# CSE 256 Final Project: Text Visual Question Answering with Text Correlation Prediction

**Sifan Li**
sil100@ucsd.edu

## 1 Abstract

The TextVQA (Text Visual Question Answering) task requires the model to read and understand the visual and textual information within images to respond to presented questions. It requires the use of information from three modalities: question, image vision, and scene text. Among them, the scene text information in the image is crucial for correctly understanding the scene and answering questions. Existing TextVQA models indiscriminately input all texts recognized by the OCR system into subsequent models. Redundant text will interfere with answer generation and affect the accuracy of answer prediction.

In order to solve the above problems, I propose a TCM (Text Correlation Model) based on text correlation prediction. The model consists of a feature extraction module, an STA (Scene Text Augmenter) and an answer generation module. TCM initially processes the extracted question, visual, and textual features through the STA to capture key textual information relevant to specific question-answer pairs. Subsequently, it is integrated with the question and visual information and fed into the Large Language Model to produce predictive outcomes. The STA, as introduced in the report, is capable of predicting the relevance between scene text and both the question and image. It utilizes the multi-head attention mechanism to select scene text most pertinent to the question and image, thereby enhancing the model's focus on critical textual information. Experimental results on two datasets validate the effectiveness of the proposed method.

## 2 Introduction

The topic I chose to study is Text Visual Question Answering (TextVQA). It requires the model to understand and use the text information in the image to answer questions. TextVQA needs to use information on three modalities to infer the answer, namely the semantic information in the input question, the visual information in the input image, and the scene text information in the input image. Among them, the scene text information in the image is crucial for understanding the scene and answering questions correctly. Various optical character recognition (OCR) methods are used to extract text from images to prepare for further text visual question answering. However, the problem is existing OCR systems tend to output more text in the image, which are not necessarily valid for specific questions. If all texts are directly input into the subsequent model without processing, it will interfere with question answering and affect the accuracy. To solve the above problems, I proposed a Text Correlation Model(TCM), which includes a Scene Text Augmenter (STA). It is a method to predict the relevance of scene text and question-image.

I planned to complete the following:

- Collect and preprocess two datasets: DONE.
- Build and train baseline model on collected datasets and examine its performance: DONE.
- Build my model TCM by adding a scene text augmenter to capture key textual information relevant to specific question-answer pairs: DONE.
- Train and evaluate TCM on collected datasets, and perform case study: DONE.

## 3 Related work

Text features in images are an important part of the TextVQA task. Making full use of scene text information is crucial to improving the accuracy of question answering. In recent years, various models have designed some methods to utilize scene text features. Signh (Singh et al., 2019) first introduced a new dataset TextVQA, which contains questions that require the model to answer by reasoning about image information. They also proposed the LoRRA model, which uses the OCR system as one of the modules to extract text information from the image and combines it with a fixed vocabulary to form an answer space. The M4C model uses an external OCR system to extract four types of text features, including FastText vectors, appearance features, PHOC vectors, and position features (Hu et al., 2020). SS-Baseline also extracts these four types of information like M4C, but further divides them into semantic information and visual information, and further utilizes text features through corresponding attention branches (Zhu et al., 2021).

Inspired by the success of pre-training in NLP tasks, some studies have also adopted pre-training strategies in scene text question-answering tasks. TAP proposed text-aware pre-training, which better learned multimodal representations through three pre-training tasks: masked language modeling, image-text comparison matching, and relative spatial position prediction (Yang et al., 2021). The LaTr model chooses to perform layout-aware denoising pre-training on documents, and uses weak data without answer annotations in the pre-training stage to encourage the model to make full use of text and layout information (Biten et al., 2022). PreSTU proposed a pre-training method designed for scene text understanding, with a pre-training target called SPLITOCR, which operates from image pixels and combines the OCR-aware pre-training target with a large-scale image-text dataset with OCR information (Kil et al., 2023).

# 4 Dataset

## 4.1 Dataset introduction

I use the TextVQA dataset (Singh et al., 2019) and the ST-VQA dataset (Biten et al., 2019) for experiments and evaluation.

