SemEval-2022 Task 9: R2VQ – Competence-based Multimodal Question Answering

Jingxuan Tu, Eben Holderness, Kyeongmin Rim, Kelley Lynch, Richard Brutti, James Pustejovsky
Lab for Linguistics & Computation
Department of Computer Science
Brandeis University
{jxtu,egh,krim,kmlynch,brutti,jamesp}@brandeis.edu

Marco Maru[1], **Simone Conia**[1], **Roberto Navigli**[2]
Sapienza NLP Group
[1]Department of Computer Science
[2]Department of Computer, Control and Management Engineering
Sapienza University of Rome
first.lastname@uniroma1.it

Abstract

In this task, we identify a challenge that is reflective of linguistic and cognitive competencies that humans have when speaking and reasoning. Particularly, given the intuition that textual and visual information mutually inform each other for semantic reasoning, we formulate a Competence-based Question Answering challenge, designed to involve rich semantic annotation and aligned text-video objects. The task is to answer questions from a collection of English language cooking recipes and videos, where each question belongs to a "question family" reflecting a specific reasoning competence. The data and task result is publicly available. [1]

1 Introduction

One of the fundamental goals of Artificial Intelligence (AI) has been to create systems that interact with human users fluently and intelligently, by demonstrating inferencing and reasoning capabilities that would be expected of a human partner. This includes a growing interest in posing larger challenges to end-to-end systems employing architectures with deep neural networks (DNNs) (Ribeiro et al., 2020; Prabhumoye et al., 2020; Rogers et al., 2021; Minaee et al., 2021). Here we argue that we should start focusing on linguistic *competencies*, and not just on Question Answering (QA) skills or "challenge checklisting". There are some moves in this direction already (Johnson et al., 2017), but there is still no generally accepted distinction in current Natural Language Processing (NLP) between challenge-based tasks and competence-based performance (Bentivogli et al., 2017). Analogous to human cognitive competencies, there is both a methodological and modeling advantage to focusing a system's performance on competence-based learning rather than a narrowly defined task or challenge checklist.

First we define competence-based knowledge, and then the questions that can be generated from such knowledge. While Chomsky (1965)'s distinction between competence and performance has long been debated in linguistics, the term *competence-based* has been applied to a number of different concepts in both the science of learning and educational communities (Bechtel et al., 1999; Voorhees, 2001; Chyung et al., 2006; Platanios et al., 2019; Hsiao et al., 2020). The common core to both is a concept capturing a coherent set of abilities that an individual has in a specific domain (Doignon and Falmagne, 1985; Heller et al., 2013).

Here we focus on *lexical competence* as deployed in both single and multiple sentence composition (Pustejovsky, 1995; Marconi, 1997; Geeraerts, 2009; Asher, 2011). A competence-based question will query competence-based knowledge structures. For this task, lexical competence will involve the following:

- Understanding implicit arguments that are not present (due to syntactic ellipsis or semantic defaulting or shadowing), and being able to use this (missing) information to formulate knowledge about the event or situation (Malmaud et al., 2014; Kiddon et al., 2015);
- Understanding the dynamics of the text or narrative and how events can change an object or contribute to new properties (and subsequent descriptions) of objects in the text (Tandon et al., 2018; Das et al., 2018; Brown et al., 2018).

It is clearly the case that these two phenomena require non-extractive QA capabilities of some sort. We describe our dataset, Recipe-to-Video Questions (R2VQ), and summarize the procedures implemented by task participants for answering such

[1] https://competitions.codalab.org/competitions/34056

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 1244 - 1255

questions in the remainder of the paper.

2 Overview

2.1 Summary of the task

The task is structured as QA pairs, querying how well a system understands the semantics of English language recipes.

We hope that this task will help move NLP system design and evaluation towards the construction of meaning representations involving linguistic and multimodal situated grounding. In the present context, this involves identifying cooking entities and activities from recipe text, as well as linking them to videos of related recipes, entities, and activities.

Participants are provided with a multimodal training set, and are asked to provide answers to unseen queries. These questions can be answered using a unimodal dataset of text recipes and associated annotations. Participants are also encouraged to explore the full multimodal training set with additional cooking videos to potentially improve the results from the unimodal models. Following SemEval guidelines, the R2VQ dataset is publicly available[2] in CONLL-U format, with annotations encoded in plain text files.

2.2 Impact of the task

When we apply our existing knowledge to new situations, we demonstrate a kind of understanding of how the knowledge (through tasks) is applied. When viewed over a conceptual domain, this constitutes what we will refer to as a *competence*, and the corresponding challenge can be called a competence-based challenge. Competence-based evaluations can be seen as a new approach for designing NLP challenges, in order to better characterize the underlying operational knowledge that a system has for a conceptual domain, rather than focusing on individual tasks.

3 Related Work

NLP challenges have helped drive progress in the field recently. These challenges in part have been framed as specific tasks, and advances are largely driven by leaderboards on benchmark datasets or model comparison on individual datasets. Common benchmarks such as GLUE (Wang et al., 2018) and SuperGLUE (Wang et al., 2019) have been used widely. They contain several language understanding tasks such as Winograd Natural Language Inference (WNLI) (Levesque et al., 2011) as an inference task, and Winograd Schema Challenge (WSC) (Levesque et al., 2011) as a coreference resolution task. A survey (Rogers et al., 2021) showed the recent trend to measure various machine reasoning capabilities using different designs of QA tasks.

