HIT&QMUL at SemEval-2022 Task 9: Label-Enclosed Generative Question Answering (LEG-QA)

Weihe Zhai[*1,2], Mingqiang Feng[*1], Arkaitz Zubiaga[2], Bingquan Liu[†1]

[1]Harbin Institute of Technology, China
[2]Queen Mary University of London, UK
{weihezhai, mqfeng}@insun.hit.edu.cn
a.zubiaga@qmul.ac.uk
liubq@hit.edu.cn

Abstract

This paper presents the second place system for the R2VQ: competence-based multimodal question answering shared task. The task consisted in building question answering systems that could process procedural recipes involving both text and image, and enriched with semantic and cooking roles. We tackled the task by using a text-to-text generative model based on the transformer architecture, with the aim of generalising across different question types. Our proposed architecture incorporates a novel approach for enriching input texts by incorporating semantic and cooking role labels through what we call Label-Enclosed Generative Question Answering (LEG-QA). Our model achieves a score of 91.3, with a significant improvement over the baseline (65.34) and close to the top-ranked system ((92.5). After describing the submitted system, we analyse the impact of the different components of LEG-QA as well as perform an error analysis.

1 Introduction

The objective of text-and-image multimodal question answering (QA) is to jointly leverage both textual and visual information to mutually inform each other for semantic reasoning (Ben-Younes et al., 2017). A SemEval 2022 shared task titled Competence-based Multimodal Question Answering (R2VQ) focuses on this task. In the R2VQ task, participants were invited to develop QA models to resolve questions associated with procedural recipe instructions. The corpus provided for the task is made of recipes, which include rich semantic annotations and where textual instructions are aligned with images illustrating them. The authors proposed to address a set of 18 question families, for which participants could develop and evaluate their own proposed solutions.

This paper describes the participation of the HIT&QMUL team in the R2VQ task, for which we proposed a methodology that we call label-enclosed generative question answering (**LEG-QA**). Through this methodology, we proposed enclosing labels providing semantic information embedded in the input texts. This methodology has proven competitive by achieving a score of 91.3% in exact match accuracy, ranking 2nd overall in the competition.

Our code is available at https://github.com/weihezhai/HIT-QMUL-at-SemEval-2022-Task-9.

2 Task and System Description

Building on the transformer architecture (Vaswani et al., 2017), we use T5 an encoder-decoder model (Raffel et al., 2020) implemented using Hugging Face[1]. We chose T5 given its reasonably good general language learning abilities, and provided that the downstream task of R2VQ covers a diverse set of task types that are also shared by T5. Intuitively, a task-agnostic model like T5 would be expected to perform well on R2VQ. We further adapt the T5 model to the task by altering the input to include semantic and cooking role labels enclosed in the textual recipe instructions.

2.1 Task and Data

The R2VQ (Tu et al.) task proposed the use of multimodal models to leverage both text and image for QA in the context of recipes. The R2VQ task adopts the definition of 'Question Family' from the CLEVR dataset (Johnson et al., 2017), where each type of question-answer pair comes from a template identified by task organisers. Each of these question type is meant to evaluate a different ability for reasoning.

The R2VQ dataset consists of a collection of 1,000 recipes, of which 800 are used for training

*contribute equally
†corresponding author

[1]https://huggingface.co/

and two sets of 100 recipes are used for validation and testing. These recipes involve more than 30K question-answer pairs, where each recipe consists of texts with procedural instructions as well as associated images. Questions are intended to require both visual and textual information jointly to produce effective answers. However, based on our initial explorations of the dataset provided and after considering the use of multimodal models, we found that the majority of the questions could be answered through the sole use of text. Hence, we thought that the use of an image processing component could be avoided while producing an accurate answer generation model for the dataset at hand. Further improvement of the model through the use of an image processing component is therefore left for future work.

For in-depth analysis of the system results, we grouped the 11 question types[2] provided by the organisers into 4 categories, i.e. generative, number reasoning, classification and extractive (see Table 5 in the Appendix for category details). Note that each of these classes comprises questions intended to evaluate different abilities of models.

