RecipeQA: A Challenge Dataset for Multimodal Comprehension of Cooking Recipes

Semih Yagcioglu, Aykut Erdem, Erkut Erdem and Nazli Ikizler-Cinbis

Hacettepe University Computer Vision Lab

Dept. of Computer Engineering, Hacettepe University, Ankara, TURKEY

`semih.yagcioglu@hacettepe.edu.tr,{aykut,erkut,nazli}@cs.hacettepe.edu.tr`

Abstract

Understanding and reasoning about cooking recipes is a fruitful research direction towards enabling machines to interpret procedural text. In this work, we introduce RecipeQA, a dataset for multimodal comprehension of cooking recipes. It comprises of approximately 20K instructional recipes with multiple modalities such as titles, descriptions and aligned set of images. With over 36K automatically generated question-answer pairs, we design a set of comprehension and reasoning tasks that require joint understanding of images and text, capturing the temporal flow of events and making sense of procedural knowledge. Our preliminary results indicate that RecipeQA will serve as a challenging test bed and an ideal benchmark for evaluating machine comprehension systems. The data and leaderboard are available at http://hucvl.github.io/recipeqa.

1 Introduction

There is a rich literature in natural language processing (NLP) and information retrieval on question answering (QA) (Hirschman and Gaizauskas, 2001), but recently deep learning has sparked interest in a special kind of QA, commonly referred to as reading comprehension (RC) (Vanderwende, 2007). The aim in RC research is to build intelligent systems with the abilities to read and understand natural language text and answer questions related to it (Burges, 2013). Such tests are appealing as they require joint understanding of the question and the related passage (*i.e.* context), and moreover, they can analyze many different types of skills in a rather objective way (Sugawara et al., 2017).

Despite the progress made in recent years, there is still a significant performance gap between humans and deep neural models in RC, and researchers are pushing forward our understanding of the limitations and capabilities of these approaches by introducing new datasets. Existing tasks for RC mainly differ in two major respects: the question-answer formats, *e.g.* cloze (fill-in-the-blank), span selection or multiple choice, and the text sources they use, such as news articles (Hermann et al., 2015; Trischler et al., 2017), fictional stories (Hill et al., 2016), Wikipedia articles (Kočiský et al., 2018; Hewlett et al., 2016; Rajpurkar et al., 2016) or other web sources (Joshi et al., 2017). A popular topic in computer vision closely related to RC is Visual Question Answering (VQA) in which context takes the form of an image in the comprehension task, where recent datasets have also been compiled, such as (Antol et al., 2015; Yu et al., 2015; Johnson et al., 2017; Goyal et al., 2017), to name a few.

More recently, research in QA has been extended to focus on the multimodal aspects of the problem where different modalities are being explored. Tapaswi et al. (2016) introduced MovieQA where they concentrate on evaluating automatic story comprehension from both video and text. In COMICS, Iyyer et al. (2017) turned to comic books to test understanding of closure, transitions in the narrative from one panel to the next. In AI2D (Kembhavi et al., 2016) and FigureQA (Kahou et al., 2018), the authors addressed comprehension of scientific diagrams and graphical plots. Last but not least, Kembhavi et al. (2017) has proposed another comprehensive and challenging dataset named TQA, which comprised of middle school science lessons of diagrams and texts.

In this study, we focus on *multimodal comprehension of cooking recipes* with images and text. To this end, we introduce a new QA dataset called *RecipeQA* that consists of recipe instructions and related questions (see Fig. 1 for an example text cloze style question). There are a handful of reasons why understanding and reasoning about

Text Cloze Style Question	Context Modalities: Images and Descriptions of Steps

Recipe: Last-Minute Lasagna

1. Heat oven to 375 degrees F. Spoon a thin layer of sauce over the bottom of a 9-by-13-inch baking dish.
2. Cover with a single layer of ravioli.
3. Top with half the spinach half the mozzarella and a third of the remaining sauce.
4. Repeat with another layer of ravioli and the remaining spinach mozzarella and half the remaining sauce.
5. Top with another layer of ravioli and the remaining sauce not all the ravioli may be needed. Sprinkle with the Parmesan.
6. Cover with foil and bake for 30 minutes. Uncover and bake until bubbly, 5 to 10 minutes.
7. Let cool 5 minutes before spooning onto individual plates.