The TextVQA dataset includes 28,408 images. It uses the Open Image v3 dataset as the image source, including 45,336 questions about scene text (37,912 different questions) and 453,360 answers (26,263 different answers). Each image has 1-2 questions, and each question has 10 answers. The questions are divided into training set (34,602), validation set (5,000) and test set (5,734). For each category of images, 100 images are randomly selected and passed through the OCR model Rosetta to obtain the average number of OCR boxes, which are normalized and used as weights.

Table 1: The source of ST-VQA dataset

| Source | Image number | Question number |
|---|---|---|
| Coco-text | 7,520 | 10,854 |
| Visual Genome | 8,490 | 11,195 |
| VizWiz | 835 | 1,303 |
| ICDAR | 1,088 | 1,423 |
| ImageNet | 3,680 | 5,165 |
| IIIT-STR | 1,425 | 1,890 |
| Total | 23,038 | 31,791 |

The ST-VQA dataset contains 23,038 images and 31,791 questions from six different datasets, namely Coco-text, Visual Genome, VizWiz, ICDAR, ImageNet, and IIIT-STR. ST-VQA reduces dataset bias by adopting multiple datasets. The images selected from each dataset and the number of questions set are shown in Table1. The training set contains 19,027 images and 26,308 questions, and the test set contains 2,993 images and 4,163 questions.

**Why do I use two datasets?** ST-VQA is conceptually similar to the TextVQA dataset. The main difference is that ST-VQA uses datasets from different sources, while all images in TextVQA come from the same dataset; ST-VQA requires that each image annotation contain at least two text instances, while TextVQA samples images according to the category containing text; about 39% of the answers to questions in the TextVQA dataset cannot be found from the OCR results, while ST-VQA can almost all answer questions using the text recognized in the image, with less ambiguity (Biten et al., 2019). The two datasets are highly complementary, so they can be combined to train models with stronger generalization capabilities.

## 4.2 Dataset examples

Figure 1 shows some examples from TextVQA dataset. The images, questions and ground truth answers are from the dataset. ”Rosetta-en” represents the image text recognized by the OCR system, and blue font represents the key text related to the question answering. Blue font in ”key text” means it is in the OCR recognition list, and red means the OCR system does not recognize the key text.

For Figure (a), Rosetta-en successfully recognizes three key words; in Figure (b), the key text ”Music” and ”player” are not recognized; in Figure (c), part of the key text is recognized, and ”WISLER” is mistakenly recognized as ”WISLE”. From these examples, it can be seen that the accuracy of text visual question answering tasks depends on the results of OCR system recognition, and how to use these recognized texts is a core issue. For different questions, the key text will be different. If we can find the image text that is most relevant to the question answering and use it for the input question and image, it will help improve the accuracy of question answering.

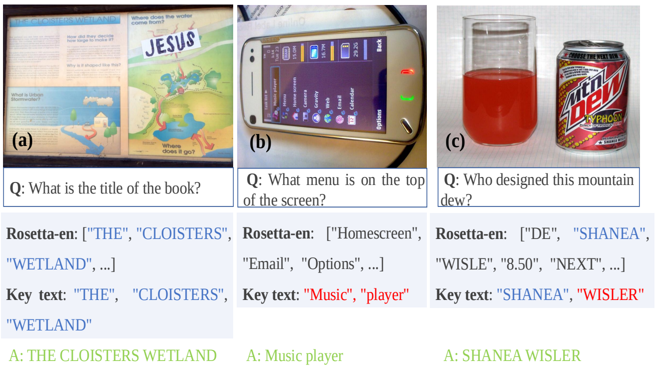
(a)
Q: What is the title of the book?
Rosetta-en: ["THE", "CLOISTERS", "WETLAND", ...]
Key text: "THE", "CLOISTERS", "WETLAND"
A: THE CLOISTERS WETLAND
(b)
Q: What menu is on the top of the screen?
Rosetta-en: ["Homescreen", "Email", "Options", ...]
Key text: "Music", "player"
A: Music player
(c)
Q: Who designed this mountain dew?
Rosetta-en: ["DE", "SHANEA", "WISLE", "8.50", "NEXT", ...]
Key text: "SHANEA", "WISLER"
A: SHANEA WISLER

