2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

DeCA: A Decomposition-Enhanced Framework for Query-Focused Table Summarization

Luyi Wang[1], Yake Niu[1], Tiantian Peng[1], Renjie Ci[1], and Hui Zhao[1,2(✉)]

[1] Software Engineering Institute, East China Normal University, Shanghai, China
[2] Shanghai Key Laboratory of Trustworthy Computing, Shanghai, China
{luyi.wang,yake.niu,tiantian.peng,cirenjie}@stu.ecnu.edu.cn,
hzhao@sei.ecnu.edu.cn

Abstract. Query-focused table summarization aims to generate personalized summaries by reasoning and analyzing tabular data in response to user queries. Existing large language models (LLMs) based methods enhance summarization by utilizing intermediate facts to support reasoning. However, their effectiveness is constrained by limited inference rules and the generation of erroneous or redundant sub-queries, which can misguide the reasoning process and introduce misleading information into the summary. To address these issues, we propose DeCA, an innovative framework designed to generate non-repetitive and relevant sub-queries that support LLM reasoning and improve summary quality. Our framework comprises four modules: (1) Table Schema Extractor that interprets table structure and information; (2) Query Decomposer that recursively decomposes queries; (3) Sub-query Checker that verifies non-repetition, relevance, and dependencies among sub-queries; and (4) Answer Generator that generates summaries employing a hint-based answering strategy. Furthermore, we construct CQTS, the first large-scale Chinese table dataset for query-focused table summarization, consisting of 2,956 tables and 6,721 query-summary pairs. Extensive experiments on CQTS and two English datasets, QTSumm and FeTaQA, demonstrate that DeCA enhances LLM reasoning and outperforms existing methods in summary generation and sub-queries formulation.

Keywords: Query Decomposition, Query-Focused Table Summarization, Large Language Model Reasoning, Chinese Table Dataset.

1 Introduction

Tables provide a concise and structured representation of information, enabling efficient data retrieval and analysis. Query-focused table summarization task [1] addresses the challenge of producing a tailored summary by reasoning over tabular content in response to a user query. Despite advancements in natural language understanding and generation, large language models (LLMs) still face substantial difficulties in generating faithful and comprehensive summaries. As shown in Fig. 1, when presented with a query containing rich semantic content and intricate conditions, LLMs must utilize

reasoning abilities to identify relationships among data points within the table and generate a summary that satisfies the informational requirements of the query.

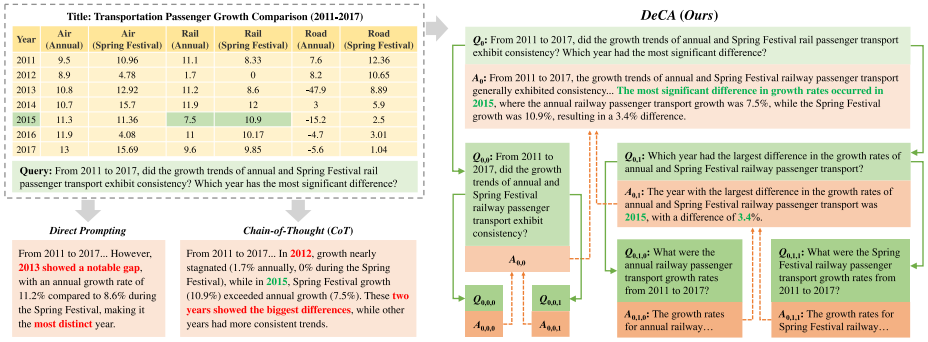

Fig. 1. An example of query-focused table summarization. Given a table and a complex query, LLMs' incorrect reasoning and analysis results in inaccurate summaries, with errors highlighted in red. We propose the DeCA framework to optimize the reasoning process of LLMs by decomposing and checking sub-queries before generating summaries with hints.

Recent approaches [1, 2, 3] have utilized auxiliary information, referred to as facts or hints. By leveraging these facts, LLM can comprehend complex queries and generate high-quality summaries from tabular data. However, these facts come with certain limitations. Rule-based facts [1] are constrained by a human-predefined inference type set, hindering flexibility and adaptability across diverse scenarios. Another type of fact [2] involves query-relevant cells extracted from the tables, potentially underutilizing the model's reasoning capabilities. Additionally, using sub-queries and sub-answers as hints [3] introduces challenges, as uncontrolled decomposition may lead to repetitive, irrelevant, and erroneous sub-queries. Thus, these ineffective sub-queries fail to support the model's reasoning process, leading to incorrect summaries.

To address these issues in the query-focused table summarization task, we propose **DeCA**, a novel framework to enhance the reasoning capabilities of LLMs through structured query **de**composition, sub-queries quality **c**hecking, and **a**nswer generation. DeCA consists of four modules. (1) **Table Schema Extractor** analyzes table information and extracts the table schema. (2) **Query Decomposer** recursively decomposes the query into sub-queries until no further decomposition is possible. (3) **Sub-query Checker** evaluates each sub-query based on three criteria: non-repetition, relevance, and dependency. (4) **Answer Generator** generates summaries for each sub-query with its children, dependency, and ancestor queries as contextual hints.

Furthermore, we observe a lack of query-focused table summarization datasets in Chinese. Current Chinese datasets over tabular data [4, 5] mainly focus on text-to-SQL and table question answering. Moreover, some Chinese table datasets [6, 7] are merely translations of English datasets, leading to inaccuracies in idiomatic expressions and language-specific phrases. Therefore, we construct **CQTS**, the first **C**hinese table dataset for **q**uery-focused **t**able **s**ummarization, comprising 2,956 tables and 6,721 query-summary pairs. This dataset contributes to advancing Chinese table processing and addresses the shortage of Chinese datasets in the query-focused table summarization task.

Prior works rely solely on automatic metrics. Automatic evaluation struggles to consistently measure query-focused table summarization [1, 8], whereas LLM-based evaluation aligns more closely with human judgment [9]. Different from prior works, we incorporate automatic and LLM-based evaluation methods to assess summary quality in our experiments.

