TSQA: Tabular Scenario Based Question Answering

Xiao Li, Yawei Sun, Gong Cheng
State Key Laboratory for Novel Software Technology, Nanjing University, China
{xiaoli.nju, ywsun}@smail.nju.edu.cn, gcheng@nju.edu.cn

Abstract

Scenario-based question answering (SQA) has attracted an increasing research interest. Compared with the well-studied machine reading comprehension (MRC), SQA is a more challenging task: a scenario may contain not only a textual passage to read but also structured data like tables, i.e., tabular scenario based question answering (TSQA). AI applications of TSQA such as answering multiple-choice questions in high-school exams require synthesizing data in multiple cells and combining tables with texts and domain knowledge to infer answers. To support the study of this task, we construct GeoTSQA. This dataset contains 1k real questions contextualized by tabular scenarios in the geography domain. To solve the task, we extend state-of-the-art MRC methods with TTGen, a novel table-to-text generator. It generates sentences from variously synthesized tabular data and feeds the downstream MRC method with the most useful sentences. Its sentence ranking model fuses the information in the scenario, question, and domain knowledge. Our approach outperforms a variety of strong baseline methods on GeoTSQA.

1 Introduction

Scenario-based question answering (SQA) is to answer questions contextualized by scenarios (Lally et al. 2017). Compared with the well-studied task of machine reading comprehension (MRC) which requires reading a passage to extract or infer an answer (Rajpurkar et al. 2016; Lai et al. 2017), a SQA task requires reading a scenario which commonly contains both a textual passage and a set of structured data. One such prominent AI application of SQA is answering multiple-choice questions in high-school geography exams (Ding et al. 2018; Huang et al. 2019). Those questions are contextualized by scenarios containing tables and diagrams, where the rich information cannot be captured by current MRC methods but have to be manually interpreted using natural language. Thus, one natural research question arises: can we solve SQA in a fully automated manner?

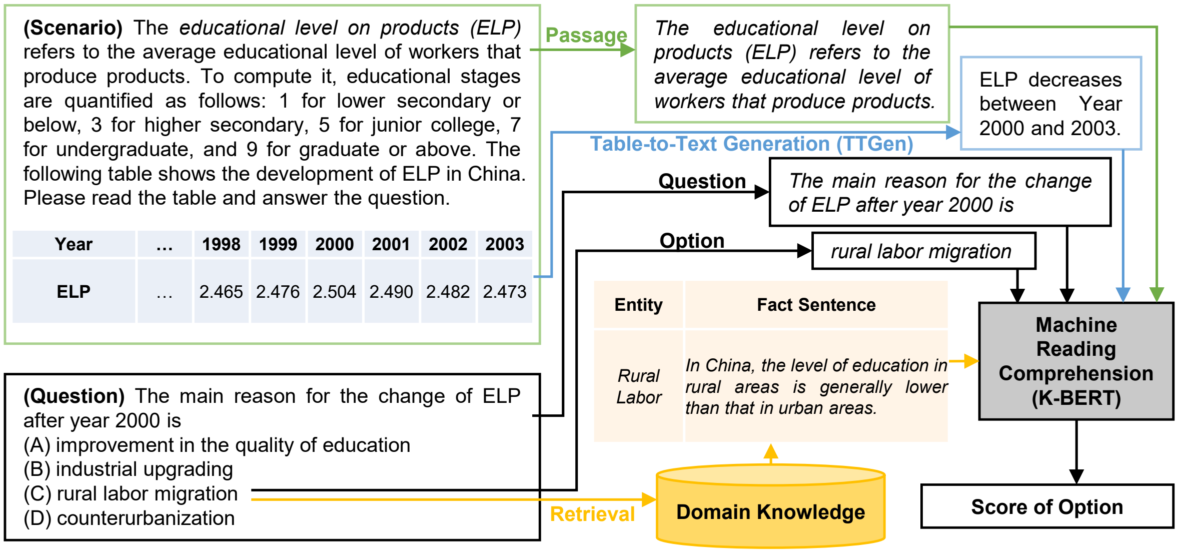
Task and Challenges. Specifically, we focus on questions contextualized by a scenario consisting of a textual passage and a set of tables. We refer to this branch of SQA as **TSQA**, short for *Tabular Scenario based Question Answering*. To support the study of this task, we construct a dataset named **GeoTSQA**. It contains 1k real questions contextualized by tabular scenarios in the geography domain, collected from China's high-school exams. Compared with existing datasets for table-based question answering like WikiTableQuestions (Pasupat and Liang 2015), GeoTSQA requires fundamentally different reading and reasoning skills, and poses new research challenges.

For instance, Figure 1 shows a question in GeoTSQA. To answer it, tabular data needs to be synthesized via a complex operation: identifying a monotonic increase in ELP over the interval 2000–2003. Focusing on this particular interval rather than many other intervals is implicitly suggested in the question: after year 2000. Moreover, the passage in the scenario helps to link ELP with educational level, and the retrieved domain knowledge bridges the gap between educational level and rural labor which is the correct answer. To conclude, TSQA methods need to *properly manipulate tabular data*, and *comprehend fused textual information*.

Our Approach. To meet the challenges, considering that text reading has been extensively studied in MRC research, we propose to extend state-of-the-art MRC methods with a novel table-to-text generator named **TTGen** to specifically handle tabular data. The basic idea is straightforward: feeding a MRC model with sentences generated from tables *using templates that encapsulate many and various predefined operations for manipulating tabular data*. However, the potentially large number (e.g., hundreds) of generated sentences may easily exceed the capacity of typical MRC models, and produce much noise information influencing the accuracy of reading comprehension. To address this problem, TTGen incorporates a sentence ranking model that fuses the information in the scenario, question, and domain knowledge to effectively *select sentences that are most useful for answering the question*. It outperforms a variety of strong baseline methods in extensive experiments on GeoTSQA.