2.2 Soft Constrained Generative QA as Multitask Transfer Learning

Soft Constrained Text-to-Text Generation Mainstream text-to-text generation methods mostly aim to learn meaningful mappings between input and output sequences. This is particularly the case for the recent pre-trained language models (Lewis et al., 2020; Raffel et al., 2020), where the model is expected to identify what to attend to in the source input and what to include in the model output. Models like UniLM, T5 and GPT2 unify the generation and understanding tasks within a single model, but none of them investigates the model's ability of generating free-form answers which includes both generative and discriminative tasks.

In R2VQ, all questions are created through a semi-automated method. Generative questions such as implicit argument identification (e.g. how do you drain the pasta?) cannot be answered independently by an extractive question answering approach, hence some question types should be treated as a soft constrained generation (SCGen) problem (See et al., 2017; Dou et al., 2021). SCGen implicitly specifies token constraints that the model needs to focus on in the answer output. For instance, in the question example above, one of the following words must be present in the answer: "by", "using" (gerund of a verb), "in/on/at", "with". There are many variants of models considering soft constraints, but usually they are achieved by adding an attention mechanism to the source keywords (Yao et al., 2019).

Multitask Transfer Learning Multitask learning consists in training the model on multiple tasks at a time. This means the model has an objective of simultaneously taking on more than one task. In LEG-QA, we leverage and transfer the prior knowledge from T5 and apply to our R2VQ tasks to solve the unseen SCGen problems. The reason why we do not train separate models for each of 4 categories in Table 5 is that we observe non-negligible accuracy drop when missing some of the question families.

2.3 Label-Enclosed Input

The key modification we made on the T5 model to prepare our submitted system is a rarely adopted heuristic method for embedding label information. Instead of appending supportive external features to the text sequence (He et al., 2017, 2018), we fuse the hidden cooking entities into the original text through our proposed approach called **label-enclosed input**. Typically, to grammatically and syntactically maintain the textual structure, the text should not be broken down into pieces by inserting annotations. However, we noticed that with the knowledge of labels appearing in the enclosed form, a pretrained text-to-text multitask model (in our case T5) can effectively process the enclosed noise from external information. In turn, the model using this enriched input behaves better than using a clean text input, when generatively answering questions with soft constraints.

2.4 Input Format

Figure 1 depicts the pipeline for the input data preprocessing through which the attributes of cooking roles are transformed and enclosed into the input sequences. We employed different processing approaches for each type of semantic and cooking roles. Through close observation, the most frequent attributes that take place in the answers are the 'Hidden' labels which consist of multiple values and keywords. As shown in the example, after text regularisation and reorganisation, attributes

[2]To be clear, 11 types of questions are defined on the R2VQ homepage but not the types of question-id in the dataset.

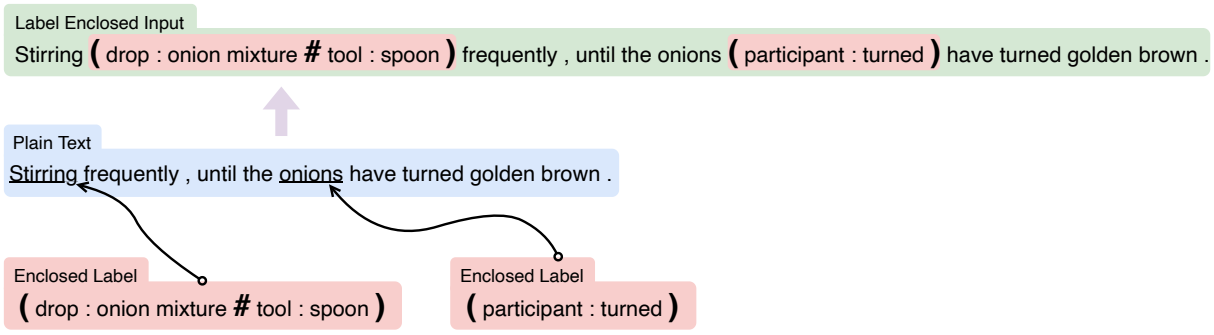

Figure 1: Pipeline for creating label-enclosed inputs. In this case, labels are wrapped in braces, and labels for the same event but are of different categories are separated by hashtag.