Our main contributions are concluded as follows:

- We propose DeCA, a novel framework for query-focused table summarization. By structured query decomposition, sub-queries quality checking, and a hint-based answering mechanism, DeCA enhances LLM reasoning abilities to generate faithful and comprehensive summaries.
- We develop the CQTS dataset, consisting of 2,956 tables and 6,721 query-summary pairs. To the best of our knowledge, CQTS fills a blank in Chinese data for query-focused table summarization and contributes to developing methods for processing Chinese tables.
- We conduct extensive experiments across our proposed CQTS and two English datasets, QTSumm [1] and FeTaQA [10], on seven LLMs. The results indicate that DeCA surpasses existing methods in generating faithful and comprehensive summaries and decomposing non-repetitive and relevant sub-queries.

2 Related Work

2.1 Question Decomposition

Question decomposition effectively enhances the reasoning abilities of models by breaking down complex questions into sub-questions [11, 12]. For table-related tasks, Dater [13] and TaPERA [3] utilize LLMs to generate sub-queries and sub-answers to produce final results. However, most existing methods treat the LLM output as the final decomposition without checking the quality of sub-queries. A single-round decomposition often fails to provide sufficient facts to resolve the original question.

2.2 Table Related Dataset

Most existing works focus on training language-specific models with various English table datasets [14, 15], while Chinese table datasets remain limited. Current Chinese table datasets primarily focus on text-to-SQL [6, 16] and table question answering [5, 17]. Some researchers employ machine translation to convert English datasets into Chinese [6, 7]. Nevertheless, these translated datasets often fail to capture Chinese linguistic features, particularly idiomatic phrases and language-specific terms.

3 Preliminary

The query-focused table summarization task can be formulated as follows. Given a query Q and a table T, the table $T = W \cup \{t_{i,j} \mid i \leq R_T, j \leq C_T\}$ contains R_T rows and C_T

columns. W represents the table title. $t_{i,j}$ indicates the textual content in the (i,j)-th cell. The task objective is to generate a paragraph-long textual summary $Y = (y_1, y_2, \ldots, y_n)$ based on the query Q and table T:

$$Y = \text{argmax} \prod_{i=1}^{n} P(y_i \mid y_{<i}, Q, T; \theta) \quad (1)$$

where θ denotes the parameters of a text generation model, and y_i denotes the i-th tokens in the generated summary.

4 Dataset CQTS

4.1 Dataset Construction

In constructing the CQTS, we adhere to several principles that guide the collection of tables, queries, and summaries to ensure the quality and reliability of the CQTS dataset:

- **Informativeness**: The table should be rich in comparisons, statistics, and analytical content.
- **Meaningfulness**: The query should reflect practical, real-world information needs when analyzing tables.
- **Complexity**: The query should involve multi-hop reasoning and diverse structures.
- **Fluency**: Both the query and its corresponding table summary should be coherent and linguistically fluent.
- **Faithfulness**: The summary should be factually consistent with the content in the source table.
- **Comprehensiveness**: The summary should provide sufficient details and analyses of the source table to address the query fully.

Table Source. The source tables in CQTS are derived from two Chinese table datasets: the text-to-SQL dataset TableQA [4] and the table QA dataset IM-TQA [5]. For TableQA, we randomly sample 2,815 tables and conduct manual modifications to address several data quality issues. For IM-TQA, we select horizontal and vertical tables and merge small tables with identical schemas, yielding 141 tables. To address the absence of table titles in both datasets, we employ GPT-4o to generate concise and semantically appropriate titles based on table content. All generated titles are manually reviewed and validated for accuracy and relevance.

4.2 Data Annotation Pipeline

LLMs have already served as scalable and cost-efficient data annotators. However, their outputs do not always align with human preferences and values. Integrating human feedback enhances LLM-generated content's quality, diversity, and reliability. Thus, we propose a hybrid annotation pipeline to integrate the efficiency of LLM-based generation with the precision of human validation. The pipeline comprises three steps, as

shown in Fig. 2A. Given a table, the *LLM Generator* first produces queries, corresponding table summaries, and relevant cell positions. Subsequently, the *LLM Checker* evaluates and refines queries lacking complexity or diversity using three table expansion strategies, as illustrated in Fig. 2B. Finally, the *Human Annotators* select high-quality annotations and provide feedback to improve the query generation iteratively. We utilize GPT-4o as the backbone of the Generator and Checker.

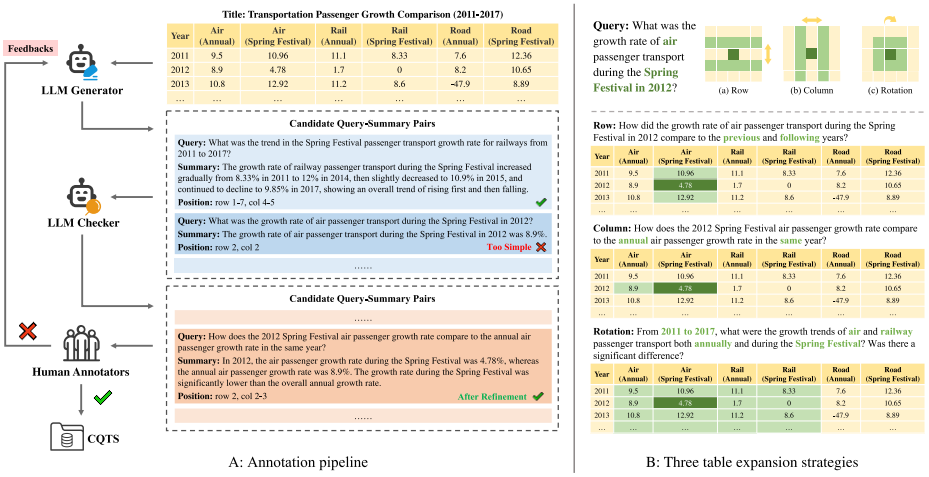

Fig. 2. A: Our proposed hybrid pipeline combines LLM-based annotation with human verification and feedback. B: Three table expansion strategies enhance the LLM Checker's ability to improve complexity and diversity. The example is a Chinese sample from the CQTS.