While all the tasks aim to advance the research towards corresponding NLP challenges, whether these reflect human competencies remains a question, especially in recent years with the success of transformers (Devlin et al., 2019; Yang et al., 2019; Liu et al., 2019). Many top-ranked NLP models that have shown better performance than humans on benchmarks may have come from overfitting to the dataset rather than addressing the challenge (Rogers, 2019). Current pre-training paradigms may also tune models towards capturing merely statistical patterns, so datasets should be designed to align the model's ability with human expectations (Linzen, 2020). Sugawara et al. (2020) found that most of the questions from common QA and reading comprehension datasets can be correctly answered by models without complex reasoning.

Recent work has been trying to identify and evaluate the tasks that are reflective of human linguistic and reasoning competencies. For example, Kim and Linzen (2020) proposed a semantic parsing dataset that evaluates the human-like compositional generalization of models. Ribeiro et al. (2020) designed three test types that can be used to test various linguistic capabilities of NLP models. More closely related to our work, QA-SRL (He et al., 2015) use predicate-argument structure to represent QA pairs. SynQG (Dhole and Manning, 2021) and RoleQ (Pyatkin et al., 2021) try to incorporate existing semantic annotations to generate comprehension questions.

4 Task Description

We formulate the task as competence-based QA, designed to involve rich semantic annotation and aligned text-video objects. The goal of this task is to answer questions from a collection of cooking recipes and images. Each question belongs to a "question family" that characterizes a specific reasoning competence to be tested. These competencies include abilities such as spatial and temporal reasoning, semantic role assignment, and object

[2] https://competitions.codalab.org/competitions/34056#participate

Recipe Title: **Appelkoek** *Passage:* Peel and cut apples into eighths (wedges). Sift together flour, baking powder and salt with 4 tablespoons of the sugar. Cut in butter. Combine egg and milk and add to flour mixture. Turn batter into greased 8 inch square cake pan. Press apple wedges partly into batter. Combine remaining 2 tbsp sugar and cinnamon. Sprinkle over apple. Bake at 425 degF for 25 to 30 minutes.

IMPLICIT	How do you cut apples into wedges? - by using a knife
ELISION	What should be sprinkled over apple wedges? - cinnamon sugar
LOC. CHANGE	Where was the batter when you press apple wedges? - in the pan
OBJ. LIFESPAN	What's in the appelkoek? - apples
SRL-TIME	For how long should you bake appelkoek? - 20 to 35 minutes
SRL-VALUE	How do you bake appelkoek? - bake at 425 degF

Table 1: Example competence-based questions. Color-coded text spans represent how information has been collected and generated in the questions.

cardinality and counting.

We adopt the concept of "question families" as outlined in the CLEVR dataset (Johnson et al., 2017). While some question families (e.g., integer comparison, counting) naturally transfer over from the Visual Question Answering (VQA) domain (Antol et al., 2015; Zhu et al., 2016), other concepts such as ellipsis and object lifespan must be employed to cover the full extent of competency within procedural texts. On the basis of the aforementioned competencies, we categorize the questions into five question families. Table 1 shows the definition of each question family as well as sample questions.

The question families are defined as follows:

- *Cardinality*: covers concepts of integer comparison and counting.
- *Elision*: deals with identifying arguments (ingredients in most cases) that are omitted from a text, but can be understood from context.
- *Implicit*: covers both implicit tools and habitats introduced in the text. This is distinct from elision, as these are not solved merely through contextual clues. Instead, they require general competence; applying world knowledge of an action and its requirements to a novel situation.
- *Obj. Lifespan*: covers different states of an object in a cooking event.
- *Semantic Role Labeling (SRL)* covers semantic roles that are modifiers to a cooking event.

5 Data and Resources

The textual component of our dataset consists of a collection of English language recipes sourced from two open-source recipe wikis, Recipe Fandom[3] and Foodista[4], and is labeled according to three distinct annotation layers: (i) Cooking Role Labeling (CRL), (ii) Semantic Role Labeling (SRL), and (iii) aligned key frames image triples taken from creative commons cooking videos downloaded from YouTube.

Compared to text of news or narratives, procedural text such as recipes and user manuals tend to be task-oriented, and the main content is split into steps that describe small goals to accomplish the final task. We believe such texts are a good fit for our task, as it involves the understanding of how to reach the goal locally for each step, as well as how each step contributes to the final task globally. Further, the step-wise progression inherent in the goal-oriented narrative contributes both an interpretative dynamics as well as contextualized elision of arguments.

5.1 Train/Dev/Test Datasets

There are $1,000$ recipes released as part of the task (800 for training and 100 each for validation and testing). Table 2 shows the basic statistics of the dataset. We exclude any "less informative" recipe that has less than 4 sentences from our dataset. For each recipe, there are an average of 35 questions (5 from each question family). Each recipe is also paired with an additional set of 10 "unanswerable" questions (answers that cannot be found in a given recipe) as negative samples.