Step 1   Step 2   Step 3   Step 4

Step 5   Step 6   Step 7

Question	Choose the best text for the missing blank to correctly complete the recipe
	Cover. _____. Bake. Cool, serve.
Answer	**A. Top, sprinkle** B. Finishing touches C. Layer it up D. Ravioli bonus round

Figure 1: An illustrative text cloze style question (context, question and answer triplet). The context is comprised of recipe description and images where the question is generated using the question titles. Each paragraph in the context is taken from another step, as also true for the images. Bold answer is the correct one.

recipes is interesting. Recipes are written with a specific goal in mind, that is to teach others how to prepare a particular food. Hence, they contain immensely rich information about the real world. Recipes consist of instructions, wherein one needs to follow each instruction to successfully complete the recipe. As a classical example in introductory programming classes, each recipe might be seen as a particular way of solving a task and in that regard can also be considered as an algorithm. We believe that recipe comprehension is an elusive challenge and might be seen as important milestone in the long-standing goal of artificial intelligence and machine reasoning (Norvig, 1987; Bottou, 2014).

Among previous efforts towards multimodal machine comprehension (Tapaswi et al., 2016; Kembhavi et al., 2016; Iyyer et al., 2017; Kembhavi et al., 2017; Kahou et al., 2018), our study is closer to what Kembhavi et al. (2017) envisioned in TQA. Our task primarily differs in utilizing substantially larger number of images – the average number of images per recipe in RecipeQA is 12 whereas TQA has only 3 images per question on average. Moreover, in our case, each image is aligned with the text of a particular step in the corresponding recipe. Another important difference is that TQA contains mostly diagrams or textbook images whereas

RecipeQA consists of natural images taken by users in unconstrained environments.

Some of the important characteristics of RecipeQA are as follows:

- There are arbitrary numbers of steps in recipes and images in steps, respectively.
- There are different question styles, each requiring a specific comprehension skill.
- There exists high lexical and syntactic divergence between contexts, questions and answers.
- Answers require understanding procedural language, in particular keeping track of entities and/or actions and their state changes.
- Answers may need information coming from multiple steps (*i.e.* multiple images and multiple paragraphs).
- Answers inherently involve multimodal understanding of image(s) and text.

To sum up, we believe RecipeQA is a challenging benchmark dataset which will serve as a test bed for evaluating multimodal comprehension systems. In this paper, we present several statistical analyses on RecipeQA and also obtain baseline performances for a number of multimodal comprehension tasks that we introduce for cooking recipes.

2 RecipeQA Dataset

The Recipe Question Answering (RecipeQA) dataset is a challenging multimodal dataset that evaluates reasoning over real-life cooking recipes. It consists of approximately 20K recipes from 22 food categories, and over 36K questions. Fig. 2 shows an illustrative cooking recipe from our dataset. Each recipe includes an arbitrary number of steps containing both textual and visual elements. In particular, each step of a recipe is accompanied by a 'title', a 'description' and a set of illustrative 'images' that are aligned with the title and the description. Each of these elements can be considered as a different modality of the data. The questions in RecipeQA explore the multimodal aspects of the step-by-step instructions available in the recipes through a number of specific tasks that are described in Sec. 3, namely *textual cloze*, *visual cloze*, *visual coherency* and *visual ordering*.

2.1 Data Collection

We consider cooking recipes as the main data source for our dataset. These recipes were collected from Instructables[1], which is a how-to web site where users share all kinds of instructions including but not limited to recipes.

We employed a set of heuristics that helped us collect high quality data in an automatic manner. For instance, while collecting the recipes, we downloaded only the most popular recipes by considering the popularity as an objective measure for assessing the quality of a recipe. Our assumption is that the mostly viewed recipes contain less noise and include easy-to-understand instructions with high-quality illustrative images.

In total, we collected about 20K unique recipes from the food category of Instructables. We filtered out non-English recipes using a language identification (Lui and Baldwin, 2012), and automatically removed the ones with unreadable contents such as the ones that only contain recipe videos. Finally, as a post processing step, we normalized the description text by removing non-ASCII characters from the text.

2.2 Questions and Answers

For machine comprehension and reasoning, forming the questions and the answers is crucial for evaluating the ability of a model in understanding

the content. Prior studies employed natural language questions either collected via crowdsourcing platforms such as SQuAD (Rajpurkar et al., 2016) or generated synthetically as in CNN/Daily Mail (Hermann et al., 2015). Using natural language questions is a good approach in terms of capturing human understanding, but crowdsourcing is often too costly and does not scale well as the size of the dataset grows. Synthetic question generation is a low-cost solution, but the quality of the generated questions is subject to question.