Figure 1: TextVQA exapmles

### 4.3 Dataset preprocessing

The OCR system used in the TextVQA dataset is Rosetta-en. In order to match the training settings of the TextVQA dataset, I use three existing OCR systems (Rosetta-en, SBD-Trans and Amazon-OCR) to filter and merge the recognition results of ST-VQA. At the same time, in order to make the experiment go more smoothly, I removed the question-image pairs that have empty OCR outputs. Finally, TextVQA is divided into training set (34231), validation set (4926), and test set (5622). ST-VQA is divided into training set (10492), validation set (2573), and test set (2628).

## 5 Baselines

### 5.1 Baseline introduction

My baseline is based on the architecture in LaTr (Biten et al., 2022) and uses simple feature extraction and answer generation, and does not further process the text information in the image.

The input of the model is the image and the question. For a given question consisting of K words, the pre-trained BERT-base model is used to extract the question text features (Devlin, 2018). For a given image, the Vision Transformer model is used to extract the visual features of the image (Dosovitskiy, 2020). Scene text information needs to be extracted from the image. The OCR system is used as a module in the model, and the extracted OCR mark can be obtained after the image is passed through OCR. In order to more comprehensively represent the text in the image, in addition to encoding the characters, it is also necessary to know where the text is in the image. Therefore, a spatial information extraction module is added. The bounding box of a word associated with an OCR tag $O$ can be defined as $(x_0, y_0, x_1, y_1, h, w)$, where $(x_0, y_0)$ represents the upper left corner of the bounding box, and $(x_1, y_1)$ represents the lower right corner of the bounding box. Since the text in the image can appear in any shape and size, in order to eliminate this interference, the height $h$ and width $w$ are added. By adding spatial information, the semantic representation of the corresponding text can be associated with its position information. The encoded representation of the OCR tag is the sum of its semantic feature encoding and the spatial feature encoding of the word bounding box, as shown in formula 1.

$$\begin{aligned}\varepsilon = E_O(ocr\_token) + E_x(x_0) + E_y(y_0) \\ + E_x(x_1) + E_y(y_1) + E_h(h) + E_w(w)\end{aligned} \tag{1}$$

Therefore, I can get the feature embeddings $F_q$, $F_v$, $F_{ocr}$. Finally, the image features, question word features and new text features are uniformly input into the language model T5 (Raffel et al., 2020) to generate the answer.

In general, the baseline extracts features from three modalities: question, image, and scene text, concatenates the three embeddings $F_q$, $F_v$, and $F_{ocr}$, and inputs them into the multimodal Transformer.

### 5.2 Parameter settings

The parameter settings of the baseline model are shown in Table4.

The size of the input image is set to 640 × 640, and the maximum length of the question is 40. The maximum number of OCR tokens is 100. The BERT model used to embed questions and OCR

Table 2: Parameter settings of this model

| Hyperparameters | values |
|---|---|
| Maximum question length | 40 |
| Image size | 640×640 |
| Maximum number of OCR tokens | 100 |
| BERT hidden layer size | 768 |
| BERT output layer number | 4 |
| T5(ViT) layer number | 12 |
| T5(ViT) hidden layer size | 768 |
| T5(ViT) feedforward network dimension | 3072 |
| T5(ViT) attention head number | 12 |
| T5 dropout | 0.1 |
| Optimizer | Adam |
| Batch size | training:2 test:8 |
| Learning rate | 1e-5 |

tokens is initialized with the "bert-base-uncased" weights in HuggingFace, and the [CLS] and [SEP] tags are added to the beginning and end of the processed text. The ViT model used to embed images is initialized with the "google/vit-base-patch16-224-in21k" weights in HuggingFace. Note that the token generator used in BERT is WordPiece-Tokenier, which means that a word will be divided into one or more subwords, so a word does not necessarily correspond to only one token, it may correspond to one or more tokens.

