**Frontiers in Computing and Intelligent Systems**
ISSN: 2832-6024 | Vol. 4, No. 2, 2023

# A Question Answering System for Situation Puzzle with SPQA

**Tian Zhao ***

Mathematics Department, City University of Hong Kong, Hong Kong SAR, 999077, China
*** Corresponding author Email:** zhaotian8858@gmail.com

**Abstract:** There are many questions answering (QA) system built for solving QA tasks. In 2020 and 2022, Allen Institute and the University of Washington proposed UnifiedQA and UnifiedQA-v2. Their core concept is that the semantic understanding and reasoning capabilities required by models are common, and may not require format specific models although the QA task forms are different. Behind this concept, I build a new QA model named SPQA, aiming to answer the situation puzzle questions by adding new situation-puzzle related dataset (SpQ). In addition, I evaluate the performance of SPQA and UnifiedQA-v2 for fine-tuning and prompt-tuning. The results of fine-tuning indicate that SpQ dataset is important for fine-tuning and prompt-tuning to answer situation puzzle questions well, but also make the answering ability of normal yes/no questions worse. Eventually, the results of prompt-tuning indicate that the effects of SpQ is larger and more significant on situation puzzle questions and normal yes/no questions under the same data scale. In the future work, the further research like building larger SpQ dataset should be considered.

**Keywords:** NLP; Question Answering; Situation Puzzle; UNIFIEDQA; UNIFIEDQA-V2; SPQA.

## 1. Introduction

Situation puzzles (also called “lateral thinking puzzles” or "yes/no puzzles”) are usually played by a group of players. The players asking questions which can only be answered with "yes" or "no" to the person who is hosting the game. Depending on the settings and difficulty of the puzzle, some information can be added in the answers, such as hints, simple explanations about why the answer is that, or be informed by “not related”. The puzzle is informed by “solved” when one of the players can state the same process or truth as the host’s thought [1].

### 1.1. Background

In 2017, Transformer (Vaswani et al.) has been proved for machine translation problem [2] and then widely used in various NLP problem (Radford et al., 2018; Devlin et al., 2018; McCann et al., 2018; Yu et al., 2018) [3-6].

In 2018, BERT was proposed by Google. Its “bidirectional encoder representation from transformers” was awarded the Best Long Paper Award at the 2019 North American Branch of the Association for Computational Linguistics (NAACL), and its performance on 11 NLP tasks has set a new record [7].

In the same year, the GPT model proposed by OpenAI can be migrated to NLP [8].

In 2019, BART method combined BERT and GPT model [10]. Its “Bidirectional and Auto-Regressive Transformers” built a pre training language model by Transformer model with encoder-decoder structure [9].

In the same year, with the introduction of a large-scale pre training model, almost all NLP tasks became "pre-train to fine-tune" mode. Instead of modifying the pre training model itself, people generally introduced few additional parameters (network layer) to complete downstream tasks by setting various objective functions. At this point, the focus of work has shifted to objective function engineering [10].

In 2020, Google released the T5 model. Its most important role is to provide a common framework for the entire NLP pre training model field, transforming all tasks into one form. After that, the main task became how to convert tasks into appropriate text-input and text-output [11].

However, as pre-trained language models (PLMs) become larger and larger, the requirements of hardware, data, and actual costs are also increasing. What’s more, the design of the pre-training and fine-tuning stages become complex as the result of the large and diverse downstream tasks. In order to explore smaller, more lightweight, and more universal and efficient methods, researchers attempt to use “Prompt” method. In 2021, "pre train, prompt, and predict" was introduced and the original "pre-train to fine-tune" mode has gradually been replaced by this mode. People no longer use customized objective function engineering to adapt pre training models to downstream tasks. Instead, various downstream tasks are redefined under a short text prompt to resemble as much as possible the problem forms that PLMs solves during training [12].

### 1.2. Related Works

QA tasks is a kind of downstream tasks in NLP. Although the QA task forms are different, the semantic understanding and reasoning capabilities required by models are common, and may not require format specific models. Based on this concept, Allen Institute and the University of Washington proposed the first pre training question answering model, UnifiedQA, on EMNLP in November 2020, which can handle multiple forms of questions and answers, becoming a new SOTA for multiple question answering tasks. All NLP tasks can be converted to seq2seq tasks. Based on the same idea, UnifiedQA is a text-to-text pre training question answering model. The encoder receives questions spliced with "\n", and the decoder generates answers [13]. In 2022, the original team only added more pre training datasets to the original UnifiedQA for pre training, which further improved the performance of the model on both the "seen" dataset and the "unseen" dataset to UnifiedQA-V2[14].