RecipeQA includes structured data about the cooking recipes that consists of step-by-step instructions, which helps us generate questions in a fully automatic manner without compromising the quality. Our questions test the semantics of the instructions of the recipes from different aspects through the tasks described in Sec. 3. In particular, we generate a set of multiple choice questions (the number of choices is fixed as four) by following a simple procedure which apply to all of our tasks with slight modifications.

In order to generate question-answer-context triplets, we first filtered out recipes that contain less than 3 steps or more than 25 steps. We also ignored the initial step of the recipes as our preliminary analysis showed that the first step of the recipes almost always is used by the authors to provide a narrative, *e.g.* why they love making that particular food, or how it makes sense to prepare a food for some occasion, and often is not relevant to the recipe instructions. In addition, we automatically removed some indicators such as step numbers that explicitly emphasize temporal order from the step titles while generating questions.

Given a task, we first randomly select a set of steps from each recipe and construct our questions and answers from these steps according to the task at hand. In particular, we employ the modality that the comprehension task is built upon to generate the candidate answers and use the remaining content as the necessary context for our questions. For instance, if the step titles are used within the candidate answers, the context becomes the descriptions and the images of the steps. As the average number of steps per recipe is larger than four, using this strategy, we can generate multiple context-question-answer triplets from a single recipe.

Candidate answers can be generated by selecting the distractors at random from the steps of other recipes. To make our dataset more challenging, we

---

[1]All materials from the *instructables.com* were downloaded in April 2018.

Intro

This Creamy Coconut Chickpea Curry is an quick and easy to prepare vegan and gluten free Indian-cuisine-inspired dish, made from fresh ingredients. All it takes is about 5 minutes of prep time and another 20 minutes of cooking time and you have yourself a delicious and healthy dish. Deliciously satisfying!

Step 1: Ingredients

1 can (796mL) of chickpea curry 1 can (400mL) of coconut milk 2 tomatoes 1 lime 2 stalks of coriander 3 cloves of garlic 1 inch knob of ginger 1 large yellow onion 1/4 teaspoon ground black pepper 1/2 teaspoon salt 2 teaspoon curry powder 1/2 teaspoon paprika Flavourless oil like vegetable oil Tools: Cutting board Knife Skillet

Step 2: Prepping the Garlic, Ginger, Onion, and Tomato

Remove the skin from the garlic, ginger, and onion. I found it easiest to use a spoon to scrape the skin from the ginger. Mince the garlic and ginger. Dice the onion Dice the tomatoes. Once done, set aside the garlic and ginger, onions, and tomatoes on separate bowls respectively.

Step 3: Drain the Chickpeas

Drain the water from the can of chickpeas. Then run rinse the chickpeas under cold water, drain very well and leave aside. My chickpeas came with the transparent outer shells of the beans, so I removed those as well, then re-rinsed it before setting it aside.

Step 4: Prepping Lime and Coriander

You can prep the lime and the coriander while the curry is cooking because you will have time, but I find it easier to do all the prepping at once and leave the extra time for washing the dishes. Slice the lime to 6 wedges, these will be served with the curry. Chop the leaves off from the Coriander then roughly chop it to a smaller size as it will be used for garnishing.

Step 5: Cook the Onion, Garlic, and Ginger

Heat the oil in a skillet using medium heat and add in the diced onions. Cook it until the onion softens and becomes a translucent colour. This takes around 2 to 3 minutes. Once the onions are cooked, add your garlic and ginger in and cook for another 90 seconds.

Step 6: Add the Spices

Add in all the spices (curry powder, pepper, salt, and paprika) and stir it for about 30 seconds. This will cook the spices and infuse flavours of our spices together with the other ingredients.

Step 7: Add the Tomatoes

Add the tomatoes in and stir it around until it is mixed with the spices evenly. Then leave it to cook for another 3 to 5 minutes or until the tomatoes begin break down and harden. The tomatoes add a unique texture as well as a bit of sweetness and tartness to the dish.

Step 8: Add Chickpeas and Coconut Milk

Add the drained chickpeas to the skillet with the can of coconut milk. Stir it in until the curry and the coconut milk becomes uniformly mixed. Bring the heat down to medium-low and cover it for around 15 minutes to bring it to a boil until the sauce thickens up.

Step 9: Garnish and Serve

Garnish the dish with coriander and squeeze in a fresh lime on the curry to complete the dish and further elevate the flavour. Serve with with rice and enjoy!