We summarize our contributions in the paper as follows.

- We construct and publish GeoTSQA, the first dataset dedicated to TSQA. It requires reading and reasoning with tables, texts, and domain knowledge at high school level.
- We extend MRC methods with TTGen to solve TSQA. TTGen performs question and knowledge aware ranking of sentences generated from synthesized tabular data.


(Scenario) The educational level on products (ELP) refers to the average educational level of workers that produce products. To compute it, educational stages are quantified as follows: 1 for lower secondary or below, 3 for higher secondary, 5 for junior college, 7 for undergraduate, and 9 for graduate or above. The following table shows the development of ELP in China. Please read the table and answer the question.
Passage
The educational level on products (ELP) refers to the average educational level of workers that produce products.
ELP decreases between Year 2000 and 2003.
Table-to-Text Generation (TTGen)
Question
The main reason for the change of ELP after year 2000 is
Option
rural labor migration
Entity
Fact Sentence
Rural Labor
In China, the level of education in rural areas is generally lower than that in urban areas.
Machine Reading Comprehension (K-BERT)
(Question) The main reason for the change of ELP after year 2000 is
(A) improvement in the quality of education
(B) industrial upgrading
(C) rural labor migration
(D) counterurbanization
Retrieval
Domain Knowledge
Score of Option

Year	…	1998	1999	2000	2001	2002	2003
ELP	…	2.465	2.476	2.504	2.490	2.482	2.473

Figure 1: Left: an example question contextualized by a tabular scenario in GeoTSQA. Right: an overview of our approach.

Outline. The remainder of the paper is organized as follows. We discuss and compare with related work in Section 2. We formally define the TSQA task and describe the construction of the GeoTSQA dataset in Section 3. We introduce our approach in Section 4. We present experiment settings in Section 5 and report experiment results in Section 6. Finally we conclude the paper in Section 7.

Our code and data are available on GitHub.[1]

2 Related Work

2.1 SQA

SQA is an emerging AI task and has found application in many domains. The pioneering WatsonPaths system provides recommendations for diagnosis and treatment based on a medical scenario about a patient (Lally et al. 2017). In the legal domain, SQA supports judgment prediction based on the fact description of a legal case (Ye et al. 2018; Zhong et al. 2018; Yang et al. 2019b).

We focus on TSQA where a scenario contains both textual and tabular data. Such questions are common in, for example, China's high-school geography and history exams where a scenario describes a concrete fact or event to contextualize a set of questions. Previous efforts in this domain either ignore tables (Cheng et al. 2016; Zhang et al. 2018) or manually transform tables into triple-structured knowledge (Ding et al. 2018) or natural language descriptions for machine reading (Huang et al. 2019). In contrast, we aim at *solving TSQA in a fully automated manner by generating texts from tables*.

2.2 Table-to-Text Generation

Table-to-text generation has been studied for decades. Early methods rely on handcrafted rules to generate texts for specific domains such as stock market summaries (Kukich 1983) and weather forecasts (Goldberg, Driedger, and Kittredge 1994). They typically implement a pipeline of modules including content planning, sentence planning, and surface realization. Today, it is feasible to train neural generation models in an end-to-end fashion, thanks to the availability of effective pre-trained language models (Devlin et al. 2019; Radford et al. 2019) and large datasets (Lebret, Grangier, and Auli 2016; Wiseman, Shieber, and Rush 2017; Dusek, Novikova, and Rieser 2019). Current models often adopt an encoder-decoder architecture with a copy mechanism (Wiseman, Shieber, and Rush 2017; Puduppully, Dong, and Lapata 2019a). Moreover, they can be enhanced with entity representations (Puduppully, Dong, and Lapata 2019b) and external background knowledge (Chen et al. 2019).

The above methods are targeted on surface-level description of tabular data, which is insufficient for our task where data in multiple cells needs to be *synthesized using various operations* (e.g., extremum, monotonicity, trend). Generating such natural language statements that are logically entailed from tabular data, rather than superficial restatements, has recently attracted research attention (Chen et al. 2020a,d). However, they are primarily focused on high-fidelity generation, i.e., the generated text should be faithful to the tabular data. Fidelity is necessary but insufficient for our task where the generated text also needs to be useful for answering the question. It is thus essential to *select the proper operation and data from a potentially very large space*. To this end, our proposed generator TTGen features a sentence ranking model that fuses the information in the scenario, question, and domain knowledge.

2.3 Table-Based Question Answering

Similar to TSQA, there has been a line of research of answering questions over tabular data (Pasupat and Liang 2015; Jauhar, Turney, and Hovy 2016; Yin et al. 2016; Yu

[1] https://github.com/nju-websoft/TSQA

et al. 2020). Like our constructed dataset GeoTSQA, these datasets also require performing various operations over multiple cells. Differently, their questions can be answered solely on the basis of tabular data, whereas the questions in GeoTSQA are more naturally contextualized by a scenario containing *both* a set of tables and a textual passage which are equally important and are *dependent on each other*.

From this angle, the most similar dataset to GeoTSQA is HybridQA (Chen et al. 2020c), where table cells are linked with Wikipedia pages. However, GeoTSQA has its *unique challenges* due to the source of questions—high-school geography exams. For example, table cells mainly contain non-linkable numeric values; more complex operations (e.g., monotonicity) are needed; it would be helpful to incorporate domain knowledge into question answering.

3 Task and Dataset

We firstly define the task of TSQA, and then we construct the GeoTSQA dataset to support the study of TSQA.

3.1 Task Definition

A TSQA task consists of a scenario $\langle P, T \rangle$, a question Q, and a set of options O as candidate answers of which only one is correct. The scenario contains a passage P and a set of tables T. Each table in T has a header row, a header column, and a set of content cells. The goal is to select an option from O as the answer to Q contextualized by $\langle P, T \rangle$.