5.2 Cooking Role Labeling

Cooking Role Labeling (CRL) is a domain-specific dependency relation annotation for the cooking domain. CRL is done via a two-phase annotation. First, to identify mentions of cooking events and

[3] https://recipes.fandom.com/
[4] http://foodista.com/

	Train	Dev	Test
# of recipes	800	100	100
Avg. # of sentences per recipe	8	7.9	7.8
Max. # of sentences	26	16	31
Min. # of sentences	4	4	4
Avg. sentence length per recipe	12.5	13.4	12.5
Max. sentence length	32	25	19
Min. sentence length	6	6	7

Table 2: Statistics of the train, dev and test subsets of the R2VQ dataset.

Figure 1: Docanno environment for event and entity annotation.

entities and put labels on them, and then to establish relations between those mentions.

Each step in a given recipe is annotated for cooking-related *events* and the associated *entities* (ingredients and props such as tools, containers, and habitats). The ingredients can be either labeled as explicit (those listed in the ingredients section of the recipe) or implicit (intermediate outputs of applying a cooking action to a set of explicit ingredients).

We post-process the data by running the Stanza pipeline (Qi et al., 2020) on the raw text of each recipe to get tokenization and other basic linguistic features including word lemmas, part-of-speech tags. We took a semi-automated approach to performing the span-level entity annotations. First, using a small labeled dataset as seed training data, we trained a character-level named entity recognition (NER) model using Flair embeddings (Akbik et al., 2019) to pre-annotate the recipe text. We then validated the model predictions using the Docanno annotation tool (Nakayama et al., 2018) to create our gold set of event and entity mentions. Figure 1 shows an example from the Docanno environment, with annotations for Event, Implicit Ingredient, and Habitat.

For the next phase of annotation, we developed Deep Event & Entity Palette or DEEP, a specialized annotation environment to manually annotate cooking role relations. Annotators start from documents that are already annotated with span-level entity tagging from Docanno. The primary job of annotators is to draw links between entities (coreference) or between an entity and an event (participant). DEEP provides an intuitive and easy interface for pairwise linking annotation, as well as a holistic view of the document-level context using color coding of tokens related to the selected events or entities, as shown in Figure 2. All annotation is done at document-level, namely, annotators can create long distance links. For example, a food entity from a previous step can be linked to an event in the next step even if the direct object of the event is omitted on surface (or "hidden"). And finally, DEEP also provides an interface to add such hidden entities with a free-text identifier and immediately link it to an event.

More specifically, event-entity links can be one of several possible link tags, which can be made between explicit spans of text or between an event and a hidden entity that does not explicitly appear in the recipe text. These relations are:

- **Ingredient**: identifies the food material that participates in cooking events.
- **Result**: identifies entities produced as the output of an event.
- **Tool**: relates objects with the events they are used in. Tools may appear in the text ("Cut the pear with a sharp knife"), or they may be hidden ("Cut an apple" requires an unmentioned knife).
- **Habitat**: links events with the objects in which they take place. Habitats may appear in the text ("Bake in a preheated oven"), or they may be hidden ("Saute the onion" requires an unmentioned pan).

Table 3 shows the statistics of cooking role annotation on the dataset. EVENT should always be explicit, while the other cooking roles can be either explicit text spans or hidden entities. We hired 8 student annotators for the CRL annotation work. All annotators were students at Brandeis University, ranging from undergraduate to master's level.

5.3 Semantic Role Labeling

Aside from the above-described annotation layer, which is tailored to highlight domain-specific events and entities, each step in the recipes featured in R2VQ is automatically tagged and manually validated according to the predicates and constituents identified at the Semantic Role Labeling (SRL) level, i.e., the task of identifying and labeling predicate-argument structures within a sentence (Gildea and Jurafsky, 2002). More specifically,

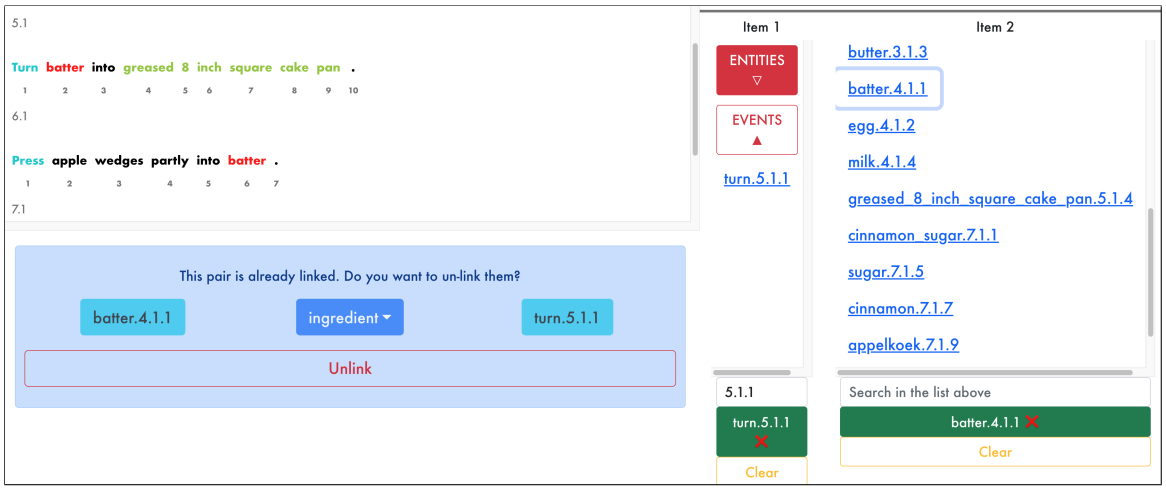

Figure 2: DEEP environment for CRL entity linking.