# 6 My approach

## 6.1 My approach introduction

**Overview of TCM**

Based on the question of how to make more effective use of OCR text, this paper proposes a text visual question answering model (TCM) based on text relevance prediction. The structure of the model is shown in Figure 2, where squares of the same color represent features of the same mode.

TCM is implemented based on the framework of the LaTr (Biten et al., 2022) model. It can be divided into three parts: the first part is the extraction of three modal information: question, image, and text; the second part is a scene text augmenter (STA) to obtain the most relevant text; and the third part is a language model T5 for answer generation. Among them, the first and third are the same as my baseline, and the second part is my improvement.

The features extracted from these three modalities are passed through STA. STA consists of multiple multi-head attention modules, which are used to select the image text that is most relevant to the question-image, thereby reducing the interference of redundant OCR text. First, find the image visual feature information that is most relevant to the input question, then find the image-guided question features, get the question-image pair, and finally calculate the similarity score between the text features extracted by OCR and the question-image pair. At the same time, analyze the correlation between the entire OCR list and the question. If the predicted answer appears in the OCR list, the word with the highest similarity score is added to the original OCR list again.

**Overview of STA**

The purpose of the scene text enhancer (STA) is to select the OCR text that is most relevant to the question-image feature pair and decide whether to add the most relevant text to the OCR text list. The structure of STA is shown in Figure3.

(1) in Figure3 is the module for correlation calculation, which is based on the multi-head attention mechanism. Its purpose is to measure the contribution of the OCR text unit extracted from the image to the question answering, that is, to calculate the correlation score between the OCR text and the question-image feature pair.

First, the image visual features most relevant to the question text features are found through the multi-head attention network 0. The question feature embedding $F_q$ is input as the query, and the image visual feature embedding $F_v$ is used as the key. The similarity is calculated through the soft-