(e.g. stirring) are enclosed in special tokens before embedding them into the input. While the example shows the use of brackets for enclosing labels, we tested different approaches, which we discuss in more detail in Section 3.

3 Experiments

Since different formats of enclosing special tokens can result in considerably different scores, we compare a range of experiment settings to evaluate their performance. For more detail on the different enclosing methods tested, see Table 4 in Appendix A.

We conduct a series of experiments to answer the following questions:

Q1: Which way of enclosing special tokens performs best? Is there any big difference between them, and any measurable explanation for the gap?

Q2: Which label combos achieve the best accuracy?

Q3: How significant is the difference between results generated by size-equivalent models?

3.1 Experimental Settings

For the purpose of, as much as possible, controlling variables unrelated to the pre-trained language model, we experiment with five input variants on T5-large (our suboptimal result model). For comparison, we also show results for models based on different architectures. In this case, we use BART(Lewis et al., 2020), a denoising text-to-text model.

All the performance scores we show indicate the exact match (EM) accuracy on the development data.

	Enclosing Method	GEN	CLS	NUM	EXT	Overall
T5-Large	brackets	87.3	97.0	**82.6**	93.4	89.9
	hash	86.9	**97.4**	80.6	93.6	89.5
	dollar sign	**88.2**	96.9	81.0	93.8	**90.2**
	[BOL] [EOL]	87.0	97.2	81.0	93.4	89.5
	parallel	81.2	97.0	79.7	**94.1**	86.5

Table 1: Detailed results for different enclosing methods. The default method is "( )" which is used for our final submission. The dollar sign enclosing method is evaluated after the final submission so is not reported as the best practice during the competition.

3.2 Comparative study

To answer Q1, we report EM score for every 2K steps up to a total 26K steps. This is equivalent to approximately 10 epochs. Table 1 summarises scores achieved by the T5-Large model with the 5 different label-enclosing methods. The *Hidden, Part and Event* labels are examined together, given for that our best-performing model so far follows this paradigm. The special token "$" dollar sign has a noticeable positive impact on the GEN task, leading to the overall best performance. The "Parallel" enclosing method refers to directly attaching all labels horizontally aligned with plain text. Additionally, the [BOL] and [EOL] are special symbols inherent to BERT-like models. Unsurprisingly, joining labels in parallel with text without breaking sentence syntax helps achieve better EXT score. The reason for ultimately choosing "( )" is that we believe that its directional attribute could help the model parsing the label structure to some extent. A further combination comparison is discussed when answering Q2.

To answer Q2, we carry out label combination experiments to compare the effectiveness of different roles when contributing to the 4 question categories. Table 2 analyses the benefit of gradually

	Labels Combo	GEN	CLS	NUM	EXT	Overall
T5-Large	HIDDEN	86.5	**97.2**	81.7	94.3	89.5
	HIDDEN + PART	87.2	**97.2**	80.4	**94.6**	89.8
	HIDDEN + PART + EVENT	**87.3**	97.0	**82.6**	93.4	**89.9**
	Text Only	46.8	93.8	66.4	93.9	66.7

Table 2: Scores achieved with ensembles of selected labels, each of which is picked out as a result of benefiting certain types of questions.

appending additional roles to each identified entity, i.e. adding more extrinsic information which frequently matches with answers. Note that the *Hidden* label is the dominant among participants, as it appears approximately in all generative questions, and compared with the text-only method, there is a noticeable improvement after adding *Hidden* to it. First, introducing more applicable labels has a positive impact on the overall accuracy. We observe the triplet leads to overall best performance. Second, plain text behaves consistently to what we find in answering Q1, which performs worse in tasks like generative questions, however achieving strong performance in the extractive questions.