3.2 Dataset Construction

We constructed GeoTSQA. To the best of our knowledge, it is the first dataset dedicated to the TSQA task.

Collecting Questions. We collected multiple-choice questions contextualized by tabular scenarios in the geography domain from China's high-school exams. A related dataset is GeoSQA (Huang et al. 2019). We not only collected all the questions from GeoSQA but also reused the code for constructing GeoSQA to crawl much more questions from the Web to expand our dataset.

However, many collected scenarios are not tabular. Indeed, each scenario is associated with a set of image files. Each image file depicts either a table or another kind of diagram such as a map or a histogram. Therefore, we need to identify images depicting tables or table-like diagrams.

Identifying Tables. We looked for tables, or charts that can be straightforwardly converted to tables (e.g., histograms, line charts). We manually identified 200 such image files as positive examples and another 200 image files as negative examples. We used them to train an image classifier (Szegedy et al. 2016) to classify all the remaining image files. Finally, for all the image files that were classified as positive, we manually checked them for classification errors.

Extracting Tables. We recruited 15 undergraduate students from a university in China as annotators. For image files depicting tables, we used Baidu's OCR tool to extract tabular data. OCR errors were manually corrected by annotators. For image files depicting charts, annotators manually extracted tabular data, assisted with a tool we developed. The annotator used that tool to easily click key points in the image, e.g., the origin, coordinate axes, data points. The tool then automatically converted data points to data tables.

Scenarios	556	
Chinese characters per passage	52.42	±32.99
Tables per scenario	1.58	±0.93
Cells per table	26.98	±17.51
Questions	1,012	
Chinese characters per question	44.02	±15.89

Table 1: Statistics about GeoTSQA.

Annotators manually checked each extracted table and filtered out irregular tables (e.g., with multi-level headers).

Filtering Questions. Last but not least, annotators filtered out questions that can be answered without using any table. Therefore, every question in GeoTSQA is contextualized by a tabular scenario, and it is essential to employ the information in the given tables to answer the question.

3.3 Dataset Statistics

GeoTSQA contains 556 scenarios and 1,012 multiple-choice questions. Each question has four options. More statistics about the dataset are shown in Table 1.

Out of the 878 tables in GeoTSQA, 96% only contain numeric content cells. It differs from HybridQA (Chen et al. 2020c) where content cells are often entities linked with Wikipedia pages, thereby providing extra background knowledge for answering questions. For GeoTSQA, to obtain information that is not explicitly given in the scenario but critical for answering questions, it is essential to entail from tabular data via operations over multiple cells.

4 Approach

We propose a two-step approach to solve TSQA. As illustrated in Figure 1, the first step (Section 4.2) is a table-to-text generator named TTGen. From the tables T in a scenario $\langle P, T \rangle$, TTGen generates top-k sentences S that are most useful for answering the question Q. The second step (Section 4.1) is a MRC method based on K-BERT (Liu et al. 2020), a state-of-the-art knowledge-enabled language model. It fuses the information in the passage P, generated sentences S, question Q, and domain knowledge K to rank the options in O.

4.1 MRC with Domain Knowledge

Our MRC method is based on K-BERT (Liu et al. 2020). This state-of-the-art language model extends BERT (Devlin et al. 2019) with the capability to utilize external knowledge such as domain knowledge.

MRC with K-BERT. For each option $o_i \in O$, we concatenate the passage P, top-k sentences $S = \{s_1, \dots, s_k\}$ generated from the tables T, question Q, and o_i in a standard way, starting with a [CLS] token and separating with [SEP]:

$$I_i^{\text{MRC}} = \text{[CLS]}\ P\ s_1 \cdots s_k\ Q\ \text{[SEP]}\ o_i\ \text{[SEP]}\ \mathit{NUMS}_i\ \text{[SEP]}\,, \tag{1}$$

where *NUMS*$_i$ is a concatenation of all the numeric tokens in P, S, Q, and o_i. Each numeric token in the original position is replaced by a special token [NUM].

We use K-BERT to obtain a vector representation for each token in I_i^{MRC} to capture its semantic features:

$$\langle \mathbf{h}_{i1}^{\text{MRC}}, \mathbf{h}_{i2}^{\text{MRC}}, \ldots \rangle = \text{K-BERT}(I_i^{\text{MRC}},\ K)\,, \tag{2}$$

where K is an external knowledge base we will explain later.

The vector representation for the [CLS] token, i.e., $\mathbf{h}_{i1}^{\text{MRC}}$, is used as an aggregate representation for I_i^{MRC}. It is fed into two dense layers followed by a softmax layer to obtain a correctness score $\hat{\omega}_i$ for each option $o_i \in O$:

$$\begin{aligned} \omega_i &= \mathbf{w}_2^{\intercal} \tanh(\mathbf{W}_1 \mathbf{h}_{i1}^{\text{MRC}} + \mathbf{b}_1) + b_2\,, \\ \mathbf{\Omega} &= [\hat{\omega}_1; \hat{\omega}_2; \ldots] = \texttt{softmax}([\omega_1; \omega_2; \ldots])\,, \end{aligned} \tag{3}$$

where $\mathbf{W}_1$ is a trainable matrix, $\mathbf{w}_2$ and $\mathbf{b}_1$ are trainable vectors, and b_2 is a trainable parameter.

In the training phase, we minimize the negative log-likelihood loss which measures the difference between $\mathbf{\Omega}$ and the binary correctness label on each option (we will detail in Section 5.1). In the test phase, we choose the option in O with the highest correctness score $\hat{\omega}$ as the answer.