	Train		Dev		Test	
Avg. # of entities per recipe	Exp.	Hidden	Exp.	Hidden	Exp.	Hidden
EVENT	14.0	N/A	13.6	N/A	13.3	N/A
INGREDIENT	13.0	6.9	14.0	10.8	12.5	8.6
RESULT	0.2	1.5	0.2	1.4	0.3	1.7
TOOL	0.6	2.1	0.7	2.2	0.6	2.0
HABITAT	2.8	4.8	2.5	6.2	2.5	4.0

Table 3: Statistics of cooking role annotation on R2VQ.

each recipe step is semantically enriched by (i) identifying all its predicates, i.e., those words or multi-word expressions that denote an event or an action, (ii) assigning the most appropriate sense label to each identified predicate according to a predefined inventory, (iii) detecting all the arguments, i.e., the parts of the text that are semantically related to each predicate, and (iv) choosing the most fitting semantic role for each predicate-argument pair. Let's consider the example "John bakes potatoes". In this case, SRL consists of (i) identifying "bake" as a predicate, that is, something that denotes an action or an event; (ii) disambiguating the predicate, that is, assigning the most appropriate sense for "bakes" in this context; (iii) identifying the arguments of each predicate, that is, those parts of the text, "John" and "potatoes" that are semantically linked to "bakes"; and (iv) assigning a semantic role to each predicate-argument pair, e.g., "John" is the *Agent* of the predicate "bakes", whereas "potatoes" is the *Patient*.

In SRL, there are two main annotation formalisms for tagging arguments: span-based and dependency-based. We adopted the former; the core and only difference between the two lies in the fact that, in the span-based SRL, semantic role labels are applied to the whole span of a given argument, whereas, in dependency-based SRL, the label is only applied to the argument's head (e.g., we label "the broccoli" and not "the").

The SRL task is often tied to a linguistic resource, which defines the inventory of predicate senses and semantic roles. For this task, we chose VerbAtlas[5] (Di Fabio et al., 2019) as our inventory of predicate senses and semantic roles given its high coverage in terms of verbal lexicon[6], the informativeness of its human-readable roles (e.g., *Agent*, *Patient*, *Instrument*), and its mapping to the PropBank frame inventory (Palmer et al., 2005) and to the BabelNet multilingual knowledge base (Navigli and Ponzetto, 2012; Navigli et al., 2021).

The annotation process for the SRL layer featured three distinct stages. In detail, we first employed the Stanza toolkit (Qi et al., 2020) to perform PoS tagging over the R2VQ corpus so as to

[5]VerbAtlas is freely available for research purposes at http://verbatlas.org/.

[6]VerbAtlas covers all the verbal senses defined in WordNet, and clusters them into predicate frames.

identify verbal predicates, and proceeded to manually include predicates that were not discovered automatically (e.g., *season* was often incorrectly labeled as a noun), as well as fixing instances erroneously labeled as predicates (such as adjectival or prenominal predicates, as well as predicates appearing within ill-formed sentences).[7] Secondly, we employed a state-of-the-art system (Conia and Navigli, 2020) to automatically label recipes in a span-based fashion, concurrently assigning VerbAtlas frames and arguments to recipes, and manually validating the whole corpus once more in order to verify the automatically-generated outputs, fixing errors and inconsistencies.[8]

We used BabelNet 5.0 as the inventory to validate predicates, first picking the most suitable word sense to disambiguate a given verb, and then selecting the relative frame in VerbAtlas according to its original mapping. As our final step, we instructed annotators to manually tag as many arguments as possible for each predicate (adding arguments where needed and removing additional arguments such as *Negation* in the process), first, referring to the predicates' prototypical arguments according to VerbAtlas, and then, providing additional arguments. We used VerbNet (Schuler, 2006) argument descriptions and examples along with in-house argument descriptions for ambiguous argument assignments (e.g., "in the oven" in "Jennifer baked the potatoes IN THE OVEN" is not a *Agent*, but rather an *Instrument* with respect to to the predicate "bake").

With respect to the SRL layer annotators, in order to make use of the Mechanical Turk platform already employed in the context of the aligned image frame annotation, we initially devised HITs for both predicate sense disambiguation and argument labeling. Though, independently of the rates and templates employed, we kept collecting low-quality or suboptimal data, likely, due to the background knowledge needed to perform such tasks in an adequate fashion. In light of this, after several attempts, we eventually decided to have one in-house annotator with extensive experience in SRL validate the whole corpus at all stages required, and asked a second annotator to review the validation instances,

[7]We also labeled word forms with typos in the original recipes as predicates (e.g. *prehet* as *preheat*). Additionally, we labeled as multi-word predicates those predicates whose form was featured as a compound in BabelNet.

[8]See Appendix A for details about the SRL annotations' format.