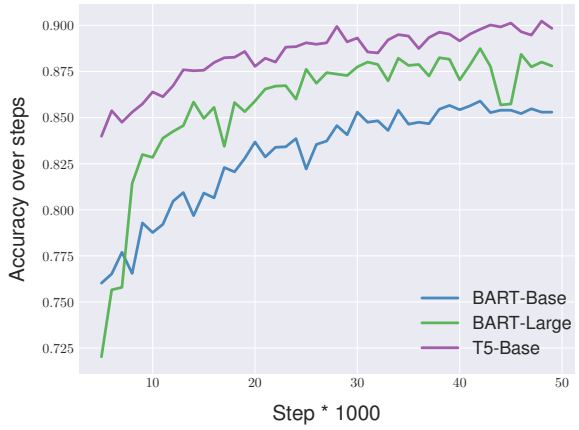

Figure 2: Models accuracy on validation dataset report every 1k step for three representative models.

To answer Q3, the tested models need to be comparable. We could say if two models (1) are roughly of the same amount of parameters (i.e. the same size). (2) share similar architectural design methodology (e.g. transformer-based), they would be considered comparable. We list the performance comparison of five models for 26k training steps in Table 3. We see that BART-Large is on a par with T5-Base from many aspects but performs poorly on a number of reasoning and generative questions. This is likely because T5 is fine-tuned on a more diverse mixture of tasks along with a very large pre-training dataset. Figure 2 demonstrates that BART-Large performs poorly at the beginning, taking off at around the 10k steps i.e. the second epoch. Moreover, later in the training stage, T5-Base has a slight advantage in achieving low variance. By contrast, the performance of BART-Large drops.

	Model	GEN	CLS	NUM	EXT	Overall
Enclosed by ( ) HD + PT + ET	BART-Base	82.9	94.2	74.7	92.7	86.0
	BART-Large (850 MB)	85.6	96.8	78.5	93.0	88.4
	T5-Base (950 MB)	87.0	96.4	81.7	**93.9**	89.6
	T5-Large	87.3	97.0	**82.6**	93.4	89.9
	T5-3B *	**88.6**	**97.4**	81.5	93.6	**90.5**

Table 3: Models comparison between T5 and BART of multiple sizes, result of T5-3B without error correction is used as our final prediction model. Note that T5-3B is 4 times the size of T5-Large. HD, PT and ET are short for HIDDEN, PART and ENTITY labels. Note that those labels are defined by the R2VQ dataset.

3.3 Error Analysis

Table 6 gives detailed examples including denoting mistakes of our suboptimal models i.e. T5-Large, which is slightly worse than our best T5-3B. As shown in questions 1 and 3, the model has a tendency to include more unrelated label information when answering generative questions. However, in question 4, it ignores some non-label important ingredients. That is in part because the transformer's attention mechanism sometimes fails at choosing whether to attend to labels or not when filling the answer template. Additionally, the worst performnace is for the 'number reasoning' question type, which is very challenging given that it needs to pay attention to multiple labels in combination. This could possibly be improved by further re-designing the transformer block or by including a memory block over the context.

3.4 Best-performing Submission

Even though we submitted multiple systems throughout the evaluation phase, our best-performing submission is the model that uses T5-3B and integrates the label-enclosing approach based on round brackets "( )". This system achieved an overall 91.3 in the test set, attaining the 2nd position in the competition.

4 Conclusion

In this paper, we describe the participation of the HIT&QMUL team in the R2VQ shared task, where

we ranked second. Our model is based on a unified generative text-to-text approach, in which we propose a novel label-enclosed input technique to include annotation labels to include semantic and cooking role labels. Our model achieved an exact match accuracy of 91.3, well over the baseline model (65.3) and only slightly behind the top-ranked system (92.53). Table 7 lists the top five final results on the R2VQ test set from all user submissions ordered by Exact Match score.

Through our comparative study, "$" enclosed labels proved to be best, with the most effective generative answering ability. A combination of HIDDEN, PART and ENTITY provides the best set of labels. Our study of the label-enclosing approach has some limitations given our focus on a small number of experimental label combinations. In future work, more analysis can be conducted exploring other label combinations potentially leading to further improved performance. In addition, the error analysis reveals that the model sometimes lacks the ability to attend to related labels possibly due to attention decay. Deeper investigation of this is left for future work.