K-BERT extends BERT with an external knowledge base K. It helps to fuse the information in P, S, Q, O, and K. We refer the reader to Liu et al. (2020) for a detailed description of K-BERT. Briefly, each entry in K is a pair $\langle$entity, fact sentence$\rangle$, or a triple $\langle$entity, property, value$\rangle$ which can be converted into a pair by concatenating the property and the value into a fact sentence. K-BERT employs K to expand the input sequence into a tree of tokens: fact sentences about an entity are retrieved from K and inserted as branches after each mention of the entity in the input sequence. In our implementation, for each entity, we retrieve top-ϵ fact sentences that are most relevant to the input sequence. The relevance of a fact sentence to the input sequence is measured by the cosine similarity between their average pre-trained BERT embedding vectors.

Domain Knowledge. For the external knowledge base K, for our experiments we use domain knowledge since all the questions in GeoTSQA are in the geography domain. We obtain domain knowledge from two sources.

First, we import all the triples in Clinga (Hu et al. 2016), a large Chinese geographical knowledge base.

Second, we reuse the corpus in (Huang et al. 2019). The corpus contains a geography textbook providing a set of entity descriptions. We pair each entity with each sentence in its description as a fact sentence. The corpus also contains a subset of Chinese Wikipedia. We treat the title of each page as an entity and pair it with each sentence in the page as a fact sentence.

4.2 Table-to-Text Generation (TTGen)

Below we describe the generation of sentences from tables to be fed into our MRC method. We rely on templates that encapsulate predefined operations for manipulating tabular data. It enables us to perform complex operations that are needed for answering hard questions such as those in GeoTSQA. We generate sentences from tables using all the applicable templates. However, it is infeasible for a MRC model like K-BERT to jointly encode a large number (e.g., hundreds) of sentences. Therefore, we rank the generated sentences and select k top-ranked sentences that are most useful for answering the question. By filtering the generated sentences, we can also reduce noise information that may influence the accuracy of reading comprehension.

Sentence Generation. By significantly extending the operations considered in Chen et al. (2020a,b), we define six table-to-text templates that encapsulate different powerful operations for synthesizing numeric tabular data. As we will show in the experiments, these templates have covered most needs about tables in GeoTSQA. One can easily add new templates to accommodate other applications.

- **Extremum.** This template reports the maximum or minimum value of a row or column. An example sentence generated from the table in Figure 1 is: *ELP reaches a maximum of 2.504 at Year 2000.*
- **Special values.** This template reports or compares with a special value (e.g., under a column header that is mentioned in the question), e.g., *ELP at Year 2000 is 2.504.*
- **Comparison with average.** This template reports a maximal sequence of cells where all the values are above or below the average of the entire row or column, e.g., *ELP is relatively large between Year 2000 and 2002.*
- **Monotonicity.** This template reports a monotonic increase or decrease over a maximal sequence of cells, e.g., *ELP decreases between Year 2000 and 2003.*
- **Trend.** This template reports the overall trend of a row or column, e.g., *ELP generally increases and then decreases.*
- **Range comparison.** This template reports a comparison between two maximal corresponding sequences of cells from different rows or columns.

For non-numeric tabular data, we simply concatenate each row header, each column header, and the corresponding content cell into a sentence.

Sentence Ranking. Let $\hat{S}$ be the set of sentences generated from the tables T using all the applicable templates. We compute a usefulness score for each sentence $s_j \in \hat{S}$, and choose k top-ranked sentences $S \subseteq \hat{S}$. To select sentences that are most useful for answering the question, our ranking model employs K-BERT to fuse the information in the passage P, question Q, and domain knowledge K to perform question and knowledge aware ranking. Figure 2 presents an overview of the model. It integrates two complementary rankers: sentence-level ranking directly assesses the usefulness of each individual sentence; template-level ranking infers useful templates purely from the passage and question.

For sentence-level ranking, we concatenate the passage P, question Q, and sentence s_j in a standard way:

$$I_j^{\text{SR}} = \text{[CLS]}\ P\ Q\ \text{[SEP]}\ s_j\ \text{[SEP]}\ \textit{NUMS}_j\ \text{[SEP]}\,, \tag{4}$$

where *NUMS*$_j$ is a concatenation of all the numeric tokens in P, Q, and s_j. Each numeric token in the original position

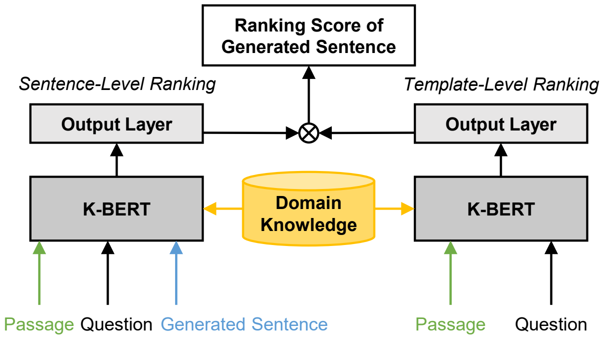


Figure 2: Sentence ranking model in TTGen.

is replaced by a special token [NUM]. We use K-BERT to obtain a vector representation for each token in I^{SR}_j:

$$\langle \mathbf{h}^{\text{SR}}_{j1}, \mathbf{h}^{\text{SR}}_{j2}, \ldots \rangle = \text{K-BERT}(I^{\text{SR}}_j,\ K)\,. \tag{5}$$

The vector representation for the [CLS] token, i.e., $\mathbf{h}^{\text{SR}}_{j1}$, is fed into two dense layers followed by a softmax layer to obtain a usefulness score $\hat{\phi}_j$ for each sentence $s_j \in \hat{S}$:

$$\begin{aligned} \phi_j &= \mathbf{w}_4^\intercal \tanh(\mathbf{W}_3 \mathbf{h}^{\text{SR}}_{j1} + \mathbf{b}_3) + b_4\,, \\ \boldsymbol{\Phi} &= [\hat{\phi}_1; \hat{\phi}_2; \ldots] = \texttt{softmax}([\phi_1; \phi_2; \ldots])\,, \end{aligned} \tag{6}$$

where $\mathbf{W}_3$ is a trainable matrix, $\mathbf{w}_4$ and $\mathbf{b}_3$ are trainable vectors, and b_4 is a trainable parameter. In the training phase, we minimize the negative log-likelihood loss which measures the difference between $\boldsymbol{\Phi}$ and the binary usefulness label on each generated sentence (we will detail in Section 5.1).