Figure 2: A recipe of 'Creamy Coconut Chickpea Curry' with 9 steps, taken from Instructables.

employ a different strategy and select the distractors from the relevant modalities (titles, descriptions or images), which are not too far or too close from the correct answer. Specifically, we employ the following simple heuristic. We first find k nearest neighbors ($k = 100$) from other recipes. We then define an adaptive neighborhood by finding the closest distance to the query and remove the candidates that are too close. The remaining candidates are similar enough to be adversarial but not too similar to semantically substitute for the groundtruth. Finally, we randomly sample distractors from that pool. Details of the question generation procedure for each of the tasks are given in Sec. 3.

2.3 Dataset Statistics

RecipeQA dataset contains approximately 20K cooking recipes and over 36K question-answer pairs divided into four major question types reflecting each of the task at hand. The data is split into non-overlapping training, validation and test sets so that one set does not include a recipe and/or questions about that recipe which are available in other sets. There are 22 different food categories

	train	valid	test
# of recipes	15847	1963	1969
. . . avg. # of steps	5.99	6.01	6.00
. . . avg. # of tokens (titles)	17.79	17.40	17.67
. . . avg. # of tokens (descr.)	443.01	440.51	435.33
. . . avg. # of images	12.67	12.74	12.65
# of question-answers	29657	3562	3567
. . . textual cloze	7837	961	963
. . . visual cloze	7144	842	848
. . . visual coherence	7118	830	851
. . . visual ordering	7558	929	905

Table 1: RecipeQA dataset statistics.

across our dataset whose distribution is shown in Fig. 3. While splitting the recipes into sets, we take into account these categories so that all the sets have a similar distribution of recipes across all the categories. In Table 1, we show the detailed statistics about our RecipeQA dataset. Moreover, to visualize the token frequencies, we also provide the word clouds of the titles and the descriptions from the recipes in Fig. 4.
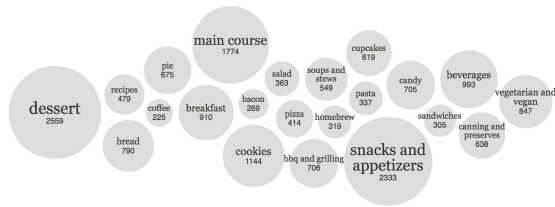
Figure 3: Distribution of the food categories across the RecipeQA.

Step titles   Step descriptions

Figure 4: Word clouds of the tokens for the titles and the descriptions of the recipes from RecipeQA.

3 Tasks

RecipeQA includes four different types of tasks: (1) Textual cloze, (2) Visual cloze, (3) Visual coherence, and (4) Visual ordering. Each of these tasks requires different reasoning skills as discussed in (Sugawara et al., 2017), and considers different modalities in their contexts and candidate answer sets. By modalities, we refer to the following pieces of information: (i) titles of steps, (ii) descriptions of steps and (iii) illustrative images of steps. While generating the questions for these tasks, we rather employ fixed templates as will be discussed below, which helps us to automatically construct question-answers pairs from the recipes with no human intervention. Using these tasks, we can easily evaluate complex relationships between different steps of a recipe via their titles, their descriptions and/or their illustrative images. Hence, our question-answers pairs are multimodal in nature. In the following, we provide a detailed description of each one of these tasks and discuss our strategies while selecting candidate answers.

3.1 Textual Cloze

Textual cloze style questions test the ability to infer missing text either in the title or in the step description by taking into account the question's context which includes a set of illustrative images besides text. While generating the question-answer pairs for this task, we randomly select a step from the candidate steps of a given recipe, hide its title and description, and ask for identifying this text amongst the multiple choices from the remaining modalities. To construct the distractor answers, we use the strategy in Sec. 2.2 that depends on the WMD (Kusner et al., 2015) distance measure. In Fig. 1, we provide a sample text cloze question from RecipeQA generated automatically in this way.

3.2 Visual Cloze

Visual cloze style questions test a skill similar to that of textual cloze task with the difference that the missing information in this task reside in the visual domain. Here, just like the textual cloze task, for a recipe we randomly select a step, hide its representative image, and ask to infer this image amongst the multiple choices. The context for this task is all textual and is in the form of a sequence of titles and descriptions. To construct the distractor images, we use Euclidean distances of 2048-d *pool5* features extracted from a ResNet-50 (He et al., 2016) pre-trained on ImageNet classification task. We show a sample visual cloze style question in Fig. 5 (second row).