4.3 Data Statistics and Analysis

Key Statistics. Our dataset consists of 2,956 tables and 6,721 query-summary pairs. Detailed statistical information is provided in Table 1.

Table 1. Basic statistics of CQTS dataset.

Property	Value
Unique Tables	2,956
Query-Summary Pairs	6,721
Rows per Table (Median/Avg)	10/12.4
Columns per Table (Median/Avg)	7/7.5
Table Title Length (Median/Avg)	14/16.2
Query Length (Median/Avg)	34/34.9
Summary Length (Median/Avg)	79/84.8
Training Set Size (Table/Summary)	2,070/4,710 (70%)
Validation Set Size (Table/Summary)	443/1,004 (15%)
Test Set Size (Table/Summary)	443/1,007 (15%)

Dataset Domains. Fig. 3 illustrates the distribution of domains within the CQTS dataset. The visualization highlights the dataset's diversity, capturing a wide array of domains that reflect various real-world scenarios.

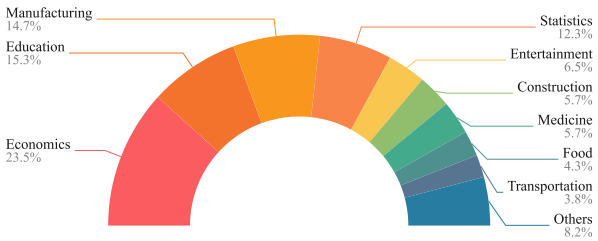

Fig. 3. Domain distribution of CQTS tables.

Comparison with Existing Datasets. Table 2 presents a comparison of CQTS with existing Chinese table datasets.

Table 2. Comparison between CQTS and existing table datasets in Chinese.

Dataset	Table Source	#Tables	#Queries	Answer Format
	Text-to-SQL			
TableQA [4]	Reports, Spreadsheets	6,029	64,891	SQL
CSpider [18]	Spider [19]	876	9,691	SQL
CRUDSQL [16]	TableQA	625	10,000	SQL
	Table Question Answering			
FewTUD [20]	Baidu Baike, E-commerce website	108	2,200	Short Text
IM-TQA [5]	Reports, Baidu Baike	1,200	5,000	Short Text
RETQA [17]	Real Estate Reports	4,932	20,762	Short Text, SQL
	Query-Focused Table Summarization			
CQTS (Ours)	**TableQA, IM-TQA**	**2,956**	**6,721**	**Paragraph-long Text**

5 DeCA Framework

5.1 Overview

DeCA comprises four key components, as illustrated in Fig. 4. Given a query Q over a table T, the *Table Schema Extractor* analyzes table information and extracts the schema T_{schema}. Utilizing T_{schema}, the *Query Decomposer* splits Q into a sequence of sub-queries. Each sub-query is evaluated by the *Sub-query Checker* to ensure non-repetition, relevance, and appropriate dependencies. Checked sub-queries are recursively decomposed by the *Query Decomposer* until further decomposition is no longer feasible. Upon completing the decomposition and checking process, a decomposition directed acyclic graph (Decomposition DAG) is formed, containing all sub-queries and their relationships. Finally, the Decomposition DAG is input into the *Answer Generator* to generate the final summary.

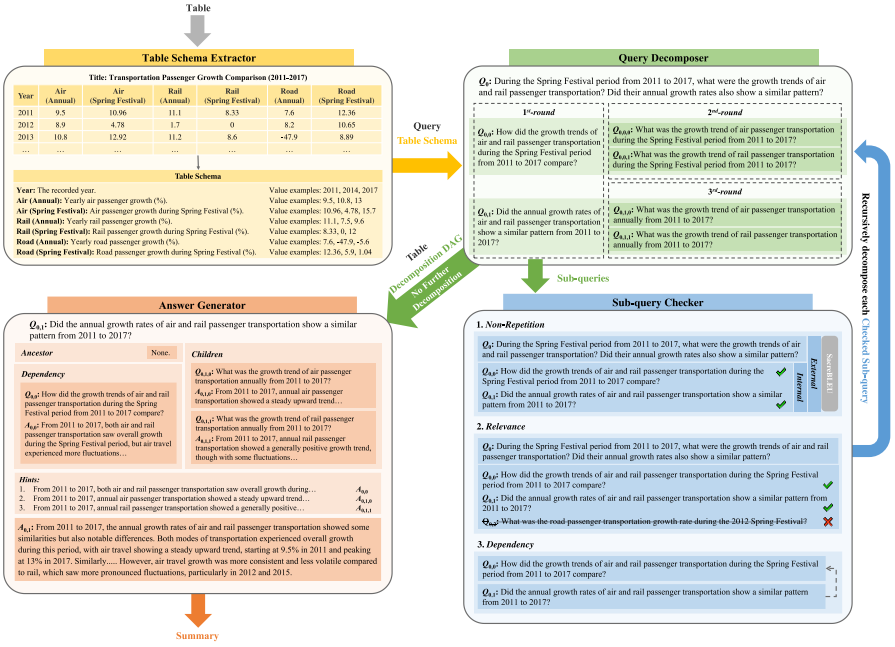

Fig. 4. The DeCA framework. Given a query over a table, DeCA first extracts the table schema. It then recursively decomposes the query and checks sub-queries' quality. Finally, a Decomposition DAG is constructed to generate the summary.

