe, so try to chop them to the same size. Picture 3. Lastly place the bell pepper. I know you have been cooking in the oven, but not on your own. Finely take off the outer skin of the pepper, slice them then chop them into strips, then simply lightly heat them up as you like.

Step 1: Potato Prep + Seasonings

Water (Caribbean Curried Goat)

Step 4: Add Everything Else.
Simmer the broth, and then add the wine. Be sure to check the meat and water. 1/2 cup all full cup of water near the pot to add a bit of flavor. If you need the meat to stand up for the entire cooking process. In the picture below I've dropped in another level of the curry powder and stir things about. When it starts to turn the heat down and add a bit of water into the onions. It is recommended to be used in the onions. The vegetable really doesn't matter too much in this dish. A 4 work long enough that you've got LOTS of water to be added and ready.

Milk (Flowerpot Chicken)

Step 2: Cooking.
Heads the butter and add 1/2 cup chopped onions. When the onions are dark, add the remaining vegetables. In the picture below I've used a family of vegetables. Add the rest of the curry powder and stir things about. When it starts to turn the heat down and add a bit of water into the onions. It is recommended to be used in the onions. The vegetable really doesn't matter too much in this dish. A 4 work long enough that you've got LOTS of water to be added and ready.

Milk (Family Size Lasagne)

Step 2: West Coast.
Pour over 130 degrees Rule. Never add the mixture in a large pan. Stirring on the flat portion of the measuring spoon may not need a little more. Scoop the mixture into some paper towel to be used in the pot. Place the mixture in a large bowl of chopped potatoes. Be creative/why not, add even more. Be creative/why not. Be creative/why not. Cook the mixture in the same manner, mixing, baking and serving for at least 30 minutes.

Tomatoes (Flowerpot Chicken)

Step 1: This is a cheap and easy method of an ancient cooking technique known as pot cooking using a common terra cotta flowerpot and saucer. You can spend over $100 on a clay cooker or you can use a flower pot and saucer for free. Picture 1. Finely chop the garlic, you want it lumpy but the potatoes will help the potato mixture. I use a potato masher or you can just use a fork. You still want the destination of the garden supply. You choose. Some of you may already have the pot lying in your garden or on a shelf. Once you put the plant will probably be cooking all kinds of things in it.

Tomatoes (Seven Layer Seven Grain Bread)

Step 1: Ingredients
Pepperoni (used what was left in a package which was enough for one layer) 1/3 cup water tomatoes onion as thin as is reasonable slice the garlic as thin as you can. Thoroughly mix the onion and slice the bell pepper. Then add the butter and add 1/3 cup milk. When the onion and bell pepper are well cooked, make sure all potatoes are ready. Picture 1. Lastly place the bell pepper. I know you have been cooking in the oven, but not on your own. Finely take off the outer skin of the pepper, slice them then chop them into strips, then simply lightly heat them up as you like.

Tomatoes (How to Make Chicken Cacciatore)

Step 1: Gather Your Ingredients...

Water (Caribbean Curried Goat)

Step 4: Add Everything Else.
Simmer the broth, and then add the wine. Be sure to check the meat and water. 1/2 cup all full cup of water near the pot to add a bit of flavor. If you need the meat to stand up for the entire cooking process. In the picture below I've dropped in another level of the curry powder and stir things about. When it starts to turn the heat down and add a bit of water into the onions. It is recommended to be used in the onions. The vegetable really doesn't matter too much in this dish. A 4 work long enough that you've got LOTS of water to be added and ready.

Milk (Flowerpot Chicken)

Step 2: Cooking.
Heads the butter and add 1/2 cup chopped onions. When the onions are dark, add the remaining vegetables. In the picture below I've used a family of vegetables. Add the rest of the curry powder and stir things about. When it starts to turn the heat down and add a bit of water into the onions. It is recommended to be used in the onions. The vegetable really doesn't matter too much in this dish. A 4 work long enough that you've got LOTS of water to be added and ready.

Milk (Family Size Lasagne)

Step 2: West Coast.
Pour over 130 degrees Rule. Never add the mixture in a large pan. Stirring on the flat portion of the measuring spoon may not need a little more. Scoop the mixture into some paper towel to be used in the pot. Place the mixture in a large bowl of chopped potatoes. Be creative/why not, add even more. Be creative/why not. Cook the mixture in the same manner, mixing, baking and serving for at least 30 minutes.

Tomatoes (Flowerpot Chicken)

Step 1: This is a cheap and easy method of an ancient cooking technique known as pot cooking using a common terra cotta flowerpot and saucer. You can spend over $100 on a clay cooker or you can use a flower pot and saucer for free. Picture 1. Finely chop the garlic, you want it lumpy but the potatoes will help the potato mixture. I use a potato masher or you can just use a fork. You still want the destination of the garden supply. You choose. Some of you may already have the pot lying in your garden or on a shelf. Once you put the plant will probably be cooking all kinds of things in it.