3.3 Visual Coherence

Visual coherence style questions test the capability to identify an incoherent image in an ordered set of images given the titles and descriptions of the corresponding recipe as the context. Hence, to be successful at this task, a system needs to not only understand the relations between candidate steps, but also align and relate different modalities existing in the context and the answers. While generating the answer candidates for this task, we randomly select a single representative image from a single step and replace this image with a distractor image via employing the distractor selection strategy used for visual cloze task. In Fig. 5 (third row), we provide a sample visual coherence style question from RecipeQA.

3.4 Visual Ordering

Visual ordering questions test the ability of a system in finding a correctly ordered sequence given a jumbled set of representative images of a recipe. As in the previous visual tasks, the context of this task consists of the titles and descriptions of a recipe. To

Recipe: Bacon Sushi

Step 1: **What You'll Need** This recipe makes enough bacon sushi to feed 2 - 4 people. 2 x 500g(1 lb.) packages of bacon (I chose an applewood smoked bacon, but any type would work). 3 tbsp. oil. 1 medium onion, finely diced. 1 1...

Step 2: **Cooking the Bacon** The bacon "nori" will have to be partially cooked before it can be rolled with the risotto filling. Preheat the oven to 350 degrees F. Lay half a package of bacon on the rack of the roasting pan, then bak...

Step 3: **Making the Risotto Filling** I once made risotto with sushi rice, since I had no Arborio rice on hand, and I decided that the starchiness was similar in the two. My experiment was a success, and the resulting dish was just as deli...

Step 4: **Jazzing Up the Risotto** Risotto is a wonderfully customizable dish, and a quick search on the internet will result in a multitude of variations. Here are two of my favorites: Asian mushroom risotto. 1 tbsp. oil. 1 package...

Step 5: **Rolling the Sushi** Cover the sushi rolling mat with a large piece of aluminum foil as protection from the risotto and bacon grease. (You don't want your next sushi dinner tasting like bacon. Or maybe you do...) Lay the stri...

Step 6: **Baking and Slicing** Preheat the oven to 350 degrees F. Place the aluminum foil-covered sushi rolls in the oven and bake for 20 minutes. This will warm all the ingredients and crisp the bacon a little more. It will also melt a...

Step 7: **And You're Done!** Serve the sushi with a light crispy vegetable side dish, such as refreshing cucumber sticks, or a green salad. White wine makes an excellent compliment to the meal, especially if it is the same wine used in ...

Visual Cloze Style Question

Question Choose the best image for the missing blank to correctly complete the recipe

Answer A.   **B.**   C.   D.

Visual Coherence Style Question

Question Select the incoherent image in the following sequence of images
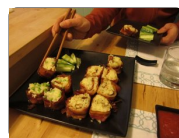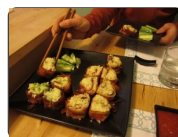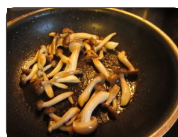
Answer A.   **B.**   C.   D.

Visual Ordering Style Question

Question Choose the correct order of the images to make a complete recipe

(i)   (ii)   (iii)   (iv)

Answer A. (iv)-(iii)-(ii)-(i)   **B. (iv)-(iii)-(i)-(ii)**   C. (i)-(ii)-(iii)-(iv)   D. (ii)-(iv)-(i)-(iii)

Figure 5: Sample visual cloze, visual coherence and visual ordering style questions (context, question and answer triplet) taken from the RecipeQA training set (Question Ids: 2000-3708-0-1-4-5, 3000-3708-2-3-4-6, 4000-3708-1-2-3-6). Here, the context is comprised of step titles and descriptions where the questions are generated using the images in the recipe. The correct answers are shown with green frames or in bold.

successfully complete this task, the system needs to understand the temporal occurrence of a sequence of recipe steps and infer temporal relations between candidates, *i.e.* boiling the water first, putting the spaghetti next, so that the ordered sequence of images aligns with the given recipe. To generate answer choices, we simply use random permutations of the illustrative images in the recipe steps. In

Fig. 5 (last row), we illustrate this visual ordering task through an example question. Here, we should note that a similar task has been previously investigated by Agrawal et al. (2016) for visual stories where the task is to order a jumbled set of aligned image-description pairs.

4 Experiments

4.1 Data Preparation

Ingredient Detection. We employed the method proposed in (Salvador et al., 2017) to detect recipe ingredients. To learn more effective word embeddings, we transformed the ingredients with compound words such as *olive oil* into single word ingredients with a proper hyphenation as *olive_oil*.