5.2 Table Schema Extractor

The objective of the Table Schema Extractor is to convert the implicit metadata in titles and headers of tables to explicit descriptions. We employ In-Context Learning (ICL) to prompt the LLM with a demonstration. The demonstration includes a table and a human-annotated table schema comprising column descriptions and example cells. Based on the provided demonstration, the LLM generates a short description and selects cells as examples for each column in a given table T. The resulting collection of metadata descriptions and example cells constitutes the table schema T_{schema}, which serves as a condensed representation of the original table T.

5.3 Query Decomposer

Given the original query Q and the table schema T_{schema}, the Query Decomposer recursively decomposes Q until it meets termination conditions. To ensure the sub-queries are non-repetitive, relevant, and organized in a dependency-aware order, we introduce the Sub-query Checker (Section 5.4) to check candidate sub-queries at each decomposition step. These candidate sub-queries can be further decomposed if they pass the checking. During the entire process, a Decomposition DAG is maintained to represent the structure among queries.

Termination Conditions. The following three termination conditions govern the query decomposition process:

- **LLM-based judgment**: We prompt the Query Decomposer to determine whether the decomposition is necessary for q. If required, q is broken down into a set of sub-queries. Otherwise, the decomposition process is ceased.
- **Maximum decomposition attempts**: If the sub-queries fail to pass the checking, q must be re-decomposed. To prevent excessive attempts, we define a threshold $TH_{attempt}$ to limit the number of decomposition attempts. The decomposition is retried if fewer than two sub-queries pass the checking in an attempt. The decomposition is terminated if no set of sub-queries passes the checking after $TH_{attempt}$ attempts.
- **Maximum decomposition depth**: To avoid infinite recursion, we define a maximum recursive depth, TH_{depth}. The depth of the original query Q is initialized as 0, and the depth of sub-queries is incremented by one within each decomposition step. The decomposition of q is stopped once the depth of q reaches TH_{depth}.

Decomposition DAG Construction. In the Decomposition DAG, nodes represent queries. Edges denote queries' relationships, as shown in Fig. 5. At each decomposition step, checked sub-queries are added to the DAG and sequentially assigned identifiers. Two types of relationships are then established: (1) *decomposition relationships*, which link each sub-query to its parent query, and (2) *dependency relationships*, which capture dependencies among sub-queries. The dependency relationships are obtained from the result of checking dependency in the Sub-query Checker. Based on the relationships, the queries associated with a given query q can be categorized into three types:

- **Children queries (Sub-queries)** are decompositions of q, providing fine-grained support for constructing the summary.
- **Dependency queries** represent queries with which q has dependency relationships, ensuring logical consistency in the reasoning process.
- **Ancestor queries** consist of (1) the dependency queries of q's parent query and (2) ancestor queries inherited from q's parent query, offering broader contextual knowledge.

As illustrated in Fig. 5, query $Q_{0,1,1}$ is an example to demonstrate the three types of related queries in the Decomposition DAG.

Retrieval of Context in the Decomposition Prompt. We adopt a tree structure to retrieve the context for the decomposition prompt. Before decomposing any query q, we extract a Decomposition Tree from the current Decomposition DAG. The Tree retains all existing queries as nodes while preserving only decomposition relationships as edges. Taking the original query Q as the root node, we traverse the Tree to obtain the current leaf nodes. These leaf nodes are sorted in dictionary order by their identifiers, and the associated queries collectively form the Current Decomposition Plan. The Plan is used as contextual input in the prompt for decomposing q. As shown in Fig. 5, when decomposing $Q_{0,0}$, the current leaf nodes are $\{Q_{0,0}, Q_{0,1}, \ldots, Q_{0,i}\}$ with Q_0 as the root.

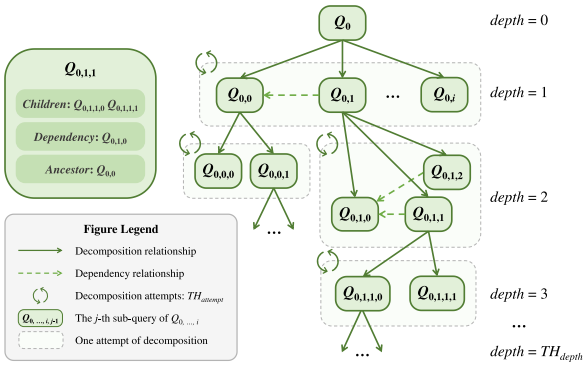

Fig. 5. An example of the Decomposition DAG.

5.4 Sub-query Checker

We introduce the Sub-query Checker to evaluate the quality of sub-queries in each decomposition step, guided by three criteria: non-repetition, relevance, and dependency.

- **Non-repetition**: To reduce semantic repetition, we compute BLEU scores [21] to measure similarity (1) between each sub-query and its parent query and (2) among sub-queries within the same decomposition step. A sub-query q is removed if its similarity to the parent query exceeds a predefined threshold. Additionally, if two sub-queries are overly similar, one is discarded.
- **Relevance**: We input the table schema, the query, and corresponding sub-queries into the LLM and prompt it to evaluate whether the sub-queries capture the intent of the original query. Irrelevant sub-queries identified by the LLM are deleted.
- **Dependency**: For each sub-query q, we prompt the LLM to identify dependency relationships, specifying which sub-queries need to be answered before q in a given set of sub-queries. Based on these dependency relationships, we apply a topological sort to arrange the sub-queries in a logical answering order.

After completing these checking steps, we obtain a set of distinct and semantically faithful sub-queries along with their dependency relationships.

5.5 Answer Generator

After completing the decomposition and checking process, we utilize the constructed Decomposition DAG to generate a summary for the original query Q. The summary generation process follows a depth-first search (DFS) on the Decomposition DAG. For each unanswered query q, we first retrieve available answers as hints from related queries in the DAG, including q's children, dependency, and ancestor queries. The collected hints, along with the query q and table T, are provided to the LLM to generate a paragraph-long answer as the summary for q. In contrast to the forward-driven decomposition process, the summary generation adopts a backtracking strategy. The final generated answer serves as the summary of the original query Q.