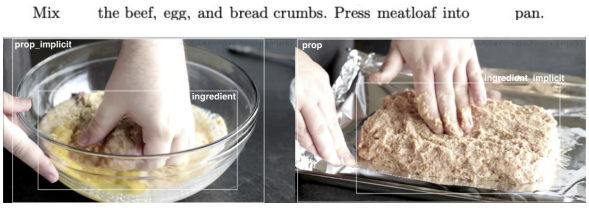

Figure 3: An aligned CRL-Image Frame annotation.

seeking agreement in case of discrepancies. As an additional step to ensure data quality, a third, external annotator was assigned with the task of reviewing recipes in order to look for potential formatting issues.[9]

5.4 Aligned Image Frame Annotation

Accompanying each recipe is a series of images extracted from YouTube videos that are associated with a particular event in the recipe. We pulled the images from a set of YouTube videos that were selected by querying YouTube for recipe titles. For each recipe title, we downloaded 5 Creative Commons licensed videos. These videos were indexed by generating an embedding using the Tensorflow implementation of the S3D Text-Video model trained on HowTo100M using MIL-NCE (Miech et al., 2020, 2019). For each cooking event in the recipes, the 5 closest clips as scored by L2 distance were selected from the YouTube videos we downloaded. We showed the annotators the first, middle and last frame from each 4 second clip alongside a list of the CRL representations of the events in the recipe. We asked the annotators to rank the match of the image and the cooking event as a good match, a partial match, or not a match. The Swipe Labeler (Peterson Jenessa, 2021) tool was used to conduct the annotation. The tool was modified to included the recipe event text, with the full recipe displayed and the current step in bold text. An example of the frame annotation is shown in Figure 4.

Due to complex combinations of ingredients in many of the recipes and the limitation of considering only Creative Commons videos, many events did not match with any of the detected segments. Partial matches were included in order to increase the total number of events represented. Importantly, the action represented in the image clips does nec-

[9]All annotators employed in the SRL layer have effective operational proficiency in English and received a wage in line with their country of residence. Annotation has been carried out by means of user-friendly shared worksheets.

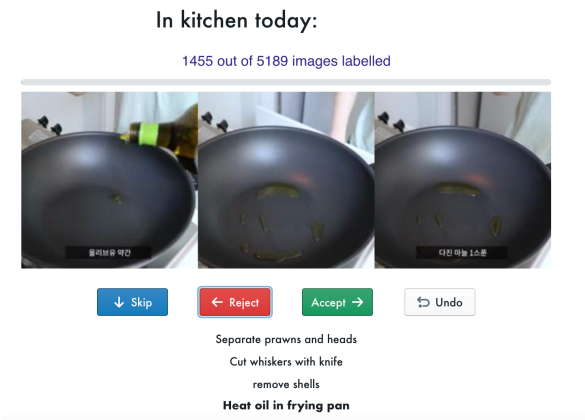

Figure 4: Swipe Labeler Annotation Tool

essarily include the exact same ingredients of as those used in the recipe. The videos were chosen based on the similarity of the cooking event described in the sentence. In total, 1927 events were matched with images across 655 recipes.

5.5 Generating Competence-based Questions

We first design text templates for each type of question. Then we generate QA pairs by populating the templates in a cloze test style with the data annotated in CRL. Table 4 shows the text templates for two types of questions we want to use for the QA task. ELISION identifies arguments (ingredients in most cases) that are omitted from a text, but can be understood from context. IMPLICIT covers both implicit tools and habitats introduced in the text. This is distinct from ELISION, as these are not solved merely through contextual clues. Each text template has several slots that can be filled with corresponding entities from CRL.

To increase the variety of questions, we also include adjunct slots into the templates. As shown in Table 4, adjunct slots include tool or habitat phrases and SRL modifiers. SRL modifiers are any semantic roles that are not claimed by CRL entities such as `TIME` and `VALUE`. For example, one ELISION question can be as short as *What should be cut?* or *What should be cut on the board with a knife into eighths?* with all the adjunct slots. We argue that it is helpful to generate questions more challenging to the systems. Adding more adjunct slots completes the context for the question, but also introduces unseen context if the slots contain hidden entities.

These slotted templates are further processed to improve the readability of generated questions. We change word inflections and insert articles and agreements. For the templates with [habitat_phrase] and [tool_phrase] slots, we fill those with corresponding `LOCATION` or `INSTRUMENT` spans from SRL. If a slot is filled with a hidden entity that has no associated semantic roles, we run a BERT-based model (Devlin et al., 2019) to get the most likely preposition given the sentence as context through the masked language modeling task. SRL modifiers are populated in the same order as they were in the original sentence.

5.6 Details of copyright

All recipes are distributed under Creative Commons license. The YouTube videos queried were limited to Creative Commons videos only. No personally identifying information is included in either the text or visual components of the dataset.

6 Participation

We discuss the baseline system and the systems from participants in this section.

6.1 Evaluation Metrics

All systems are asked to provide answers to the open-ended questions based on the textual and visual information encoded in the dataset. The results are evaluated using exact match (EM) and token-level F1 score (F1) following Rajpurkar et al. (2018).

6.2 Baselines

To build a model that is reflective of the nature of the abstractive question answering task and benefits from the aligned key frames to the text, we adopt a vision-and-language text genertation model as the baseline for our task. We build the baseline with the model framework that is proposed by Cho et al. (2021). They propose the model VL-T5 based on T5 text generation model (Raffel et al., 2020) by extending the original T5 text encoder to a multimodal encoder that can take both textual and visual embeddings as the input.