Textual Embeddings. We trained a distributed memory model, namely Doc2Vec (Le and Mikolov, 2014) and used it to learn word level and document level embeddings while encoding the semantic similarity by taking into account the word order within the provided context. In this way, we can represent each word, sentence or paragraph by a fixed sized vector. In our experiments, we employed 100-d vectors to represent all of the textual modalities (titles and descriptions). We made sure that the embeddings encode semantically useful information by exploring nearest neighbors (see Fig. 6 for some examples.)

Query	Nearest Neighbor
Then add the green onion and garlic.	Then add the white onion, red pepper and garlic.
It will thicken some while it cools	Some cornflour to thicken.
Slowly whisk in the milk, scraping the bottom and sides with a heatproof spatula to make sure all the dry ingredients are mixed in.	Stir the dry ingredients in, incrementally, mixing on low speed and scraping with a spatula after each addition.

Figure 6: Sample nearest neighbors from the embeddings by the trained Doc2Vec model.

Visual Features. We used the final activation of the ResNet-50 (He et al., 2016) model trained on the ImageNet dataset (Russakovsky et al., 2015) to extract 2048-d dense visual representations. Then, we further utilized an autoencoder to decrease the dimension of the visual features to 100-d so that they become compatible in size with the text embeddings.

4.2 Baseline Models

Neural Baselines. For our neural baselines, we adapted the Impatient Reader model in (Hermann et al., 2015), which was originally developed only for the cloze style text comprehension questions in the CNN/Daily Mail dataset. In our implementation, we used a uni-directional stacked LSTM architecture with 3 layers, in which we feed the context of the question to the network in a sequential manner. Particularly, we preserve the temporal order of the steps of the recipe while feeding it to the neural model, by mimicking the most common reading strategy – reading from top to bottom. For the multimodal setting, since images are represented with vectors which are of the same size with the text embeddings, we also feed the images to the network in the same order they are presented in the recipe.

In order to account for different question types, we employ a modular architecture, which requires small adjustments to be made for each task. For instance, we place the candidate answers into query for the cloze style questions or remove the candidate answer from the query for the visual coherence type questions. In training our Impatient Reader baseline model, we use a cosine similarity function and employed the *hinge ranking loss* (Collobert et al., 2011) as follows:

$$L = \max\{0, M - cos(q, a_+) + cos(q, a_-)\} \quad (1)$$

where M is a scalar denoting the margin, a_+ represents the ground truth answer, and a_- corresponds to an incorrect answer which is sampled randomly from the whole answer space. For all of our experiments, we select M as 1.5 and employ a simple heuristic to prevent overfitting by following an early stopping scheme with patience set to 10 against the validation set accuracy after the initial epoch. For the optimizer, we use ADAM and set the learning rate to $1e-3$. The training took around 18 to 24 hours on GTX 1080Ti on a single GPU. We did not perform any hyperparameter tuning.

Simple Baselines. We adapt the Hasty Student model described in (Tapaswi et al., 2016), which does not consider the provided context and simply answers questions by only looking at the similarities or the dissimilarities between the elements in questions and the candidate answers.

For the textual close task, each candidate answer is compared against the titles or descriptions of

	Visual Cloze	Textual Cloze	Visual Coherence	Visual Ordering
Hasty Student	27.35	26.89	**65.80**	**40.88**
Impatient Reader (Text only)	–	28.03	–	–
Impatient Reader (Multimodal)	**27.36**	**29.07**	28.08	26.74

Table 2: Results for simple and neural models on the test set of RecipeQA dataset.

Dataset	#Images	#Questions	Modality
COMICS	1.2M	750K	Image/Text
MovieQA	408	14,944	Image/Video/Text
TQA	3,455	26,260	Image/Text
RecipeQA	250,730	36,786	Image/Text

Table 3: Comparison of the RecipeQA dataset to other multimodal machine comprehension datasets.

the steps by using WMD (Kusner et al., 2015) distance, where such distances are averaged. Then, the choice closest to all of the question steps is selected as the final answer. For the visual cloze task, a similar approach is carried out by considering images instead of text using deep visual features. For the visual coherence task, since the aim is to find the incoherent image among other images, the final answer is chosen as the most dissimilar one to the remaining images on average. Lastly, for the visual ordering task, first, the distances between each consecutive image pair in a candidate ordering of the jumbled image set is estimated. Then, each candidate ordering is scored based on the average of these pairwise distances and the choice with the minimum average distance is set as the final answer. In all these simple baseline models, we use the cosine distance to rank the candidates.