6 Experiments

6.1 Experiment Setting

We utilize three datasets: our proposed Chinese dataset, CQTS, and two English datasets, QTSumm [1] and FeTaQA [10]. We conduct extensive experiments to answer the following research questions:

- **RQ1**: How do fine-tuning and LLM prompting methods perform on our proposed CQTS dataset?
- **RQ2**: How effective is our proposed DeCA framework across multiple datasets, including CQTS, QTSumm, and FeTaQA?
- **RQ3**: Does the quality of query decomposition impact the quality of summaries?

6.2 Baselines

Existing baselines for the query-focused table summarization task can be primarily categorized into fine-tuning methods and LLM prompting methods:

- **Fine-tuning**: We fine-tune two representative text generation models, BART [22] and mT5 [23], using the training set of the CQTS dataset.
- **LLM Prompting**: We evaluate both open-source and closed-source LLMs using various prompting methods. Open-source models include the Llama-3.1 series[1] (8B, 70B) and the Qwen-2.5 series[2] (7B, 14B, 32B). Closed-source models[3] include `GPT-3.5-turbo-1106` and `GPT-4o-0806`. The following prompting methods are employed across these LLMs: Zero/One-shot Direct Prompting, Chain-of-Thought (CoT) [24], Blueprint [25], ReFactor [1], Dater [13], and TaPERA [3].

6.3 Evaluation Metrics

In addition to widely used automatic natural language generation (NLG) metrics, we employ LLM-based evaluation for a more comprehensive assessment. We evaluate the entire test set for automatic metrics, while for G-Eval, we randomly select 200 samples.

- **Automatic Evaluation**: We utilize SacreBLEU [21], ROUGE-L [26], METEOR [27], BERTScore [28] and PARENT [29] to evaluate the quality of summaries.
- **LLM-based Evaluation**: We leverage the G-Eval [9] to assess the faithfulness and comprehensiveness of summaries. We also conduct a pairwise comparison to assess the quality of query decomposition, prompting GPT-4o to select the better result based on non-repetition and relevance.

---

1 https://huggingface.co/meta-llama/Llama-3.1-{size}B-Instruct
2 https://huggingface.co/Qwen/Qwen2.5-{size}B-Instruct
3 https://openai.com/api/

6.4 Implementation Details

- **Fine-tuning methods**: We utilize the `large` version of models, conducting training on two NVIDIA Tesla V100 32GB GPUs. Each model is trained for 20 epochs. We set the batch size to 4 for `BART-large-Chinese`[4] and 2 for `mT5-large`[5].
- **LLM prompting methods**: We use the `instruct` version for all open-source LLMs. The hyperparameters of LLMs are set with the temperature to 0.7, Top P to 1.0, and maximum output length to 512. Following the previous works, tabular data is represented in *Markdown* format.
- **Settings in DeCA**: Due to budget constraints, we set the maximum decomposition attempts $TH_{attempt}$ to 3 and the maximum depth TH_{depth} to 4 for DeCA.

6.5 Main Results

RQ1: How do fine-tuning and LLM prompting methods perform on our proposed CQTS dataset? We assess the CQTS dataset using fine-tuning and LLM prompting methods. As shown in Table 3, models fine-tuned on the CQTS consistently underperform relative to LLM prompting methods, particularly under G-Eval evaluation. For LLM prompting methods, reasoning-enhanced prompting (CoT, Blueprint, and DeCA) performs better than few-shot prompting (zero-shot and one-shot). These findings emphasize that explicit reasoning guidance substantially enhances model performance for the query-focused table summarization task.

RQ2: How effective is our proposed DeCA framework across multiple datasets, including CQTS, QTSumm, and FeTaQA? We compare the performance of DeCA against other LLM prompting methods on CQTS, QTSumm, and FeTaQA. As shown in Tables 3, 4, and 5, few-shot prompting is not consistently beneficial. The performance of reasoning-enhanced methods, including CoT and Blueprint, is also unstable. In contrast, our proposed DeCA consistently improves G-Eval scores across all datasets, indicating that DeCA is highly effective and generalizable.

RQ3: Does the quality of query decomposition impact the quality of summaries? We compare the sub-queries generated by Blueprint and DeCA. For an original query, Blueprint's sub-queries are extracted from its QA-plan, while DeCA's are derived from the Current Decomposition Plan. Given a table and two decomposition results, we prompt the LLM to select the superior decomposition based on criteria of non-redundancy and relevance. As shown in Fig. 6, DeCA consistently generates higher-quality sub-queries than Blueprint. While the summary quality generated by Blueprint varies across different models, DeCA consistently achieves higher G-Eval scores, as presented in Tables 3, 4, and 5. These results confirm that high-quality query decomposition is critical in improving summary generation.

[4] https://huggingface.co/fnlp/bart-large-chinese
[5] https://huggingface.co/google/mt5-large

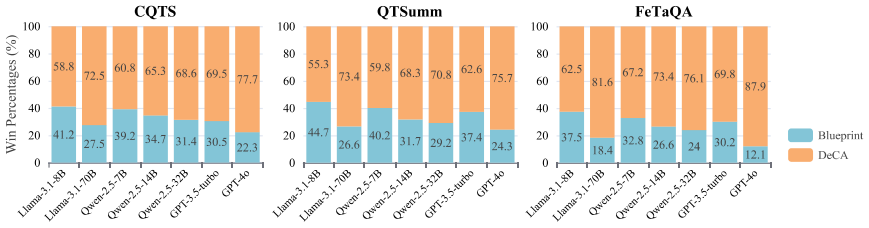

Fig. 6. The pairwise comparison results between the Blueprint and DeCA framework. We randomly select 200 samples for each category of results.

Table 3. Results on the CQTS. The best results are highlighted in green, while the second best are in blue.