Tomatoes (Seven Layer Seven Grain Bread)

Step 1: Ingredients
Pepperoni (used what was left in a package which was enough for one layer) 1/3 cup water tomatoes onion as thin as is reasonable slice the garlic as thin as you can. Thoroughly mix the onion and slice the bell pepper. Then add the butter and add 1/3 cup milk. When the onion and bell pepper are well cooked, make sure all potatoes are ready. Picture 1. Lastly place the bell pepper. I know you have been cooking in the oven, but not on your own. Finely take off the outer skin of the pepper, slice them then chop them into strips, then simply lightly heat them up as you like.

Tomatoes (How to Make Chicken Cacciatore)

Step 1: Gather Your Ingredients...

Water (Caribbean Curried Goat)

Step 4: Add Everything Else.
Simmer the broth, and then add the wine. Be sure to check the meat and water. 1/2 cup all full cup of water near the pot to add a bit of flavor. If you need the meat to stand up for the entire cooking process. In the picture below I've dropped in another level of the curry powder and stir things about. When it starts to turn the heat down and add a bit of water into the onions. It is recommended to be used in the onions. The vegetable really doesn't matter too much in this dish. A 4 work long enough that you've got LOTS of water to be added and ready.

Milk (Flowerpot Chicken)

Step 2: Cooking.
Heads the butter and add 1/2 cup chopped onions. When the onions are dark, add the remaining vegetables. In the picture below I've used a family of vegetables. Add the rest of the curry powder and stir things about. When it starts to turn the heat down and add a bit of water into the onions. It is recommended to be used in the onions. The vegetable really doesn't matter too much in this dish. A 4 work long enough that you've got LOTS of water to be added and ready.

Milk (Family Size Lasagne)

Step 2: West Coast.
Pour over 130 degrees Rule. Never add the mixture in a large pan. Stirring on the flat portion of the measuring spoon may not need a little more. Scoop the mixture into some paper towel to be used in the pot. Place the mixture in a large bowl of chopped potatoes. Be creative/why not, add even more. Be creative/why not. Cook the mixture in the same manner, mixing, baking and serving for at least 30 minutes.

Tomatoes (Flowerpot Chicken)

Step 1: This is a cheap and easy method of an ancient cooking technique known as pot cooking using a common terra cotta flowerpot and saucer. You can spend over $100 on a clay cooker or you can use a flower pot and saucer for free. Picture 1. Finely chop the garlic, you want it lumpy but the potatoes will help the potato mixture. I use a potato masher or you can just use a fork. You still want the destination of the garden supply. You choose. Some of you may already have the pot lying in your garden or on a shelf. Once you put the plant will probably be cooking all kinds of things in it.
nism to keep track of the changes in the states of the entities over the course of the procedure. Our experimental analysis on visual reasoning tasks in the RecipeQA dataset shows that the model significantly improves the results of the previous models, indicating that it better understands the procedural text and the accompanying images. Additionally, we carefully analyze our results and find that our approach learns meaningful dynamic representations of entities without any entity-level supervision. Although we achieve state-of-the-art results on RecipeQA, clearly there is still room for improvement compared to human performance. We also believe that the PRN architecture will be of value to other visual and textual sequential reasoning tasks.

Acknowledgements

We thank the anonymous reviewers and area chairs for their invaluable feedback. This work was supported by TUBA GEBIP fellowship awarded to E. Erdem; and by the MMVC project via an Institutional Links grant (Project No. 217E054) under the Newton-Katip Çelebi Fund partnership funded by the Scientific and Technological Research Council of Turkey (TUBITAK) and the British Council. We also thank NVIDIA Corporation for the donation of GPUs used in this research.

References

Traptit Bansal, Arvind Neelakantan, and Andrew McCallum. 2017. RelNet: End-to-End Modeling of Entities & Relations. In NeurIPS Workshop on Automated Knowledge Base Construction (AKBC).

Antoine Bosselut, Corin Ennis, Omer Levy, Ari Holtzman, Dieter Fox, and Yejin Choi. 2018. Simulating Action Dynamics with Neural Process Networks. In Proceedings of the International Conference on Learning Representations (ICLR).

Danqi Chen, Jason Bolton, and Christopher D Manning. 2016. A thorough examination of the CNN/Daily Mail Reading Comprehension Task. In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), pages 2358–2367.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural Language Processing (Almost) from Scratch. Journal of Machine Learning Research, 12:2493–2537.

Bhavana Dalvi, Lifu Huang, Niket Tandon, Wen-tau Yih, and Peter Clark. 2018. Tracking state changes in procedural text: a challenge dataset and models for process paragraph comprehension. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.

Rajarshi Das, Tsenduren Munkhdalai, Xingdi Yuan, Adam Trischler, and Andrew McCallum. 2019. Building Dynamic Knowledge Graphs from Text using Machine Reading Comprehension. In Proceedings of the International Conference on Learning Representations (ICLR).

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 248–255.

Bhuwan Dhingra, Qiao Jin, Zhihlin Yang, William W Cohen, and Ruslan Salakhutdinov. 2018. Neural Models for Reasoning over Multiple Mentions using Coreference. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT).