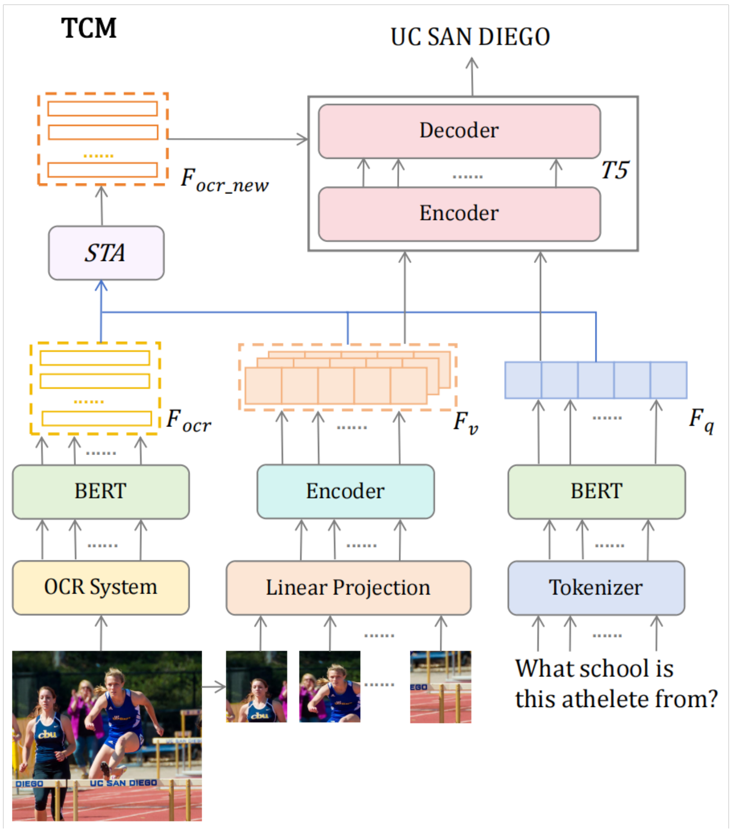

Figure 2: TCM

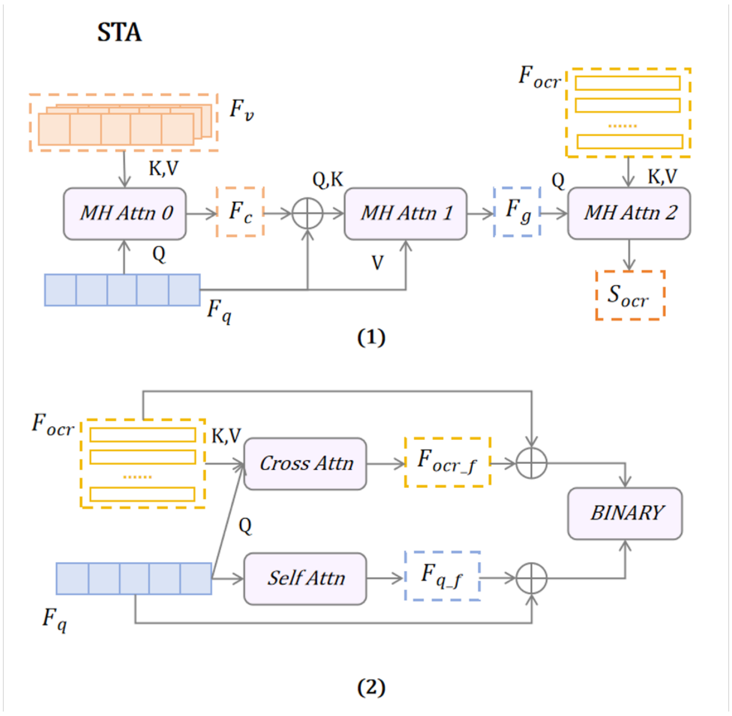

Figure 3: STA

max function, and the $F_v$ as the value is weighted according to the similarity, thereby obtaining the relevant visual features $F_c$. The calculation formula is shown in formula 5.

$$Q_0 = W_{Q0} \cdot F_q \tag{2}$$

$$K_0 = W_{K0} \cdot F_v \tag{3}$$

$$V_0 = W_{V0} \cdot F_v \tag{4}$$

$$F_c = softmax(\frac{Q_0 \cdot {K_0}^T}{\sqrt{d_k}})V_0 \tag{5}$$

Where $W_{Q0}$, $W_{K0}$, $W_{V0}$are parameter matrices that map the corresponding query, key, and value into low-dimensional vectors.

Next, the image-guided question feature is found through the multi-head attention network 1. The $F_q$ and $F_c$ obtained by connecting $F_{q\_c}$ are used as query and key inputs, and $F_q$ is used as value to obtain the image-guided question feature $F_g$, and the calculation formula is shown in formula 10.

$$F_{q\_c} \ = \ F_q \oplus F_c \tag{6}$$

$$Q_1 = W_{Q1} \cdot F_{q\_c} \tag{7}$$

$$K_1 = W_{K1} \cdot F_{q\_c} \tag{8}$$

$$V_1 = W_{V1} \cdot F_q \tag{9}$$

$$F_g = softmax(\frac{Q_1 \cdot {K_1}^T}{\sqrt{d_k}})V_1 \tag{10}$$

Where $W_{Q1}$, $W_{K1}$, $W_{V1}$are parameter matrices that map the corresponding query, key, and value into low-dimensional vectors.

Finally, the similarity score between the OCR text and the question-answer pair is found through the multi-head attention network 2. $F_g$is used as the query and $F_{ocr}$as the key. The similarity score can be obtained by the softmax function, and the calculation formula is shown in formula 13.

$$Q_2 = W_{Q2} \cdot F_g \tag{11}$$

$$K_2 = W_{K2} \cdot F_{ocr} \tag{12}$$

$$S_{ocr} = softmax(\frac{Q_2 \cdot {K_2}^T}{\sqrt{d_k}}) \tag{13}$$

Where $W_{Q2}$, $W_{K2}$are parameter matrices that map the corresponding query and key into low-dimensional vectors. Then, the scores of M OCR text units belonging to the same word are averaged to obtain a series of text word scores, and the word with the highest score is the most relevant.