Model	Method	Automatic Metrics					G-Eval	
		SacreBLEU	ROUGE-L	METEOR	BERTScore	PARENT	Faith.	Compre.
BART-large-Chinese	fine-tuning	30.03	44.21	41.40	79.97	24.06	2.25	2.36
mT5-large	fine-tuning	32.88	44.38	42.14	80.85	25.88	2.72	2.74
Llama-3.1-8B	0-shot	24.53	39.46	49.91	78.10	30.74	3.86	3.78
	1-shot	26.71	40.74	48.25	78.86	29.99	3.84	3.74
	CoT	22.84	37.79	49.05	77.53	30.44	3.87	3.82
	Blueprint	17.95	34.18	47.63	76.16	32.31	3.88	3.84
	DeCA	27.04	41.10	49.18	79.25	30.58	3.94	3.83
Llama-3.1-70B	0-shot	36.05	48.55	54.87	81.75	33.15	4.54	4.27
	1-shot	38.09	49.38	54.60	82.28	32.69	4.42	4.24
	CoT	35.49	48.37	54.90	81.72	32.80	4.53	4.31
	Blueprint	39.62	48.64	52.02	81.53	31.43	4.34	4.17
	DeCA	35.04	48.85	55.55	82.12	34.09	4.57	4.34
Qwen-2.5-7B	0-shot	27.53	43.78	54.36	80.37	34.18	4.06	4.10
	1-shot	31.78	46.68	55.02	81.57	34.72	4.27	4.17
	CoT	25.12	42.18	54.83	79.83	34.89	4.21	4.13
	Blueprint	24.95	41.76	53.86	79.58	34.35	4.26	4.12
	DeCA	25.55	43.01	55.41	80.38	36.49	4.30	4.20
Qwen-2.5-14B	0-shot	31.11	47.61	58.89	81.99	35.11	4.63	4.47
	1-shot	32.56	48.52	59.24	82.36	35.91	4.64	4.45
	CoT	28.85	45.75	58.56	81.28	35.23	4.64	4.49
	Blueprint	29.13	46.14	57.81	81.14	35.27	4.70	4.47
	DeCA	29.30	46.02	58.53	81.20	36.56	4.65	4.54
Qwen-2.5-32B	0-shot	27.39	44.95	58.06	81.23	36.00	4.73	4.58
	1-shot	29.64	46.88	59.13	81.92	36.50	4.77	4.60
	CoT	26.06	44.30	58.17	80.84	36.05	4.72	4.61
	Blueprint	27.02	44.28	58.23	80.55	37.00	4.79	4.53
	DeCA	29.54	46.94	58.95	81.67	36.58	4.81	4.63
GPT-3.5-turbo	0-shot	29.58	44.13	53.19	80.11	32.01	4.25	4.10
	1-shot	34.85	47.22	53.77	81.54	31.83	4.16	4.02
	CoT	27.57	42.36	52.87	79.54	32.60	4.27	4.17
	Blueprint	30.73	45.02	52.05	80.63	31.82	4.28	4.15
	DeCA	31.75	45.31	52.92	81.11	32.26	4.24	4.19
GPT-4o	0-shot	29.73	46.81	61.03	81.95	37.54	4.78	4.68
	1-shot	34.49	50.64	62.36	83.32	37.95	4.78	4.59
	CoT	29.16	46.71	61.21	81.86	37.77	4.81	4.58
	Blueprint	28.44	46.21	59.81	81.50	37.42	4.80	4.62
	DeCA	33.45	51.01	62.34	83.32	38.88	4.84	4.67

Table 4. Results on the QTSumm. The best results are highlighted in green, while the second best are in blue. Methods marked with * are referenced from the original papers.

Model	Method	Automatic Metrics					G-Eval	
		SacreBLEU	ROUGE-L	METEOR	BERTScore	PARENT	Faith.	Compre.
Llama-3.1-8B	0-shot	17.90	37.20	46.46	90.32	26.78	4.27	4.24
	1-shot	16.83	34.47	43.98	89.74	24.23	4.05	3.96
	CoT	17.94	36.70	46.87	90.18	26.58	4.19	4.19
	Blueprint	11.30	28.19	44.28	87.40	24.20	4.00	4.00
	DeCA	21.59	39.10	45.69	90.58	26.14	4.32	4.26
Llama-3.1-70B	0-shot	18.84	38.89	50.30	90.83	28.87	4.73	4.72
	1-shot	20.32	40.26	49.83	91.05	28.92	4.76	4.70
	CoT	19.28	39.36	50.21	90.84	29.35	4.76	4.67
	Blueprint	15.55	35.14	49.01	89.71	28.18	4.77	4.69
	DeCA	22.18	41.39	49.59	91.11	29.74	4.80	4.75
Qwen-2.5-7B	0-shot	18.98	37.22	45.34	90.37	26.48	4.14	4.15
	1-shot	19.40	38.44	46.38	90.52	28.45	4.09	4.17
	CoT	18.50	36.99	45.77	90.32	26.89	4.24	4.21
	Blueprint	16.84	35.75	46.865	90.06	27.40	4.20	4.28
	DeCA	18.46	38.17	47.37	90.38	28.48	4.22	4.27
Qwen-2.5-14B	0-shot	14.48	33.13	46.75	89.92	25.84	4.66	4.65
	1-shot	16.01	35.27	47.67	90.21	27.04	4.59	4.56
	CoT	15.05	33.80	47.42	90.05	26.36	4.64	4.56
	Blueprint	13.47	31.58	46.97	89.54	27.31	4.67	4.66
	DeCA	21.65	41.47	50.09	91.14	30.31	4.74	4.67
Qwen-2.5-32B	0-shot	16.76	35.87	48.73	90.39	28.12	4.78	4.72
	1-shot	16.85	36.26	49.25	90.46	28.40	4.75	4.74
	CoT	16.74	35.66	48.50	90.34	27.40	4.84	4.81
	Blueprint	14.46	32.97	48.07	89.76	28.41	4.78	4.77
	DeCA	21.29	41.28	50.54	91.17	30.19	4.85	4.80
GPT-3.5-turbo	0-shot	20.06	37.91	46.64	90.69	26.94	4.39	4.29
	1-shot	19.98	38.37	47.84	90.74	28.30	4.43	4.30
	CoT	19.95	37.57	46.59	90.65	26.92	4.26	4.17
	Blueprint	14.07	30.31	44.61	89.31	25.38	4.26	4.18
	ReFactor*	19.90	39.50	48.80	91.20	-	-	-
	Dater*	16.60	35.20	35.50	82.90	-	-	-
	TaPERA*	14.60	33.00	33.20	88.70	-	-	-
	DeCA	21.26	40.21	47.49	90.91	27.48	4.48	4.41
GPT-4o	0-shot	16.68	36.65	50.63	90.54	29.70	4.84	4.85
	1-shot	19.09	39.71	51.58	91.03	29.83	4.91	4.86
	CoT	17.07	37.47	51.14	90.64	29.97	4.92	4.91
	Blueprint	12.66	30.74	47.58	89.41	28.01	4.86	4.85
	DeCA	20.38	42.38	52.86	91.27	32.22	4.95	4.90