References

Hedi Ben-Younes, Rémi Cadene, Matthieu Cord, and Nicolas Thome. 2017. Mutan: Multimodal tucker fusion for visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2612–2620.

Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. 2021. GSum: A general framework for guided neural abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4830–4842, Online. Association for Computational Linguistics.

Luheng He, Kenton Lee, Omer Levy, and Luke Zettlemoyer. 2018. Jointly predicting predicates and arguments in neural semantic role labeling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 364–369.

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017. Deep semantic role labeling: What works and whats next. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 473–483.

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.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

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*, 21:1–67.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Jingxuan Tu, Marco Maru, Eben Holderness, Kyeongmin Rim, Kelley Lynch, Simone Conia, Richard Brutti, Roberto Navigli, and James Pustejovsky. Semeval-2022 task 9: R2vq competence-based multimodal question answering. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Lili Yao, Nanyun Peng, Ralph Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. 2019. Plan-and-write: Towards better automatic storytelling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7378–7385.

Appendix

A Label-enclosed Example

Enclosing Method	Example
( )	Stirring (drop : onion mixture # tool : spoon) frequently, until the onions (participant : turned) have turned golden brown.
#	Stirring # drop : onion mixture # tool : spoon # frequently , until the onions # participant : turned # have turned golden brown .
$	Stirring $ drop : onion mixture # tool : spoon $ frequently, until the onions $ participant : turned $ have turned golden brown.
[BOL] [EOL]	Stirring [BOL] drop : onion mixture # tool : spoon [EOL] frequently, until the onions [BOL] participant : turned [EOL] have turned golden brown.
Parallel	Stirring frequently, until the onions have turned golden brown. [space] drop : onion mixture tool : spoon [space] [space] [space] [space] [space] participant : turned [space] [space] [space] [space] [space]

Table 4

B Question Family Categorisation

Catagory	Question Family
Number Reasoning (NUM)	Cardinality
Classification (CLS)	Event Ordering, Unanswerable
Generative (GEN)	Implicit Argument Identification, Ellipsis, Object Lifespan
Extractive (EXT)	Coreference Location Change, Attribute, Temporal, Result, Cause, Co-Patient

Table 5

C Error Example

Context
1. Mash eggs & mix with salad cream or mayonnaise. If you prefer a sweeter taste, go with salad cream. I like mine with mayo.
2. Clean & devein prawns; separate the heads. Blanch prawns & heads.
3. Drain & transfer to ice water to prevent them from over-cooking. Dice prawns & set aside.
4. Remove apple & mango skin & dice fruits into small cubes. Soak apple in salt water, lemon juice or cider vinegar to prevent it from browning. I usually use the traditional method of soaking in salt water handed down by my mom. Add them into egg mixture, together with the diced prawns.
5. Remove excess sauce from beancurd skin & stuff mixture into the pockets. You may have to cut & adjust the pocket height.
6. Cut up watermelon & start plating your dish. Chill your appetiser & youre ready impress your guests with this Inari Age Laughing Prawns Salad.

Question	T5-Large	Gold Answer
1. How did you get the mixture?	by adding the diced prawns, apple, mango and prawns to the bowl	by adding the apple, mango and prawns to the bowl
2. How do you soak apple to prevent it from browning?	soak apple in salt water, lemon juice or cider vinegar	soak apple in salt water , lemon juice or cider vinegar
3. What's in the inari prawn salad?	the pocket height and pockets	the pockets
4. What should be diced?	the apple mango	the fruits, apple and mango
5. How many times is the pot used?	2	3

Table 6

D Leader Board

Username	EM	F1
t.dryjanski	92.53	94.34
weihezhai	91.34	94.23
ruan	78.21	82.62
kartikaggarwal98	69.49	77.37
r2vq (baseline from organizers)	65.34	75.22

Table 7