Following closely the VL-T5 work (Cho et al., 2021), we prepare the key frames as model input by encoding them into visual embeddings using Faster-R-CNN. We prepare the text input by appending the task-specific prefix to the question and context text: `"question: {question_str} context: {recipe_str}"`. The `recipe_str` is the concatenation of the text of all cooking steps from the recipe the question is generated from. We fine-tune the VL-T5 model for our QA task on the

QUESTION TYPE	TEXT TEMPLATE	QUESTION-ANSWER PAIR
Elision	What should be verb [habitat_phrase] [tool_phrase] [modifiers]? — ingredient_obj	What should be cut on the board with a knife into eighths? — apples
Implicit	What do you use to verb obj [habitat_phrase] [modifiers]? — tool Where do you verb obj [tool_phrase] [modifiers]? — habitat_phrase	What do you use to sauté the onions [in the pan]? — spatula Where do you arrange the slices [into rounds]? — in the casserole

Table 4: Text templates and example of generated questions. The squared brackets ([...]) in the templates indicates adjunct slots.

training set, and run the fine-tuned model on the test set. As a comparison, we also fine-tuned the T5 model with text input only. Baseline results are shown in Table 5 along with other results from participants.

	EM	F1	Key Frames?
SRPOL	92.53	94.34	
ITNLP&QMUL	91.33	94.23	
PINGAN_AI	78.21	82.62	
Slug	69.49	77.37	
BASELINE (VL-T5)	69.37	77.77	✔
BASELINE (T5)	65.34	75.22	
ych	10.23	10.23	
UoR	5.90	15.78	✔
CLT6	0.0	0.0	

Table 5: Task results from participant teams and the baseline. The ranking is based on EM score. The last column indicates whether the system uses key frames for training.

6.3 Description of team submissions

We collect successful submissions from 8 participating teams (including the baseline), as well as one participating team that did not submit predictions that passed our automated evaluation script. The results and final ranking are shown in Table 5. We summarize their work below:

- **SRPOL**: This system attains the highest scores in this task by adopting a hybrid approach. The system includes a rule-based system for intent identification and finding N/A questions. It also applies a transformer-based model ELECTRA for generating extractive answers.
- **ITNLP&QMUL**: This system attains the second highest scores in this task. The system adapts a T5 model to the task by altering the input to include semantic and cooking role labels that are provided in the data.
- **PINGAN_AI**: This system attains the third highest scores in this task. The system uses the BERT model as the backbone, and enhances the model by incorporating additional knowledge about cooking entities and part-of-speech tags in the format of plain text and embeddings.
- **Slug**: Semantic labels were preprocessed using BERT and handmade rules, with hidden roles infused into the recipe. A task-finetuned T5 model was then used for question answering.
- **UoR**: The only submission that exploited the visual information provided in the dataset, this system used an Inception V3 model (pretrained on ImageNet), to extract image features that were used to train an image captioning model on the MS-COCO dataset. These captions were included alongside the recipe text in a Retrieval-Augmented Generation model for question answering.

7 Discussion

In this paper we have described the new task of Competence-based Multimodal Question Answering. In this task, we extended the traditional question answering by providing text-visual aligned data as the context, and asking questions that reflects reasoning competences over the question context. To create the dataset for our task, we proposed and applied a rich annotation of semantic role labels, cooking role labels and aligned video key frames to a set of cooking recipes.

A criticism of the approach we adopted to create annotated dataset is that the video key frames are not well aligned with the text, thus making it difficult to include those into the modeling training. Although with the full awareness of this, video annotation and alignment remains a very difficult task. Copyright issues also make it challenging for us to get enough video sources to work with. Future work to improve the key frame annotation may include utilizing entity recognition so that more ac-

curate alignment to text can be made. We will also consider reusing the key frames and adding static images to represent similar events from different recipes to increase the coverage of annotation. Another criticism of the data is the semantic ambiguity and loose definition of certain questions. For example, the same How-to question can have multiple reasonable answers, but only one is considered as the gold answer. Although this is the semantic ambiguity as it is, we intend to improve it by replacing the question phrase "How to ..." to more specific phrase like "What tool ..." based on the answer it is inquiring about.

An analysis of the systems that participated in our task showed the major improvement over the evaluation scores is achieved by making the hidden information appear on the surface. In general, two approaches are proven to be useful for this purpose by the participating systems. One is to train an end-to-end system to generate text that contains CRL-SRL annotation, so that the hidden information is expressed explicitly in the generated text. Then an extractive QA system can be adopted to identify text spans as answers. The second approach involves rules and heuristics to identify question intents, and get auxiliary knowledge. Intent identification can help classify questions into different categories. Each question category is associated with a rather fixed set of answer templates and possible entity types to be filled in. Auxiliary knowledge is generated by associating specific entities with their co-referred mentions or result ingredients (e.g. "small balls" to "flour mixture").