It should be noted that some answer words may not be recognized by OCR, that is, they do not appear in the initial OCR text list. At this time, if the "most relevant text words" are introduced additionally, it may cause interference. Therefore, this paper adds a screening module to STA to analyze the correlation between the entire OCR text list and the question. The schematic diagram is shown in (2) in Figure 3. $F_{ocr}$ and $F_q$ are cross-attention calculated to obtain the OCR text feature $F_{ocr\_f}$ related to the question. The calculation formula is shown in formula 17.

$$Q_3 = W_{Q3} \cdot F_q \tag{14}$$

$$K_3 = W_{K3} \cdot F_{ocr} \tag{15}$$

$$V_3 = W_{V3} \cdot F_{ocr} \tag{16}$$

$$F_{ocr\_f} = softmax(\frac{Q_3 \cdot {K_3}^T}{\sqrt{d_k}})V_3 \tag{17}$$

Calculate $F_q$ through self-attention to obtain enhanced question features $F_{q\_f}$. The calculation formula is shown in 21.

$$Q_4 = W_{Q4} \cdot F_q \tag{18}$$

$$K_4 = W_{K4} \cdot F_q \tag{19}$$

$$V_4 = W_{V4} \cdot F_q \tag{20}$$

$$F_{q\_f} = softmax(\frac{Q_4 \cdot {K_4}^T}{\sqrt{d_k}})V_4 \tag{21}$$

Then connect the two with the original corresponding features and input them into the binary classifier Binary together, setting the threshold to $\tau \ = \ 0.5$. If the classification value is greater than $\tau$ , it means that the most relevant word should be added to the OCR list once more, otherwise no operation is required. In this way, a new OCR text embedding $F_{ocr\_new}$ can be obtained.

### 6.2 Working implementation

During the training process, Adam is used as the optimizer. The loss function uses the cross entropy loss function. The batch size used in the training process is 2, and the batch size used in the test is 8. The experiment uses the accuracy on the test set as the evaluation indicator to determine whether the answer predicted by the model is in the answer list set in the original dataset, thereby obtaining the accuracy.

The file "pre_select/models/cross_attn.py" is the main implementation of my approach STA. The

Table 3: Experiment Results on TextVQA

| | OCR | STA(1) | STA(2) | Accuracy(%) |
|---|---|---|---|---|
| (a) | Rosetta | × | × | 42.57 |
| (b) | Rosetta | ✓ | × | 44.25 |
| (c) | Rosetta | ✓ | ✓ | 45.13 |

Table 4: Experiment Results on ST-VQA

| | OCR | STA(1) | STA(2) | Accuracy(%) |
|---|---|---|---|---|
| (a) | Rosetta | × | × | 42.06 |
| (b) | Rosetta | ✓ | × | 43.82 |
| (c) | Rosetta | ✓ | ✓ | 43.19 |

file ”pre_select/engine.py” is the feature extractor, answer generator, train and evaluate process. The file ”dataset1/dataset.py” is the data processing and input file. Other files provide auxiliary classes and models and other functions.

### 6.3 Compute

The experiment was conducted on NVIDIA RTX A6000, which is the server in the laboratory I work in. The Python version is 3.7.16. The process of connecting to the server and configuring the environment is rather cumbersome. In the end, I used the VS code remote ssh tool for remote connection and used conda to create a virtual environment. I run the model for about 6 hours to get a stable output.

### 6.4 Results

The experimental results of the proposed model on two datasets are shown in the following table 3 and 4. STA(1) represents the first part of the scene text enhancer, which uses multiple multi-head attention mechanism modules to select the OCR text most relevant to the question-image feature pair; STA(2) represents the second part of the scene text enhancer, which uses a binary classifier to determine whether the calculated ”most relevant text” should be added to the OCR list. The first group (a) is the baseline model, which only uses the text extracted from the OCR source Rosetta-en without adding other optimizations. It is used to explore the performance without text utilization related optimization as a control. In the second group, STA(1) is added to the architecture to test the impact of selecting key text on question answering performance. In the third group, STA(2) is added to test the impact of adding a filtering module to remove some interference on the effectiveness of the STA module and the impact on question answering performance.