For template-level ranking, we concatenate the passage P and question Q in a standard way:

$$I^{\text{TR}} = \text{[CLS]}\ P\ Q\ \text{[SEP]}\,. \tag{7}$$

We use K-BERT to obtain a vector representation for each token in I^{TR}:

$$\langle \mathbf{h}^{\text{TR}}_{1}, \mathbf{h}^{\text{TR}}_{2}, \ldots \rangle = \text{K-BERT}(I^{\text{TR}},\ K)\,. \tag{8}$$

The vector representation for the [CLS] token, i.e., $\mathbf{h}^{\text{TR}}_1$, is fed into two dense layers followed by a sigmoid layer to obtain a usefulness score $\hat{\psi}$ for each of the six templates:

$$\begin{aligned} [\psi_1; \ldots; \psi_6] &= \mathbf{W}_6 \tanh(\mathbf{W}_5 \mathbf{h}^{\text{TR}}_1 + \mathbf{b}_5) + \mathbf{b}_6\,, \\ \boldsymbol{\Psi} &= [\hat{\psi}_1; \ldots; \hat{\psi}_6] = \texttt{sigmoid}([\psi_1; \ldots; \psi_6])\,, \end{aligned} \tag{9}$$

where $\mathbf{W}_5$ and $\mathbf{W}_6$ are trainable matrices, $\mathbf{b}_5$ and $\mathbf{b}_6$ are trainable vectors. Let sentence s_j be generated by the τ_j-th template. We derive usefulness labels on templates for training from usefulness labels on generated sentences: a template is labeled useful if and only if at least one sentence it generates is labeled useful. Multiple sentences and hence multiple templates may be labeled useful for answering a question. Therefore, in the training phase, we formulate a multi-label binary classification task, and we minimize the binary cross-entropy loss which measures the difference between $\boldsymbol{\Psi}$ and the binary usefulness label on each template.

Finally, in the test phase, we compute:

$$\text{usefulness score of } s_j = \hat{\phi}_j \cdot \hat{\psi}_{\tau_j}\,. \tag{10}$$

Output of linearization for the table in Figure 1:

... ELP at Year 1998 is 2.465. ELP at Year 1999 is 2.476. ELP at Year 2000 is 2.504. ELP at Year 2001 is 2.490. ELP at Year 2002 is 2.482. ELP at Year 2003 is 2.473.

Table 2: Example output of Linearization.

5 Experiment Setup

We compared our approach with a variety of strong baseline methods for TSQA. We also evaluated our sentence ranking model, which is the core component of our approach.

5.1 Labeled Data

Correctness Labels on Options. For each question, from its known correct answer, we derived a label for each of the four options indicating whether it is the correct answer. These binary correctness labels were used to train and evaluate TSQA methods.

Usefulness Labels on Generated Sentences. The number of all the sentences $\hat{S}$ generated by our templates for a question is in the range of 2–176, with a mean of 41.58 and a median of 38. For each question, we asked an annotator (recruited in Section 3.2) to read $\hat{S}$ and assign a label to each sentence indicating whether it is useful for answering the question. These binary usefulness labels were used to train and evaluate sentence ranking models.

Gold-Standard Sentences. Furthermore, the annotator manually summarized the tables in one sentence describing necessary information for answering the question. This gold-standard sentence was used for comparison.

We randomly sampled 100 questions from GeoTSQA. For 92 questions, $\hat{S}$ fully covers the information in the gold-standard sentence. For 6 questions, $\hat{S}$ partially covers that information. Therefore, our six templates show good coverage of the various operations required by GeoTSQA.

5.2 Baselines

Our approach extends MRC methods. It is not our focus to compare existing MRC methods. Instead, table-to-text generation is our major technical contribution. Therefore, in the experiments we consistently used the MRC method based on K-BERT described in Section 4.1, but fed it with sentences generated from tables by the following different methods.

Supervised Methods. Firstly, we compared with three table-to-text generators that achieved state-of-the-art results on the recent LogicNLG dataset (Chen et al. 2020a) which, similar to our GeoTSQA, requires synthesizing data in multiple cells. These generators are open source. **Field-Infusing** employs LSTM to encode each table into a sequence of vectors and then applies Transformer to generate text. **GPT-Linearization** linearizes each table as a paragraph by horizontally scanning the table and concatenating each content cell with its row header and column header into a sentence. Table 2 illustrates such a paragraph. The resulting paragraph

Output of templation for the table in Figure 1:

... ELP at Year 2000 is 2.504. ... ELP decreases between Year 2000 and 2003. ... ELP generally increases and then decreases. ... ELP reaches a maximum of 2.504 at Year 2000. ... ELP is relatively large between Year 2000 and 2002. ...

Table 3: Example output of Templation.

is then fed into GPT-2 to generate a new text. **Coarse-to-Fine** is an enhanced version of GPT-Linearization. It adopts a two-step text generation process: generating a template and then filling it.

Furthermore, we implemented an enhanced version of GPT-Linearization and Coarse-to-Fine, referred to as **GPT-Linearization^{+}** and **Coarse-to-Fine^{+}**, respectively. At the beginning of the paragraph fed into GPT-2, we inserted the scenario passage and question to enable GPT-2 to perform question-aware text generation.

All the above supervised table-to-text generators were trained based on sentences with positive usefulness labels.