The analysis of the results from participating systems also reveals some interesting characteristics about the dataset and is useful for future task design. Despite the error rate of the top-performing systems such as SRPOL and ITNLP&QMUL is only 8%, the cardinality questions and How-to questions solely contribute the majority of the errors. As it is mentioned above, the innate ambiguity of How-to questions makes it difficult for both humans and systems to get a single correct answer. The poor performance on cardinality questions shows that the "counting reasoning" remains a big challenge to current transformer-based systems. In the R2VQ dataset specifically, the mentions of the entity involved a cardinality question can scatter over the whole recipe, which requires a larger context to answer such questions. Due to nature of "constant ingredient transformation" in cooking recipes, the mentions of the same entity could vary in our definition. For example, in the appelkoek recipe (Table 1), *apples*, *peeled apples*, *apple wedges*, *apples with batter* all refer to the same entity *Apple*. This characteristic of cardinality questions also hinders the systems from counting the mentions of the entity properly.

The human benchmark created by the SRPOL team provides useful insights on our future QA task design. They asked six linguists to answer $2,000$ questions selected randomly from the validation set. By examining the manual annotation on the questions, they found that although 73% of the annotated QA pairs have the same meaning as the gold answers, the EM score is quite low. This reveals the fact that traditional QA metrics that focus on string match might be too strict in our task. For example, from the analysis of the human benchmark, for the question *What's in the mixture?*, both the gold answer *the egg and mixture* and the human answer *the butter, sugar, tangerine zest, vanilla, baking powder, salt and egg* can be considered correct. Other metrics like BERTScore (Zhang et al., 2019) might be a good compliment to account for the syntactic and semantic variance between the model inference and the gold answer.

8 Conclusion

In this paper we described *SemEval-2022 Task 9: R2VQ – Competence-based Multimodal Question Answering*. The task is to answer questions from a collection of cooking recipes and videos, where each question belongs to a "question family" reflecting a specific reasoning competence. We developed a new dataset of cooking recipes with rich annotation for cooking roles, semantic roles and aligned video key frames. We collected 8 result submissions and analyzed the participating systems by highlighting and summarizing their findings to help future research pertaining the topic of our task.

References

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. Flair: An easy-to-use framework for state-of-the-art nlp. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and

Devi Parikh. 2015. VQA: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433.

Nicholas Asher. 2011. *Lexical meaning in context: A web of words*. Cambridge University Press.

Gregory A Bechtel, Ruth Davidhizar, and Martha J Bradshaw. 1999. Problem-based learning in a competency-based world. *Nurse Education Today*, 19(3):182–187.

Luisa Bentivogli, Ido Dagan, and Bernardo Magnini. 2017. The recognizing textual entailment challenges: Datasets and methodologies. In Nancy Ide and James Pustejovsky, editors, *Handbook of Linguistic Annotation*, pages 1119–1147. Springer.

Susan Windisch Brown, James Pustejovsky, Annie Zaenen, and Martha Palmer. 2018. Integrating generative lexicon event structures into verbnet. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Jaemin Cho, Jie Lei, Haochen Tan, and Mohit Bansal. 2021. Unifying vision-and-language tasks via text generation. *ArXiv*, abs/2102.02779.

Noam Chomsky. 1965. *Aspects of the Theory of Syntax*, volume 11. MIT Press.

Seung Youn Chyung, Donald Stepich, and David Cox. 2006. Building a competency-based curriculum architecture to educate 21st-century business practitioners. *Journal of Education for Business*, 81(6):307–314.

Simone Conia and Roberto Navigli. 2020. Bridging the gap in multilingual semantic role labeling: a language-agnostic approach. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1396–1410, Barcelona, Spain (Online).

Rajarshi Das, Tsendsuren Munkhdalai, Xingdi Yuan, Adam Trischler, and Andrew McCallum. 2018. Building dynamic knowledge graphs from text using machine reading comprehension. In *International Conference on Learning Representations*.

J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Kaustubh D. Dhole and Christopher D. Manning. 2021. Syn-qg: Syntactic and shallow semantic rules for question generation.

Andrea Di Fabio, Simone Conia, and Roberto Navigli. 2019. VerbAtlas: a novel large-scale verbal semantic resource and its application to semantic role labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 627–637, Hong Kong, China.

Jean-Paul Doignon and Jean-Claude Falmagne. 1985. Spaces for the assessment of knowledge. *International journal of man-machine studies*, 23(2):175–196.

Dirk Geeraerts. 2009. *Theories of lexical semantics*. OUP Oxford.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational linguistics*, 28(3):245–288.

Luheng He, Mike Lewis, and Luke Zettlemoyer. 2015. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 643–653, Lisbon, Portugal. Association for Computational Linguistics.

Jürgen Heller, Thomas Augustin, Cord Hockemeyer, Luca Stefanutti, and Dietrich Albert. 2013. Recent developments in competence-based knowledge space theory. In *Knowledge spaces*, pages 243–286. Springer.

Cheng-Ting Hsiao, Fremen ChihChen Chou, Chih-Cheng Hsieh, Li Chun Chang, and Chih-Ming Hsu. 2020. Developing a competency-based learning and assessment system for residency training: analysis study of user requirements and acceptance. *Journal of medical Internet research*, 22(4):e15655.

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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2901–2910.

Chloé Kiddon, Ganesa Thandavam Ponnuraj, Luke Zettlemoyer, and Yejin Choi. 2015. Mise en place: Unsupervised interpretation of instructional recipes. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 982–992.

Najoung Kim and Tal Linzen. 2020. Cogs: A compositional generalization challenge based on semantic interpretation. In *EMNLP*.