Xinya Du, Bhavana Dalvi Mishra, Niket Tandon, Antoine Bosselut, Wen-tau Yih, Peter Clark, and Claire Cardie. 2019. Be consistent! improving procedural text comprehension using label consistency. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT).

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In Proceedings of Workshop for NLP Open Source Software (NLP-OSS), pages 1–6, Melbourne, Australia. Association for Computational Linguistics.

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, pages 770–778.

Mikael Henaff, Jason Weston, Arthur Szlam, Antoine Bordes, and Yann LeCun. 2017. Tracking The World State with Recurrent Entity Networks. In Proceedings of the International Conference on Learning Representations (ICLR).

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching Machines to Read and Comprehend. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), pages 1693–1701.

Schmidhuber J. Hochreiter, S. 1997. Long Short-Term Memory. Neural computation, 9(8):1735–1780.
Mohit Iyyer, Varun Manjunatha, Anupam Guha, Yogarshi Vyas, Jordan Boyd-Graber, Hal Daumé III, and Larry Davis. 2017. The amazing mysteries of the gutter: Drawing inferences between panels in comic book narratives. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Yangfeng Ji, Chenhao Tan, Sebastian Martstach, Yejin Choi, and Noah A Smith. 2017. Dynamic Entity Representations in Neural Language Models. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).

Robin Jia and Percy Liang. 2017. Adversarial Examples for Evaluating Reading Comprehension Systems. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).

Samira Ebrahimi Kahou, Adam Atkinson, Vincent Michalski, Akos Kadár, Adam Trischler, and Yoshua Bengio. 2017. FigureQA: An Annotated Figure Dataset for Visual Reasoning. In Proceedings of the International Conference on Learning Representations Workshop (ICLR Workshop).

Aniruddha Kembhavi, Minjoon Seo, Dustin Schwenk, Jonghyun Choi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Are You Smarter Than A Sixth Grader? Textbook Question Answering for Multimodal Machine Comprehension. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).

Javier Marin, Aritro Biswas, Ferda Ofli, Nicholas Hynes, Amaia Salvador, Yusuf Aytar, Ingmar Weber, and Antonio Torralba. 2018. Recipe1M: A Dataset for Learning Cross-Modal Embeddings for Cooking Recipes and Food Images. arXiv preprint arXiv:1810.06553.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. arXiv preprint arXiv:1806.08730.

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.

Juan Pavez, Hector Allende, and Hector Allende-Cid. 2018. Working memory networks: Augmenting memory networks with a relational reasoning module. In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), pages 1000–1009.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1532–1543.

Julien Perez and Fei Liu. 2017. Dialog state tracking, a machine reading approach using memory network. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pages 305–314.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), pages 2227–2237.

Adam Santoro, Ryan Faulkner, David Raposo, Jack Rae, Mike Chrzanowski, Theophane Weber, Daan Wierstra, Oriol Vinyals, Razvan Pascanu, and Timothy Lillicrap. 2018. Relational Recurrent Neural Networks. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), pages 7299–7310.

M. Schuster and K. K. Paliwal. 1997. Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing, 45(11):2673–2681.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017a. Bidirectional Attention Flow for Machine Comprehension. In Proceedings of the International Conference on Learning Representations (ICLR).

Minjoon Seo, Sewon Min, Ali Farhadi, and Hannaneh Hajishirzi. 2017b. Query-Reduction Networks for Question Answering. In Proceedings of the International Conference on Learning Representations (ICLR).

R. K. Srivastava, K. Greff, and J. Schmidhuber. 2015. Highway networks. In Proceedings of the International Conference on Machine Learning (ICML).

Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-To-End Memory Networks. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), pages 2440–2448.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2818–2826.

Niket Tandon, Bhavana Dalvi, Joel Grus, Wen-tau Yih, Antoine Bosselut, and Peter Clark. 2018. Reasoning about actions and state changes by injecting commonsense knowledge. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).
Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2016. MovieQA: Understanding Stories in Movies Through Question-Answering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 4631–4640.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), pages 5998–6008.

Hai Wang, Takeshi Onishi, Kevin Gimpel, and David McAllester. 2017. Emergent predication structure in hidden state vectors of neural readers. In Proceedings of the 2nd Workshop on Representation Learning for NLP, pages 26–36, Vancouver, Canada. Association for Computational Linguistics.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. 2016. Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks. In Proceedings of the International Conference on Learning Representations (ICLR).

Jason Weston, Sumit Chopra, and Antoine Bordes. 2015. Memory Networks. In Proceedings of the International Conference on Learning Representations (ICLR).

Semih Yagcioglu, Aykut Erdem, Erkut Erdem, and Nazli Ikizler-Cinbis. 2018. RecipeQA: A Challenge Dataset for Multimodal Comprehension of Cooking Recipes. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).

Yummly. 2015. Kaggle Whats Cooking? https://www.kaggle.com/c/whats-cooking/data. [Accessed: 2018-05-31].