Table 3 reports the results on TextVQA. (a) corresponds to the result without STA, with an accuracy of 42.57%. In row (b), key text selection is added to help the answer generation module T5 focus more on the text related to the question answer. The accuracy is improved by +1.68%. In row (c), the performance is further improved by adding a binary classification module. The accuracy reaches 45.13%, which is +2.56% higher than (a) and +0.88% higher than (b), proving that it can play a role in reducing interference and proves the effectiveness of each module in this model. The experimental results on the ST-VQA dataset also verify the effectiveness of my method.

## 7 Error analysis

As shown in Figure 4 (a-b), in addition to the keywords of the correct answer, the OCR system will output some redundant scene text, which will confuse the question answering process and affect the accuracy of the final answer prediction. Taking Figure 4 (b) as an example, Rosetta+T5 processes all the obtained scene texts indiscriminately, among which ”CERVESA” interferes with the answer reasoning process, resulting in an error in answer prediction. In contrast, the STA module proposed in this paper can predict the correlation between the scene text and the question-image before using T5 to generate the answer, and add the key text ”moritz” once more to the OCR text list, so that the question answering module pays more attention to the key text and obtains an accurate answer.

As shown in Figure 4 (a), in addition to the keywords of the correct answer, Rosetta+T5 also adds text that is not related to the question answer to the predicted answer (i.e., the identified ”FAVORITE”), resulting in an incorrect answer prediction. Therefore, even if keywords are used as the source of answers in the question answering process, these redundant scene texts will still affect the accuracy of question answers. The STA module in this paper strengthens the role of key text by adding more keywords, thereby reducing the interference of redundant text on the answer generation module.

As shown in Figure 4 (b), Rosetta+T5 only pre-

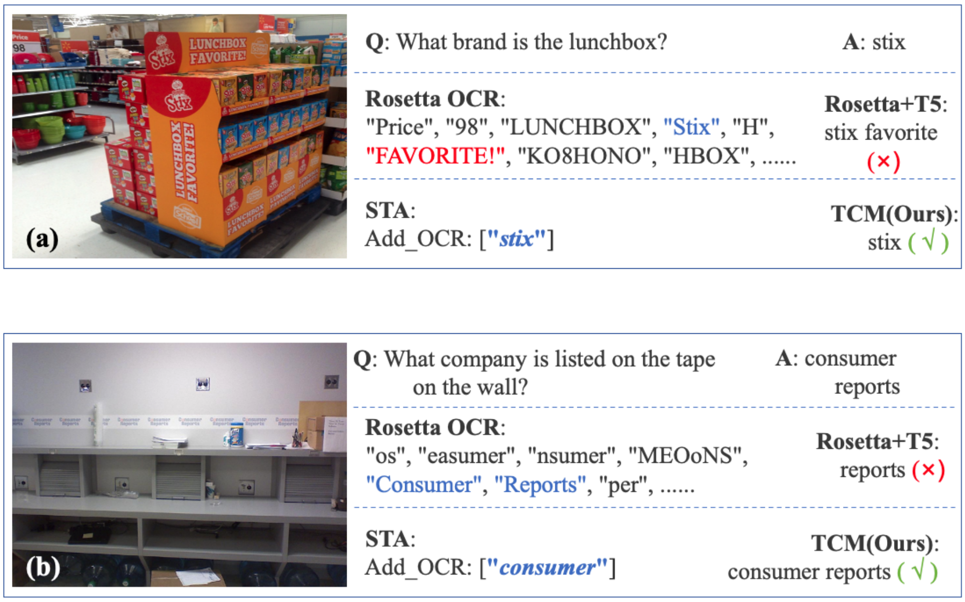
(a)
Q: What brand is the lunchbox?
A: stix
Rosetta OCR:
"Price", "98", "LUNCHBOX", "Stix", "H", "FAVORITE!", "KO8HONO", "HBOX", ......
Rosetta+T5:
stix favorite (×)
STA:
Add_OCR: ["stix"]
TCM(Ours):
stix ( √ )
(b)
Q: What company is listed on the tape on the wall?
A: consumer reports
Rosetta OCR:
"os", "easumer", "nsumer", "MEOoNS", "Consumer", "Reports", "per", ......
Rosetta+T5:
reports (×)
STA:
Add_OCR: ["consumer"]
TCM(Ours):
consumer reports ( √ )