4.3 Baseline Results

We report the performance of the baseline models in Table 2 which indicates the ratio of correct answers against the total questions in the test. For the textual cloze, the comparison between text-only and multimodal Impatient Reader models shows that the additional visual modality helps the model to understand the question better and to provide more accurate answers. While for the cloze style questions, the Impatient Reader outperforms the Hasty student, for the visual coherence and visual ordering style questions Hasty student gives way better results. This demonstrates that better neural models are needed to be able to effectively deal with this kind of questions. Some qualitative examples are provided in the supplementary material.

5 Related Work

Question Answering has been studied extensively in the literature. With the success of deep learning approaches in question answering, comprehension and reasoning aspects of the task has attracted researchers to investigate QA as a medium to measure intelligence. Various datasets and methods

have been proposed for measuring different aspects of the comprehension and reasoning problem. Each dataset has its own merits as well as weaknesses. Recently, a thorough analysis by (Chen et al., 2016) revealed that the required reasoning and inference level was quite simple for CNN/Daily Mail dataset (Hermann et al., 2015). To make reasoning task more realistic, new datasets such as SQuAD (Rajpurkar et al., 2016), NewsQA (Trischler et al., 2017), MSMARCO (Nguyen et al., 2016), CLEVR (Johnson et al., 2017), COMICS (Iyyer et al., 2017) and FigureQA (Kahou et al., 2018) have been proposed.

In the following, we briefly discuss the publicly available datasets that are closely related to our problem and provide an overview in Table 3.

The closest works to ours are (Iyyer et al., 2017), (Tapaswi et al., 2016) and (Kembhavi et al., 2017) where data multi-modality is the key aspect. COMICS dataset (Iyyer et al., 2017) focus on comic book narratives and explore visual cloze style questions, introducing a dataset consisting of drawings from comic books. The dataset is constructed from 4K Golden Age (1938-1954) comic books from the Digital Comics Museum and contains 1.2M panels with 2.5M textboxes. Three tasks are evaluated in this context, namely text cloze, visual cloze, character coherence. MovieQA dataset (Tapaswi et al., 2016), comprises of 15K crowd-sourced questions about 408 movies. It consists of movie clips, subtitles, and snapshots, is about comprehending stories about movies. TQA dataset (Kembhavi et al., 2017), have 26K questions about 1K middle school science lessons with 3.5K im-

ages, mostly of diagrams and aims at addressing middle school knowledge acquisition using both images and text. Since the audience is middle school children, it requires limited reasoning.

RecipeQA substantially differentiates from the previous work in the following way. Our dataset consists of natural images that are taken by anonymous users in unconstrained environments, which is a major diversion from COMICS and TQA datasets.

It should also be noted that there has been a long history of research involving cooking recipes. Recent examples include parsing of recipes (Malmaud et al., 2014; Jermsurawong and Habash, 2015), aligning instructional text to videos (Malmaud et al., 2015; Sener et al., 2015), recipe text generation (Kiddon et al., 2016), learning cross-modal embeddings (Salvador et al., 2017), tracking entities and action transformations in recipes (Bosselut et al., 2018).

Finally, to our best knowledge, there is no dataset focusing on "how-to" instructions or recipes; hence, this work will be the first to serve multimodal comprehension of recipes having an arbitrary number of steps aligned with multiple images and multiple sentences.

6 Conclusion

We present RecipeQA, a dataset for multimodal comprehension of cooking recipes, which consists of roughly 20K cooking recipes with over 36K context-question-answer triplets. To our knowledge, RecipeQA is the first machine comprehension dataset that deals with understanding procedural knowledge in a multimodal setting. Each one of the four question styles in our dataset is specifically tailored to evaluate a particular skill and requires connecting the dots between different modalities. Results of our baseline models demonstrate that RecipeQA is a challenging dataset and we plan make it publicly available for other researchers to promote the development of new methods for multimodal machine comprehension. In the future, we also intend to extend the dataset by collecting natural language questions-answer pairs via crowdsourcing. We also hope that RecipeQA will serve other purposes for related research problems on cooking recipes as well.

References

Harsh Agrawal, Arjun Chandrasekaran, Dhruv Batra, Devi Parikh, and Mohit Bansal. 2016. Sort story: Sorting jumbled images and captions into stories. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 925–931.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. VQA: Visual question answering. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2425–2433.