Unsupervised Methods. We also compared with two naive table-to-text generators.

Recall that GPT-Linearization generates a paragraph from tables and then feeds it into GPT-2 to generate a new text. We implemented **Linearization**. It directly outputs the generated paragraph without feeding it into GPT-2.

Besides, we implemented **Templation**. It generates a paragraph consisting of all the sentences $\hat{S}$ generated by our templates. Sentences are sorted in ascending order of length so that if the paragraph has to be truncated by the maximum sequence length of K-BERT, the largest number of sentences can be retained. Table 3 illustrates such a paragraph.

Gold-Standard Sentence. Last but not least, we used manually annotated gold-standard sentence as a reference.

5.3 Implementation Details

We performed 5-fold cross-validation. For each fold, we split GeoTSQA into 80% for training and 20% for test. For model selection, we relied on an inner holdout 80%/20% training/development split. We ran all the experiments on TITAN RTX GPUs.

For K-BERT, we used BERT-wwm-ext (Cui et al. 2019), a pre-trained Chinese language model as the underlying language model. We set maximum sequence length $= 256$, self-attention layer $= 12$, hidden units $= 768$, epochs $= 15$ for MRC and template-level ranking, epochs $= 5$ for sentence-level ranking, batch size $= 8$ for MRC, batch size $= 16$ for template-level ranking and sentence-level ranking, learning rate $= 1e{-}5$, and attention heads $= 12$. For knowledge base retrieval we set $\epsilon = 2$. Inspired by Jin et al. (2020), for the K-BERT model in our MRC method (but not the one in TTGen), we coarse-tuned it on C^{3} (Sun et al. 2020), a Chinese MRC dataset.

For GPT-2, we used CDialGPT2$_{\text{LCCC-base}}$ (Wang et al. 2020), a pre-trained Chinese GPT-2 model. For CDialGPT2$_{\text{LCCC-base}}$, and for LSTM and Transformer in Field-Infusing, we followed the recommended hyperparameter settings in their original implementation.

	Accuracy
Field-Infusing	0.353 $^{\bullet}$
GPT-Linearization	0.370
Coarse-to-Fine	0.367
GPT-Linearization^{+}	0.348 $^{\bullet}$
Coarse-to-Fine^{+}	0.359 $^{\circ}$
Linearization	0.235 $^{\bullet}$
Templation	0.243 $^{\bullet}$
TTGen	**0.397**
Gold-Standard Sentence	0.418

Table 4: Accuracy of TSQA. We mark the results of baselines that are significantly lower than TTGen under $p < 0.01$ ($^{\bullet}$) or $p < 0.05$ ($^{\circ}$).

For our TTGen, by default we set $k = 2$ to only select the top-2 generated sentences for MRC. We will report a comparison in different settings of k.

5.4 Evaluation Metrics

To evaluate TSQA, we measured **accuracy**, i.e., the proportion of correctly answered questions.

To evaluate sentence ranking, we measured the quality of the whole ranked list of all the sentences $\hat{S}$ generated by our templates. We used two standard information retrieval evaluation metrics: Mean Average Precision (**MAP**) and Mean Reciprocal Rank (**MRR**).

6 Experiment Results

We report average results on the test sets over all the folds.

6.1 Results on TSQA

Comparison with Baselines. Table 4 shows the accuracy of TSQA achieved by each method. Our TTGen outperforms all the baselines by 2.7–16.2 percent of accuracy.

TTGen exceeds three state-of-the-art table-to-text generators, i.e., Field-Infusing, GPT-Linearization, and Coarse-to-Fine, by 2.7–4.4 percent of accuracy.

The enhanced version of these generators that we implemented, i.e., GPT-Linearization^{+} and Coarse-to-Fine^{+}, exhibit surprisingly worse performance than their original version. Their generation methods are significantly inferior to our TTGen by 3.8–5.1 percent of accuracy.

The two naive generators, i.e., Linearization and Templation, produce much noise information for MRC and achieve accuracy even lower than random guess (i.e., 0.25). It demonstrates the necessity of ranking and selecting generated sentences.

The accuracy of using gold-standard sentence is 0.418. On the one hand, compared with the accuracy 0.397 of our TTGen, it suggests that there is still room for improving our templates and/or our sentence ranking model. On the other hand, the achieved accuracy is not satisfying. To improve the overall performance of our approach, we need to combine our TTGen with novel MRC methods that are more powerful

	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$
Accuracy	0.390	0.397	0.352	0.343	0.330

Table 5: Accuracy of TSQA by varying k in TTGen.

	Accuracy
TTGen	**0.397**
TTGen w/o tabular data	0.372
TTGen w/o domain knowledge	0.380

Table 6: Accuracy of TSQA (ablation study).

than K-BERT to meet the unique challenges raised by the GeoTSQA dataset. This will be our future work.

Varying k. Table 5 shows the accuracy of TSQA achieved by our approach under different settings of k. Increasing k from 1 to 2 (the default value), the accuracy remains stable. Further increasing k to 3 or larger, the accuracy drops substantially, probably influenced by the extra noise information. It is thus important to rank generated sentences and only select those useful for answering the question.

Ablation Study. To analyze the usefulness of tabular data and domain knowledge in TSQA, we implemented two variants of our approach. The first variant ignored tabular data. The second variant ignored domain knowledge.

Table 6 shows the accuracy of TSQA achieved by each variant. Compared with the full version of our approach, the accuracy of both variants decrease, by 2.5 percent of accuracy without tabular data and by 1.7 percent of accuracy without domain knowledge. The results reveal the usefulness of tabular data and of domain knowledge.