At the same time, GPT-2 had 1.5 billion parameters in

2019[15]. In 2020, GPT-3 already had an astonishing 175 billion parameters [16]. In 2022, InstrumentGPT and ChatGPT [17]. On March 14, 2023, GPT-4 was released [18].

This paper focuses on constructing a question answering system based on SPQA for situation puzzles. Regarding UNFIEDQA-V2 model as original model and adding situation puzzles training set, I constructed a situation-puzzle QA model (SPQA) and then a SPQA prompt tuning model. Eventually, I compared the appearance of UnifiedQA, SPQA, SPQA-prompt and ChatGPT (GPT-3.5) on solving situation-puzzle problem.

# 2. Methodology

In this paper, I used two methods: UnifiedQA (v1 and v2) multi-format training and fine tuning, and parameter-efficient prompt tuning aim to evaluate the performance of adding spQ datasets.

## 2.1. Multi-format Training in UnifiedQA

Firstly, I want to train a SPQA model that can operate over $k$ formats $F_1, F_2, \dots, F_k$, like the structure of UnifiedQA model. For each format $F_i$, there is $\ell_i$ datasets set: $D_1^i, D_2^i, \dots, D_\ell^i$, where $D_j^i = (T_j^i, E_j^i)$, which includes training set $T_j^i$ and evaluation set $E_j^i$. If the dataset is considered to be used only for evaluation, I will ignore the $T_j^i$ aim to treat $D_j^i$ as an "unseen" dataset. In SPQA model, the "unseen" dataset only includes "yes/no questions" format datasets.

In pre-processing progress, I also transfer each training question $q$ in format $F_i$ into a plain-text input representation $enc_i(q)$, which is the same as that of UnifiedQA training datasets. I use the UnifiedQA approach of creating a mixed training pool including all available training examples:

$$\tilde{T} = \cup_{i=1}^{k} \cup_{j=1}^{\ell_i} \{enc_i(q) | q \in T_j^i\} \quad (1)$$

## 2.2. Adapter Tuning and Parameter-Efficient Prompt Tuning

Adapter tuning is related to multi-task and continual learning but also differ because the tasks don't interact and the shared parameters are fixed, which indicates that the model can remember previous tasks perfectly by using few task-specific parameters. Parameter-Efficient Prompt Tuning (also called soft-prompt tuning) proposed the use of adapter modules to transfer, thereby creating a compact and extensible model. Only a few trainable parameters added in each task, and new tasks can be added without the need to revisit previous ones [19].

# 3. Experiment

According to the paper of UnifiedQA, its concept is suitable for text-to-text encoding and therefore used T5 and BART to reach this multi-task target. UnifiedQA eventually used T5-11B and BART-large as the starting point to pretrain. For the further research, UnifiedQA-v2 is trained on 20 datasets while UnifiedQA is trained on 8 datasets. In addition, UnifiedQA-v2 is trained for 350k steps and UnifiedQA is trained for 100k steps [13-14].

In this paper, I also use T5 as the starting point to pretrain. Firstly, I collect, generate and pre-process some situation puzzle data and put them into the situation puzzle dataset (SpQ), and then train 20 UnifiedQA-v2 datasets + spQ dataset in the model. In the end, I fine-tune them by TPU and prompt-tune them by GPU and discuss the differences between the SPQA and UnifiedQA-v2.

## 3.1. Situation Puzzle Dataset

**Table 1.** Situation Puzzle Dataset Example

| Questions | Answers |
|---|---|
| am I a tramp and I will die because of the cold? | No |
| is my coming death related to my career? | Yes |
| are my pants different from the ordinary pants? | Yes |
| are my pants related to my career? | Yes |
| are my pants used to ensure my safety? | Yes |
| is my career about underwater works? | No |
| is my career about underground works? | No |
| is my career about ground works? | No |
| is my career about high altitude works? | No |
| is my career about space works? | Yes |

In this paper, I added some datasets about situation puzzle. Take an example, the contents of the story are: "My pants are torn, I know I'm going to die soon. Because I am an astronaut. One day, I was carrying out a mission in space wearing a spacesuit when I suddenly noticed that my pants were torn. Afterwards, I was exposed to space without air pressure and oxygen, and in less than a few seconds, I would die." The questions and answers show in the table 1.

## 3.2. Training Models and Datasets

In this paper, like the construction of UNIFIEDQA and UNIFIEDQA-v2, I use the T5 architecture (Raffel et al., 2020) for SPQA and train it for *102k*.