H. Levesque, E. Davis, and L. Morgenstern. 2011. The winograd schema challenge. In *KR*.

Tal Linzen. 2020. How can we accelerate progress towards human-like linguistic generalization? In *ACL*.

Y. Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

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

Diego Marconi. 1997. *Lexical competence*. MIT press.

Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. 2020. End-to-End Learning of Visual Representations from Uncurated Instructional Videos. In *CVPR*.

Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. 2019. HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips. In *ICCV*.

Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, and Jianfeng Gao. 2021. Deep learning based text classification: A comprehensive review.

Hiroki Nakayama, Takahiro Kubo, Junya Kamura, Yasufumi Taniguchi, and Xu Liang. 2018. doccano: Text annotation tool for human. Software available from https://github.com/doccano/doccano.

Roberto Navigli, Michele Bevilacqua, Simone Conia, Dario Montagnini, and Francesco Cecconi. 2021. Ten years of BabelNet: A survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4559–4567. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial intelligence*, 193:217–250.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, 31(1):71–106.

Weerasinghege Udara Peterson Jenessa, Ramesh Sumanth. 2021. Swipe-labeler.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom M Mitchell. 2019. Competence-based curriculum learning for neural machine translation. *arXiv preprint arXiv:1903.09848*.

Shrimai Prabhumoye, Ruslan Salakhutdinov, and Alan W Black. 2020. Topological sort for sentence ordering. *arXiv preprint arXiv:2005.00432*.

James Pustejovsky. 1995. *The Generative Lexicon*. MIT Press, Cambridge, MA.

Valentina Pyatkin, Paul Roit, Julian Michael, Reut Tsarfaty, Yoav Goldberg, and Ido Dagan. 2021. Asking it all: Generating contextualized questions for any semantic role.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *ArXiv*, abs/1806.03822.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Anna Rogers. 2019. How the transformers broke nlp leaderboards.

Anna Rogers, Matt Gardner, and Isabelle Augenstein. 2021. Qa dataset explosion: A taxonomy of nlp resources for question answering and reading comprehension.

Karin Kipper Schuler. 2006. *VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon*. Ph.D. thesis, University of Pennsylvania.

Saku Sugawara, Pontus Stenetorp, Kentaro Inui, and A. Aizawa. 2020. Assessing the benchmarking capacity of machine reading comprehension datasets. *ArXiv*, abs/1911.09241.

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 2018 Conference on Empirical Methods in Natural Language Processing*, pages 57–66.

Richard A Voorhees. 2001. Competency-based learning models: A necessary future. *New directions for institutional research*, 2001(110):5–13.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *NeurIPS*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *BlackboxNLP@EMNLP*.

Z. Yang, Zihang Dai, Yiming Yang, J. Carbonell, R. Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. 2016. Visual7W: Grounded question answering in images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4995–5004.

A Reading SRL Annotations in R2VQ

Predicate frames. Each predicate is labeled according to its VerbAtlas sense/frame. A value of "_" means that the corresponding word is not a predicate.

In the example below, there is only one predicate, "Cut" with the corresponding sense/frame "CUT" in position 1.

1	Cut	[...]	CUT	B-V
2	the	[...]	_	B-Patient
3	broccoli	[...]	_	I-Patient
4	into	[...]	_	B-Result
5	flowerets	[...]	_	I-Result
6	.	[...]	_	_

Semantic roles. For each predicate, we provide its semantic roles in BIO format (B - Beginning, I - Inside, O - Outside). Note that, for this dataset, we only use B and I to indicate the first token of a span and the rest of the tokens in the same span, respectively. In the example above, "the broccoli" is a *Patient* of the predicate CUT, with the token "the" as the Beginning of the span (B-Patient) and the token "broccoli" as the Inside of the span (I-Patient). Note that the predicate that refers to a specific column of semantic roles is always labeled with the notation B-V. Should the predicate consist of a multi-word expression, the other tokens apart from the first are labeled as I-V:

Should the multi-word expression be made of non-adjacent words, tokens apart from the first are instead labeled as D-V:

In the case of multiple predicates in the same sentence, there will be multiple semantic role columns, one for each predicate. For example, if there are two predicates in the sentence, one column will indicate the semantic roles for the first predicate,

1	Deep	[...]	COOK	B-V
2	-	[...]	_	I-V
3	fry	[...]	_	I-V
4	till	[...]	_	B-Result
5	crispy	[...]	_	I-Result
6	&	[...]	_	I-Result
7	golden	[...]	_	I-Result
8	brown	[...]	_	I-Result

1	Bring	[...]	CHANGE_APP./STATE	B-V
2	the	[...]	_	B-Patient
3	water	[...]	_	I-Patient
4	to	[...]	_	D-V
5	boil	[...]	_	D-V
6	.	[...]	_	_

and the following will show the semantic roles for the second predicate.

1	Reduce	[...]	REDUCE_D.	B-V	_
2	heat	[...]	_	B-Attr.	_
3	,	[...]	_	_	_
4	and	[...]	_	_	_
5	simmer	[...]	COOK	_	B-V
6	for	[...]	_	_	B-Time
7	1	[...]	_	_	I-Time
8	hour	[...]	_	_	I-Time
9	.	[...]	_	_	_