6.2 Results on Sentence Ranking

We compared our sentence ranking model with a strong baseline method: **RE2** (Yang et al. 2019a). This state-of-the-art text matcher is open source. We employed it to compute the semantic relevance of each generated sentence in $\hat{S}$ to the question. Specifically, we used RE2 as a text pair classifier to predict a ranking score for each generated sentence conditioned on (i.e., paired with) a concatenation of the scenario passage and question. We followed the recommended hyperparameter setting in its original implementation.

Table 7 shows the quality of sentence ranking computed by each method. Our TTGen exceeds RE2 by 5.2 percent of MAP and by 6.0 percent of MRR. Paired t-tests show that all these differences are statistically significant under $p < 0.01$.

6.3 Error Analysis

We randomly sampled 100 questions to which our approach provided incorrect answers. We analyzed the question answering process and identified the following three main causes of errors. Multiple causes could apply to a question.

Knowledge Base. For 76% of the errors, there is a lack of necessary domain or commonsense knowledge for answering the question, such as the location of a particular lake. It suggests expanding our knowledge base. However, this is orthogonal to our technical contribution.

	MAP	MRR
RE2	$0.434^{\bullet}$	$0.461^{\bullet}$
TTGen	**0.486**	**0.521**

Table 7: Quality of sentence ranking. We mark the results of baselines that are significantly lower than TTGen under $p < 0.01$ ($^{\bullet}$).

Reasoning Capabilities. For 62% of the errors, more advanced reasoning skills are needed. For example, some questions require multi-hop math calculations over a group of related domain concepts. K-BERT as a language model cannot calculate. It is also impracticable to encapsulate such extremely complex operations with predefined templates. Therefore, it suggests incorporating specific calculators and powerful reasoners into MRC models.

Sentence Ranking. For 54% of the errors, our sentence ranking model chooses a sentence that is not useful for answering the question. Indeed, some templates and their generated sentences are linguistically similar though logically different, e.g., *is relatively large*, *reaches maximum*, and *increases*. This sometimes challenges our sentence ranking model as well as our MRC method. We will focus on this problem in the future work.

7 Conclusion

Our study aims at solving TSQA in a fully automated manner to avoid manually interpreting tabular data using natural language descriptions as done in previous research. To support this study, we constructed and published the first dataset GeoTSQA that is dedicated to the TSQA task. With only six templates encapsulating predefined operations for synthesizing tabular data in various ways, we covered most needs about tables in GeoTSQA but then, the problem turned into selecting, among a large number of sentences generated from templates, the most useful ones for answering the question. Our proposed model effectively integrates sentence-level and template-level ranking, and exploits the scenario passage, question, and domain knowledge by fusing their information with K-BERT. Our approach has the potential to be adapted to other AI applications that require table comprehension and explanation.

Although our approach outperformed a variety of strong baselines in the experiments, its accuracy is still not satisfying. Following the results of our error analysis, for the future work, we plan to enhance our sentence ranking model with more powerful semantic matching techniques. We will also extend our MRC method to perform math calculation and logical reasoning over an expanded knowledge base.

Acknowledgments

This work was supported in part by the National Key R&D Program of China (2018YFB1005100) and in part by the NSFC (61772264). We thank the annotators for their efforts and thank the reviewers for their comments.

References

Chen, S.; Wang, J.; Feng, X.; Jiang, F.; Qin, B.; and Lin, C.-Y. 2019. Enhancing Neural Data-To-Text Generation Models with External Background Knowledge. 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)*, 3022–3032. Hong Kong, China: Association for Computational Linguistics.

Chen, W.; Chen, J.; Su, Y.; Chen, Z.; and Wang, W. Y. 2020a. Logical Natural Language Generation from Open-Domain Tables. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7929–7942. Online: Association for Computational Linguistics.

Chen, W.; Wang, H.; Chen, J.; Zhang, Y.; Wang, H.; Li, S.; Zhou, X.; and Wang, W. Y. 2020b. TabFact: A Large-scale Dataset for Table-based Fact Verification. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Chen, W.; Zha, H.; Chen, Z.; Xiong, W.; Wang, H.; and Wang, W. Y. 2020c. HybridQA: A Dataset of Multi-Hop Question Answering over Tabular and Textual Data. In Cohn, T.; He, Y.; and Liu, Y., eds., *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020*, 1026–1036. Association for Computational Linguistics.

Chen, Z.; Chen, W.; Zha, H.; Zhou, X.; Zhang, Y.; Sundaresan, S.; and Wang, W. Y. 2020d. Logic2Text: High-Fidelity Natural Language Generation from Logical Forms. In Cohn, T.; He, Y.; and Liu, Y., eds., *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020*, 2096–2111. Association for Computational Linguistics.

Cheng, G.; Zhu, W.; Wang, Z.; Chen, J.; and Qu, Y. 2016. Taking Up the Gaokao Challenge: An Information Retrieval Approach. In Kambhampati, S., ed., *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, 2479–2485. IJCAI/AAAI Press.

Cui, Y.; Che, W.; Liu, T.; Qin, B.; Yang, Z.; Wang, S.; and Hu, G. 2019. Pre-Training with Whole Word Masking for Chinese BERT. *arXiv preprint arXiv:1906.08101* .

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Ding, J.; Wang, Y.; Hu, W.; Shi, L.; and Qu, Y. 2018. Answering multiple-choice questions in geographical gaokao with a concept graph. In *European Semantic Web Conference*, 161–176. Springer.

Dusek, O.; Novikova, J.; and Rieser, V. 2019. Evaluating the State-of-the-Art of End-to-End Natural Language Generation: The E2E NLG Challenge. *Comput. Speech Lang.* 59: 123–156.

Goldberg, E.; Driedger, N.; and Kittredge, R. I. 1994. Using natural-language processing to produce weather forecasts. *IEEE Expert* 9(2): 45–53.