Antoine Bosselut, Corin Ennis, Omer Levy, Ari Holtzman, Dieter Fox, and Yejin Choi. 2018. Simulating action dynamics with neural process networks. In *International Conference on Learning Representations (ICLR)*.

Léon Bottou. 2014. From machine learning to machine reasoning - an essay. *Machine Learning*, 94(2):133–149.

Christopher JC Burges. 2013. Towards the machine comprehension of text: An essay. Technical report, Technical report, Microsoft Research Technical Report MSR-TR-2013-125.

Danqi Chen, Jason Bolton, and Christopher D Manning. 2016. A thorough examination of the CNN/Daily Mail reading comprehension task. In *Association for Computational Linguistics (ACL)*.

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*, pages 2493—-2537.

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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1693–1701.

Daniel Hewlett, Alexandre Lacoste, Llion Jones, Illia Polosukhin, Andrew Fandrianto, Jay Han, Matthew Kelcey, and David Berthelot. 2016. WikiReading: A novel large-scale language understanding task over Wikipedia. In *Association for Computational Linguistics (ACL)*.

Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2016. The goldilocks principle: Reading children's books with explicit memory representations. In *International Conference on Learning Representations (ICLR)*.

Lynette Hirschman and Robert Gaizauskas. 2001. Natural language question answering: The view from here. *Natural Language Engineering*, 7(4):275–300.

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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jermsak Jermsurawong and Nizar Habash. 2015. Predicting the structure of cooking recipes. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 781–786.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. 2017. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Association for Computational Linguistics (ACL)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Samira Ebrahimi Kahou, Adam Atkinson, Vincent Michalski, Akos Kadar, Adam Trischler, and Yoshua Bengio. 2018. FigureQA: An annotated figure dataset for visual reasoning. In *International Conference on Learning Representations (ICLR) Workshops*.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. 2016. A diagram is worth a dozen images. In *European Conference on Computer Vision*, pages 235–251. Springer.

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 *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chloé Kiddon, Luke Zettlemoyer, and Yejin Choi. 2016. Globally coherent text generation with neural checklist models. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 329—-339.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*.

Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger. 2015. From word embeddings to document distances. In *International Conference on Machine Learning (ICML)*, pages 957–966.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International Conference on Machine Learning (ICML)*, pages 1188–1196.

Marco Lui and Timothy Baldwin. 2012. `langid.py`: An off-the-shelf language identification tool. In *Association for Computational Linguistics (ACL) Demo Session*, pages 25–30.

Jonathan Malmaud, Jonathan Huang, Vivek Rathod, Nick Johnston, Andrew Rabinovich, and Kevin Murphy. 2015. What's cookin'? Interpreting cooking videos using text, speech and vision. In *North American Association for Computational Linguistics (NAACL)*.

Jonathan Malmaud, Earl Wagner, Nancy Chang, and Kevin Murphy. 2014. Cooking with semantics. In *ACL 2014 Workshop on Semantic Parsing*, pages 33–38.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *NIPS 2016 Workshop on Cognitive Computation*.

Peter Norvig. 1987. A unified theory of inference for text understanding. Technical report, University of California at Berkeley, Berkeley, CA, USA.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet: Large scale visual recognition challenge. *International Journal of Computer Vision*, 113:211–252.

Amaia Salvador, Nicholas Hynes, Yusuf Aytar, Javier Marin, Ferda Ofli, Ingmar Weber, and Antonio Torralba. 2017. Learning cross-modal embeddings for cooking recipes and food images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ozan Sener, Amir Zamir, Silvio Savarese, and Ashutosh Saxena. 2015. Unsupervised semantic parsing of video collections. In *IEEE International Conference on Computer Vision (ICCV)*.

Saku Sugawara, Hikaru Yokono, and Akiko Aizawa. 2017. Prerequisite skills for reading comprehension: Multi-perspective analysis of mctest datasets and systems. In *Association for the Advancement of Artificial Intelligence (AAAI)*, pages 3089–3096.

Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2016. MovieQA: Understanding stories in movies through question-answering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4631–4640.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. Newsqa: A machine comprehension dataset. In *2nd Workshop on Representation Learning for NLP*.

Lucy Vanderwende. 2007. Answering and questioning for machine reading. In *AAAI Spring Symposium: Machine Reading*, page 91.

Licheng Yu, Eunbyung Park, Alexander C. Berg, and Tamara L Berg. 2015. Visual madlibs: Fill in the blank description generation and question answering. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2461–2469.