Figure 4: Answer prediction samples on TextVQA dataset

dicts the ”reports” part of the correct answer ”consumer reports”, indicating that the text information is not fully utilized and the lack of key text information leads to incorrect answer prediction. The STA module in this paper selects the key text ”consumer” after predicting the text relevance and adds it to the OCR list, ”prompting” the T5 model to pay more attention to this key text in the process of multimodal fusion and answer prediction, thereby obtaining the correct answer.

## 8 Conclusion

In order to solve the problem that the TextVQA task model does not fully utilize the text information in the image, I propose a scene text augmenter, which is a module that determines the relevance of scene text and image-question. It can identify the most critical information from a large amount of scene text, thereby facilitating the subsequent answer reasoning process. The scene text augmenter module consists of multiple multi-head attention mechanisms. First, it finds the visual features most relevant to the question features, then finds the question features guided by the image, and finally obtains the similarity score between the OCR text features and the question-image features. The word with the highest score is the text with the strongest relevance. In addition, considering that the answer word may not be recognized by the OCR system, this paper also adds an additional filtering module.

This method was trained and tested on the TextVQA and ST-VQA datasets. The experimental results show that using the scene text augmenter for text relevance prediction can enhance the model’s attention to key text, thereby improving the accuracy of answer prediction.

There is still content worth further exploration. Text recognition can be further optimized. Currently, text recognition is completely dependent on the OCR system. If the OCR recognition is wrong or missed, it will affect the subsequent process. It is possible to consider adjusting the text recognition part to a trainable module, and reversely guide the optimization of the recognition part based on the accuracy of answer generation.

The use of text in images can be further optimized. The current use of image text does not consider the semantic relationship between texts, but simply outputs all recognized texts in a unified manner. However, the text in the image contains certain semantic associations, and this connection can be used to assist the answer generation process. In subsequent research, it is possible to consider helping the model recognize whether words have semantic contextual relationships.

## 9 Acknowledgements

I did not use AI tool to help me write the report.

## References

Biten, A. F., Litman, R., Xie, Y., Appalaraju, S., and Manmatha, R. (2022). Latr: Layout-aware transformer for scene-text vqa. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16548–16558.

Biten, A. F., Tito, R., Mafla, A., Gomez, L., Rusinol, M., Valveny, E., Jawahar, C., and Karatzas, D. (2019). Scene text visual question answering. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4291–4301.

Devlin, J. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dosovitskiy, A. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Hu, R., Singh, A., Darrell, T., and Rohrbach, M. (2020). Iterative answer prediction with pointer-augmented multimodal transformers for textvqa. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9992–10002.

Kil, J., Changpinyo, S., Chen, X., Hu, H., Goodman, S., Chao, W.-L., and Soricut, R. (2023). Prestu: Pre-training for scene-text understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15270–15280.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Singh, A., Natarajan, V., Shah, M., Jiang, Y., Chen, X., Batra, D., Parikh, D., and Rohrbach, M. (2019). Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326.

Yang, Z., Lu, Y., Wang, J., Yin, X., Florencio, D., Wang, L., Zhang, C., Zhang, L., and Luo, J. (2021). Tap: Text-aware pre-training for text-vqa and text-caption. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8751–8761.

Zhu, Q., Gao, C., Wang, P., and Wu, Q. (2021). Simple is not easy: A simple strong baseline for textvqa and textcaps. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3608–3615.