Table 5. Results on the FeTaQA. The best results are highlighted in green, while the second best are in blue. Methods marked with * are referenced from the original papers.

Model	Method	Automatic Metrics					G-Eval	
		SacreBLEU	ROUGE-L	METEOR	BERTScore	PARENT	Faith.	Compre.
Llama-3.1-8B	0-shot	27.10	51.45	58.81	92.33	27.76	4.65	4.55
	1-shot	24.99	49.09	57.18	91.90	26.66	4.51	4.37
	CoT	27.09	51.51	59.04	92.39	27.94	4.59	4.46
	Blueprint	12.61	34.54	51.29	87.76	22.15	4.54	4.40
	DeCA	25.28	50.71	57.83	92.15	26.77	4.68	4.56
Llama-3.1-70B	0-shot	27.09	52.75	61.63	92.62	29.63	4.86	4.79
	1-shot	28.01	53.19	62.15	92.70	30.47	4.86	4.82
	CoT	27.88	53.15	61.10	92.59	29.32	4.88	4.80
	Blueprint	14.66	44.07	54.68	90.36	25.06	4.85	4.73
	DeCA	25.58	52.09	62.54	92.47	29.95	4.89	4.84
Qwen-2.5-7B	0-shot	29.47	51.73	58.81	92.40	28.42	4.65	4.56
	1-shot	29.16	51.98	59.00	92.46	28.99	4.63	4.60
	CoT	29.46	51.76	59.01	92.44	28.54	4.66	4.57
	Blueprint	19.32	40.27	38.87	88.94	19.08	4.46	4.42
	DeCA	27.64	51.86	58.48	92.35	28.52	4.68	4.59
Qwen-2.5-14B	0-shot	28.10	52.21	59.95	92.59	28.40	4.84	4.68
	1-shot	28.92	52.96	60.57	92.63	28.85	4.84	4.69
	CoT	28.09	52.26	60.04	92.56	27.30	4.86	4.73
	Blueprint	25.81	51.47	60.21	92.23	28.43	4.78	4.74
	DeCA	27.12	52.27	60.05	92.45	28.22	4.89	4.75
Qwen-2.5-32B	0-shot	28.78	53.63	61.06	92.68	29.18	4.84	4.72
	1-shot	29.80	54.19	61.35	92.75	29.62	4.87	4.75
	CoT	28.90	53.53	60.80	92.67	29.04	4.88	4.79
	Blueprint	25.70	51.69	60.86	92.42	29.95	4.85	4.82
	DeCA	27.98	53.05	60.58	92.57	28.98	4.90	4.81
GPT-3.5-turbo	0-shot	28.27	52.15	58.17	92.40	26.85	4.62	4.41
	1-shot	27.53	52.52	59.87	92.49	28.19	4.73	4.49
	CoT	28.51	52.28	58.02	92.41	27.12	4.62	4.40
	Blueprint	26.32	49.26	54.52	91.53	24.01	4.38	4.05
	ReFactor*	26.20	53.60	57.20	87.40	-	-	-
	Dater*	29.80	54.00	59.40	88.20	-	-	-
	TaPERA*	29.50	53.40	58.20	86.10	-	-	-
	DeCA	25.07	51.57	59.90	92.32	27.61	4.71	4.55
GPT-4o	0-shot	27.31	53.79	63.47	92.75	30.67	4.92	4.88
	1-shot	29.87	54.63	63.08	92.89	30.34	4.91	4.86
	CoT	27.14	53.81	63.48	92.73	31.00	4.93	4.86
	Blueprint	25.45	52.65	61.35	92.41	29.33	4.89	4.79
	DeCA	29.34	55.82	63.22	92.91	30.55	4.95	4.90

6.6 Analyses of the Decomposition DAG

For an original query Q, DeCA constructs a Decomposition DAG to represent the generated sub-queries. The structure of the DAG offers insights of Q's complexity. To determine the nature of queries across datasets, we analyze the Decomposition DAGs generated from CQTS, QTSumm, and FeTaQA. Specifically, we report two metrics: *Avg Sub-Queries* and *Max Depth (≥95%)*. *Avg Sub-Queries* reflects the average number of sub-queries per decomposition step. *Max Depth (≥95%)* indicates that the maximum

decomposition depth for at least 95% of the queries in the dataset does not exceed this value. As shown in Table 6, the CQTS and QTSumm datasets exhibit higher *Avg Sub-Queries* and *Max Depth (≥95%)* compared to FeTaQA across all evaluated models. These findings suggest that queries in CQTS and QTSumm are generally more complex and require further decomposition.

Table 6. Statistics of Decomposition DAG in different datasets.