Training datasets shows as follows: **SPQA** (21 datasets): SQuAD 1.1, SQuAD 2, NewsQA, Quoref, ROPES, NarrativeQA, DROP, NaturalQuestions, MCTest, RACE, OpenBookQA, ARC, CommonsenseQA, QASC, PhysicalIQA, SocialIQA, Winogrande, BoolQ, MultiRC (yes/no), BoolQ-NP, SpQ.

## 3.3. Details on the Experiments

Some details on the experiments shows as follows:

(1) Models: T5(3B-TPU) and T5(GPU).

(2) Model sizes: Mostly T5(3B-TPU) which has 3 billion parameters.

(3) Input/output size: Use token-limits of size 512 and 100 for inputs and outputs.

(4) # of iterations for pretraining on the seed datasets: All models are trained for *102k* on the seed datasets.

(5) Learning rates: Use 3e-3 for T5(3B-TPU) and T5(GPU).

(6) Batch sizes: Use batches of 16 for the T5(3B-TPU) and batches of 2 for T5(GPU).

(7) Infrastructure: Use v2-8 TPUs for T5(3B-TPU) models and 16G GPUs for T5(GPU) models.

(8) Fine tuning on datasets: Fine-tuned for *102k* steps and checkpoints were saved per *20k* steps.

# 4. Evaluation and Results

In this paper, I compared the SPQA with the UnifiedQA-V2 and evaluate a fixed checkpoint across the target datasets: BoolQ, BoolQ-np, BoolQ-CS (unseen), SpQ and SpQ-test (unseen): both checkpoint 100k for SPQA and UnifiedQA-V2. In addition, I discuss the prompt tuning between the SPQA and UnifiedQA-v2. Eventually, I observe the characteristics

of the answers given by the SPQA, UnifiedQA-v2 and chatGPT (GPT3.5).

Evaluation datasets shows as follows:

(1) BoolQ (Clark et al., 2019), BoolQ-NP (Khashabi et al., 2020a) the binary (yes/no) subset of MultiRC (Khashabi et al., 2018)

(2) BoolQ-CS (unseen) from StrategyQA (Geva et al., 2021) and PubmedQA (Jin et al., 2019).

## 4.1. Metrics

I evaluate each dataset via their common metric by the accuracy. For Yes/No questions, if the model gives the correct answer ("yes" or "yes, it is right."), it gains one score. Otherwise, it gains no score. In addition, I also provide "aggregate scores" that compare the two models. For "aggregate scores", it gives two metrics:

1.the difference between the average performance score of SPQA and UnifiedQA-v2 models of the same size (indicated with 'SP – Uni2');

2.the percentage that SPQA causes a better performance than UnifiedQA-v2 of the same size (indicated with 'SP ≥ Uni2?') [14].

## 4.2. Evaluation

I evaluate each bool QA dataset via their common metric.

**Table 2.** Evaluation between UnifiedQA-v2 and SPQA with fine-tuning

| | Boolq-dev | | Boolq-np-dev | | Boolq-CS-dev(unseen) | | spQ-dev | | spQ-test(unseen) | |
|---|---|---|---|---|---|---|---|---|---|---|
| | UnifiedQA-v2 | SPQA | UnifiedQA-v2 | SPQA | UnifiedQA-v2 | SPQA | UnifiedQA-v2 | SPQA | UnifiedQA-v2 | SPQA |
| small | 70.795 | 70.306 | 63.494 | 59.795 | 36.560 | 34.553 | 65.000 | 65.000 | 70.000 | 75.000 |
| base | 76.177 | 78.960 | 72.367 | 73.486 | 44.510 | 46.084 | 75.000 | 85.000 | 70.000 | 85.000 |
| large | 77.829 | 81.865 | 74.987 | 76.764 | 45.612 | 47.580 | 80.000 | 85.000 | 95.000 | 95.000 |
| 3B | 86.606 | 85.291 | 83.149 | 81.082 | 49.587 | 47.658 | 85.000 | 80.000 | 100.000 | 100.000 |

**Table 3.** Aggregate scores that contrast the two models with fine tuning

| | Average | | Aggregated | |
|---|---|---|---|---|
| | UnifiedQA-v2 | SPQA | SP − Uni2 | SP ≥ Uni2? |
| small | 61.170 | 60.931 | -0.239 | 33 |
| base | 67.611 | 73.706 | 6.095 | 100 |
| large | 74.686 | 77.242 | 2.556 | 100 |
| 3B | 80.868 | 78.806 | -2.062 | 17 |