Hu, W.; Li, H.; Sun, Z.; Qian, X.; Xue, L.; Cao, E.; and Qu, Y. 2016. Clinga: Bringing Chinese Physical and Human Geography in Linked Open Data. In Groth, P.; Simperl, E.; Gray, A.; Sabou, M.; Krötzsch, M.; Lecue, F.; Flöck, F.; and Gil, Y., eds., *The Semantic Web – ISWC 2016*, 104–112. Cham: Springer International Publishing. ISBN 978-3-319-46547-0.

Huang, Z.; Shen, Y.; Li, X.; Wei, Y.; Cheng, G.; Zhou, L.; Dai, X.; and Qu, Y. 2019. GeoSQA: A Benchmark for Scenario-based Question Answering in the Geography Domain at High School Level. 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)*, 5866–5871. Hong Kong, China: Association for Computational Linguistics.

Jauhar, S. K.; Turney, P.; and Hovy, E. 2016. Tables as semi-structured knowledge for question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 474–483.

Jin, D.; Gao, S.; Kao, J.; Chung, T.; and Hakkani-Tür, D. 2020. MMM: Multi-Stage Multi-Task Learning for Multi-Choice Reading Comprehension. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, 8010–8017. AAAI Press.

Kukich, K. 1983. Design of a Knowledge-Based Report Generator. In *21st Annual Meeting of the Association for Computational Linguistics*, 145–150. Cambridge, Massachusetts, USA: Association for Computational Linguistics.

Lai, G.; Xie, Q.; Liu, H.; Yang, Y.; and Hovy, E. 2017. RACE: Large-scale ReAding Comprehension Dataset From Examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 785–794. Copenhagen, Denmark: Association for Computational Linguistics.

Lally, A.; Bagchi, S.; Barborak, M. A.; Buchanan, D. W.; Chu-Carroll, J.; Ferrucci, D. A.; Glass, M. R.; Kalyanpur, A.; Mueller, E. T.; Murdock, J. W.; et al. 2017. Watson-Paths: scenario-based question answering and inference over unstructured information. *AI Magazine* 38(2): 59–76.

Lebret, R.; Grangier, D.; and Auli, M. 2016. Neural Text Generation from Structured Data with Application to the Biography Domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Process-*

ing, 1203–1213. Austin, Texas: Association for Computational Linguistics.

Liu, W.; Zhou, P.; Zhao, Z.; Wang, Z.; Ju, Q.; Deng, H.; and Wang, P. 2020. K-BERT: Enabling Language Representation with Knowledge Graph. In *AAAI*, 2901–2908.

Pasupat, P.; and Liang, P. 2015. Compositional Semantic Parsing on Semi-Structured Tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 1470–1480. Beijing, China: Association for Computational Linguistics.

Puduppully, R.; Dong, L.; and Lapata, M. 2019a. Data-to-Text Generation with Content Selection and Planning. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, 6908–6915. AAAI Press.

Puduppully, R.; Dong, L.; and Lapata, M. 2019b. Data-to-text Generation with Entity Modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2023–2035. Florence, Italy: Association for Computational Linguistics.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language models are unsupervised multitask learners. *OpenAI Blog* 1(8): 9.

Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2383–2392. Austin, Texas: Association for Computational Linguistics.

Sun, K.; Yu, D.; Yu, D.; and Cardie, C. 2020. Investigating Prior Knowledge for Challenging Chinese Machine Reading Comprehension. *Transactions of the Association for Computational Linguistics* .

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the Inception Architecture for Computer Vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 2818–2826. IEEE Computer Society.

Wang, Y.; Ke, P.; Zheng, Y.; Huang, K.; Jiang, Y.; Zhu, X.; and Huang, M. 2020. A Large-Scale Chinese Short-Text Conversation Dataset. In *Nlpcc*.

Wiseman, S.; Shieber, S.; and Rush, A. 2017. Challenges in Data-to-Document Generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2253–2263. Copenhagen, Denmark: Association for Computational Linguistics.

Yang, R.; Zhang, J.; Gao, X.; Ji, F.; and Chen, H. 2019a. Simple and Effective Text Matching with Richer Alignment Features. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4699–4709. Florence, Italy: Association for Computational Linguistics.

Yang, W.; Jia, W.; Zhou, X.; and Luo, Y. 2019b. Legal Judgment Prediction via Multi-Perspective Bi-Feedback Network. In Kraus, S., ed., *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, 4085–4091. ijcai.org.

Ye, H.; Jiang, X.; Luo, Z.; and Chao, W. 2018. Interpretable Charge Predictions for Criminal Cases: Learning to Generate Court Views from Fact Descriptions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 1854–1864. New Orleans, Louisiana: Association for Computational Linguistics.

Yin, P.; Lu, Z.; Li, H.; and Kao, B. 2016. Neural Enquirer: Learning to Query Tables in Natural Language. In Kambhampati, S., ed., *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, 2308–2314. IJCAI/AAAI Press.

Yu, X.; Chen, T.; Yu, Z.; Li, H.; Yang, Y.; Jiang, X.; and Jiang, A. 2020. Dataset and Enhanced Model for Eligibility Criteria-to-SQL Semantic Parsing. In *Proceedings of The 12th Language Resources and Evaluation Conference*, 5829–5837.

Zhang, Z.; Zhang, L.; Zhang, H.; He, W.; Sun, Z.; Cheng, G.; Liu, Q.; Dai, X.; and Qu, Y. 2018. Towards Answering Geography Questions in Gaokao: A Hybrid Approach. In *Knowledge Graph and Semantic Computing. Knowledge Computing and Language Understanding - Third China Conference, CCKS 2018, Tianjin, China, August 14-17, 2018, Revised Selected Papers*, 1–13.

Zhong, H.; Guo, Z.; Tu, C.; Xiao, C.; Liu, Z.; and Sun, M. 2018. Legal judgment prediction via topological learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 3540–3549.