**Table 4.** Evaluation between UnifiedQA-v2 and SPQA with prompt-tuning

| | Boolq-dev | | Boolq-np-dev | | Boolq-CS-dev(unseen) | | spQ-dev | | spQ-test(unseen) | |
|---|---|---|---|---|---|---|---|---|---|---|
| prompt | UnifiedQA-v2 | SPQA-prompt | UnifiedQA-v2 | SPQA-prompt | UnifiedQA-v2 | SPQA-prompt | UnifiedQA-v2 | SPQA-prompt | UnifiedQA-v2 | SPQA-prompt |
| small | 0.750 | 0.650 | 0.550 | 0.500 | 0.700 | 0.700 | 0.500 | 0.400 | 0.550 | 0.550 |
| base | 0.550 | 0.550 | 0.150 | 0.300 | 0.700 | 0.700 | 0.500 | 0.400 | 0.300 | 0.500 |
| large | 0.550 | 0.550 | 0.750 | 0.600 | 0.700 | 0.650 | 0.450 | 0.450 | 0.300 | 0.650 |

**Table 5.** Aggregate scores that contrast the two models with prompt-tuning

| | Average | | Aggregated | |
|---|---|---|---|---|
| | UnifiedQA-v2 | SPQA | SP-Uni2 | SP ≥ Uni2? |
| small | **0.610** | 0.560 | -0.050 | 33 |
| base | 0.440 | **0.490** | 0.050 | 83 |
| large | 0.550 | **0.580** | 0.030 | 67 |

## 4.3. Results

Summarize the TPU (v2.8) *fine-tuning* results from Table 2 and Table 3. In all experiments, SPQA causes 2.23% performance improvements over UNIFIEDQA-v2, on average ('SP – Uni2'). The highest gains appear on mid-sized 'base' models (9.01% for overall, 6.22% for in-domain and 14.4% for out-of-domain). On the contrary, the lowest gains appear on the extreme sizes ('small' and '3B').

Similar tendency shows with 'SP ≥ Uni2?' metric (percentage that SPQA outperforms UNIFIEDQA-v2). On this metric, all the numbers are only above 10%, which demonstrates that SPQA models generally don't causes better performance on all yes/no QA datasets. However, on SpQ-test(unseen), particularly, the numbers are always 100%, which demonstrates that SPQA models causes better performance on all situation puzzle QA datasets. In addition, SPQA of size'base' outperforms UNIFIEDQA-v2 of the same size on both 100% of the datasets, for in-domain and out-domain datasets.

According to the results of *fine-tuning* part, UNIFIEDQA-v2 always keeps the performance well, especially on normal yes/no questions for small and 3B. On the other side, the SPQA always keeps better performance than UNIFIEDQA-v2, especially on situation puzzle questions from small to 3B. When the model scales are 'base' and 'large', SPQA is faster than UNIFIEDQA-v2 to get the receptable results. When the model scale reaches to 3B, UNIFIEDQA-v2 begin to surpassed SPQA on all the yes/no questions, which means the situation puzzle training dataset may affect the system to judge the normal questions.

Summarizing the GPU *prompt-tuning* results from Table 4 and Table 5. In all experiments, SPQA-prompt causes 1.88% performance improvements over UNIFIEDQA-v2-prompt, on average ('SP – Uni2'). The highest gains appear on mid-sized 'base' models (11.36% for overall, 4.17% for in-domain and 20.0% for out-of-domain). On the contrary, the lowest gains appear on the extreme size ('small').

Similar tendency shows with 'SP ≥ Uni2?' metric (percentage that SPQA-prompt outperforms UNIFIEDQA-

v2-prompt). On this metric, all the numbers are only above 30%, which demonstrates that SPQA models generally don't causes better performance on all yes/no QA datasets. However, on SpQ-test(unseen), particularly, the numbers are always 100%, which demonstrates that SPQA models causes better performance on all situation puzzle QA datasets. In addition, SPQA of size'base' outperforms UNIFIEDQA-v2-prompt of the same size on 66.7% and 100% of the datasets, for in-domain and out-domain datasets, respectively.

According to the results of *prompt-tuning* part, all datasets trained under the same data scale. UNIFIEDQA-v2 and SPQA keep the similar performance for 'small' and 'base'. For 'large' model, the situation puzzle dataset affects more significantly, make the performance of SpQ test is better and make the performance of other normal yes/no datasets worse.

### 4.4. SPQA vs UnifiedQA vs chatGPT (GPT-3.5)

According to the Table 6, I take some examples as the reference. When the contents of a situation puzzle are about the crime or the negative information, the chatGPT will reject the requests because of the legal reasons even though this topic is just a story. When the contents of a situation puzzle are about the other aspects, the chatGPT may sometimes give the answers such as "The information given is not clear", "can't answer and need more information" or "Not mentioned in the story", means that chatGPT need more information about the stories clearly mentioned in these questions. In these situation-puzzle questions, chatGPT has a serious attitude and will refuse to answer questions without clear answers, like serious humans. However, for this point, the rules need the model to answer only by "yes" or "no", so the answers sometimes given by chatGPT is unacceptable.

**Table 6.** The performance of SPQA and UnifiedQA-v2 compared with chatGPT

| | Situation Puzzle with crime | Situation Puzzle without crime |
|---|---|---|
| chatGPT | Refuse to answer | "The information given is not clear" "Not mentioned in the story" If it answered, the answers are right. |
| UnifiedQA-v2 | (Small) still have wrong answers neither "yes" nor "no" | (Small) still have wrong answers neither "yes" nor "no" |
| SPQA | Judge the task without mistakes, but the correct rate is properly the same as UnifiedQA-v2. On situation puzzle questions, the performance is faster to reach the target. | Judge the task without mistakes, but the correct rate is properly the same as UnifiedQA-v2. On situation puzzle questions, the performance is faster to reach the target. |

## 5. Conclusion

In this paper, I used the T5 architecture, like UNIFIEDQA's structure, trained and fine-tuned a new model named SPQA to answer situation puzzle questions to reach the requests by adding a new dataset SpQ on TPU-v2.8. In addition, I used less datasets on GPU to train and prompt-tuned a new model titled SPQA-prompt to answer situation puzzle questions under the same dataset scale. According to the performance as above, the conclusion is that situation puzzle datasets (spQ) is important for fine-tuning and prompt-tuning to answer situation puzzle questions, but also make the answering ability of normal yes/no questions worse. For fine-tuning, the performance of SPQA for '3B' is good enough at the current stage. However, because of TPU and GPU limit and the data scales of SpQ dataset, the effects of larger SpQ dataset, data scale of '11B' and prompt-tuning with larger datasets cannot considered in this paper. In addition, the performance and gains are not quite uniform for all datasets. I will try to build and train the model only with large SpQ dataset and other further research in the future work.

## Acknowledgments

TPU for performing experiments and researches were provided by GCPs of Google.

## References

[1] Jed Hartman, Rec.puzzles archive 27 Aug 1998 http://www.kith.org/logos/things/sitpuz/lateral.html.

[2] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.

[3] Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training.

[4] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[5] McCann, B., Keskar, N. S., Xiong, C., & Socher, R. (2018). The natural language decathlon: Multitask learning as question answering. arXiv preprint arXiv:1806.08730.

[6] Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., & Huang, T. S. (2018). Generative image inpainting with contextual attention. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 5505-5514).

[7] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[8] Radford, A., & Salimans, T. (2018). GPT: Improving Language Understanding by Generative Pre-Training. arXiv.

[9] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., ... & Zettlemoyer, L. (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461.

[10] Zhang, Y., Sun, S., Galley, M., Chen, Y. C., Brockett, C., Gao, X., ... & Dolan, B. (2019). Dialogpt: Large-scale generative pre-training for conversational response generation. arXiv preprint arXiv:1911.00536.

[11] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res., 21(140), 1-67.

[12] Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2021). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. arXiv preprint arXiv:2107.13586.

[13] Khashabi, D., Min, S., Khot, T., Sabharwal, A., Tafjord, O., Clark, P., & Hajishirzi, H. (2020). Unifiedqa: Crossing format boundaries with a single qa system. arXiv preprint arXiv: 2005.00700.

[14] Khashabi, D., Kordi, Y., & Hajishirzi, H. (2022). Unifiedqa-v2: Stronger generalization via broader cross-format training. arXiv preprint arXiv:2202.12359.

[15] Budzianowski, P., & Vulić, I. (2019). Hello, it's GPT-2--how can I help you? towards the use of pretrained language models for task-oriented dialogue systems. arXiv preprint arXiv: 1907. 05774.

[16] Floridi, L., & Chiriatti, M. (2020). GPT-3: Its nature, scope, limits, and consequences. Minds and Machines, 30(4), 681-694.

[17] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., ... & Lowe, R. (2022). Training language models to follow instructions with human feedback. arXiv preprint arXiv: 2203.02155.

[18] OpenAI. (2023). GPT-4 Technical Report. .https:// openai. com/ product/gpt-4.

[19] Lester, B., Al-Rfou, R., & Constant, N. (2021). The power of scale for parameter-efficient prompt tuning. arXiv preprint arXiv:2104.08691.