Model	CQTS		QTSumm		FeTaQA	
	Avg Sub-queries	Max Depth (≥95%)	Avg Sub-queries	Max Depth (≥95%)	Avg Sub-queries	Max Depth (≥95%)
Llama-3.1-8B	2.64	3	2.90	3	2.50	3
Llama-3.1-70B	2.80	3	3.14	3	2.60	2
Qwen-2.5-7B	2.77	3	3.05	3	2.60	2
Qwen-2.5-14B	3.39	3	3.92	3	2.85	2
Qwen-2.5-32B	3.45	3	3.95	3	3.02	2
GPT-3.5-turbo	2.32	1	2.86	2	2.72	1
GPT-4o	3.72	3	4.12	2	3.33	1

7 Conclusion

In this paper, we propose DeCA, a novel query decomposition framework based on LLMs to address the challenges of complex query-focused table summarization. Combining the table schema extractor, the query decomposer, the sub-query checker, and the answer generator, DeCA effectively utilizes sub-queries as informative hints to enhance the quality of the generated summaries. Furthermore, we introduce CQTS, the first Chinese dataset for query-focused table summarization. We conduct extensive experiments across three datasets, CQTS, QTSumm, and FeTaQA, utilizing seven LLMs. The results demonstrate that DeCA consistently outperforms existing approaches regarding summary faithfulness, comprehensiveness, and the quality of sub-query decomposition. In future work, we plan to extend DeCA to support tables with diverse structures and address a broader range of table-centric tasks.

References

1. Zhao, Y., Qi, Z., Nan, L., Mi, B., Liu, Y., Zou, W., Han, S., Chen, R., Tang, X., Xu, Y., Radev, D., Cohan, A.: QTSumm: Query-Focused Summarization over Tabular Data. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pp. 1157–1172 (2023)
2. Zhang, W., Pal, V., Huang, J.-H., Kanoulas, E., de Rijke, M.: QFMTS: Generating Query-Focused Summaries over Multi-Table Inputs. In: ECAI 2024, pp. 3875–3882 (2024)
3. Zhao, Y., Chen, L., Cohan, A., Zhao, C.: TaPERA: Enhancing Faithfulness and Interpretability in Long-Form Table QA by Content Planning and Execution-based Reasoning. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 12824–12840 (2024)

4. Sun, N., Yang, X., Liu, Y.: TableQA: a Large-Scale Chinese Text-to-SQL Dataset for Table-Aware SQL Generation, ArXiv. (2020)
5. Zheng, M., Hao, Y., Jiang, W., Lin, Z., Lyu, Y., She, Q., Wang, W.: IM-TQA: A Chinese Table Question Answering Dataset with Implicit and Multi-type Table Structures. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 5074–5094 (2023)
6. Dou, L., Gao, Y., Pan, M., Wang, D., Che, W., Zhan, D., Lou, J.-G.: MultiSpider: Towards Benchmarking Multilingual Text-to-SQL Semantic Parsing. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, pp. 12745–12753 (2023)
7. Qiu, Z., Li, J., Huang, S., Jiao, X., Zhong, W., King, I.: CLongEval: A Chinese Benchmark for Evaluating Long-Context Large Language Models. In: Findings of the Association for Computational Linguistics: EMNLP 2024, pp. 3985–4004 (2024)
8. Wang, Y., Chen, L., Cai, S., Xu, Z., Zhao, Y.: Revisiting Automated Evaluation for Long-form Table Question Answering. In: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pp. 14696–14706 (2024)
9. Liu, Y., Iter, D., Xu, Y., Wang, S., Xu, R., Zhu, C.: G-Eval: NLG Evaluation using Gpt-4 with Better Human Alignment. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pp. 2511–2522 (2023)
10. Nan, L., Hsieh, C., Mao, Z., Lin, X.V., Verma, N., Zhang, R., Kryściński, W., Schoelkopf, H., Kong, R., Tang, X., Mutuma, M., Rosand, B., Trindade, I., Bandaru, R., Cunningham, J., Xiong, C., Radev, D., Radev, D.: FeTaQA: Free-form Table Question Answering. In: Transactions of the Association for Computational Linguistics, vol. 10, pp. 35–49 (2022)
11. Patel, P., Mishra, S., Parmar, M., Baral, C.: Is a Question Decomposition Unit All We Need? In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pp. 4553–4569 (2022)
12. Zhou, D., Schärli, N., Hou, L., Wei, J., Scales, N., Wang, X., Schuurmans, D., Cui, C., Bousquet, O., Le, Q., Chi, E.: Least-to-Most Prompting Enables Complex Reasoning in Large Language Models. In: The Eleventh International Conference on Learning Representations (2023)
13. Ye, Y., Hui, B., Yang, M., Li, B., Huang, F., Li, Y.: Large Language Models are Versatile Decomposers: Decompose Evidence and Questions for Table-based Reasoning. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 174–184 (2023)
14. Herzig, J., Nowak, P.K., Müller, T., Piccinno, F., Eisenschlos, J.: TaPas: Weakly Supervised Table Parsing via Pre-training. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 4320–4333 (2020)
15. Liu, Q., Chen, B., Guo, J., Ziyadi, M., Lin, Z., Chen, W., Lou, J.-G.: TAPEX: Table Pre-training via Learning a Neural SQL Executor. In: The Tenth International Conference on Learning Representation (2022)
16. Chen, X., You, J., Li, K., Li, X.: Beyond Read-Only: Crafting a Comprehensive Chinese Text-to-SQL Dataset for Database Manipulation and Query. In: Findings of the Association for Computational Linguistics: NAACL 2024, pp. 3383–3393 (2024)
17. Wang, Z., Yang, W., Zhou, K., Zhang, Y., Jia, W.: RETQA: A Large-Scale Open-Domain Tabular Question Answering Dataset for Real Estate Sector. In: The 39th Annual AAAI Conference on Artificial Intelligence (2024)
18. Min, Q., Shi, Y., Zhang, Y.: A Pilot Study for Chinese SQL Semantic Parsing. 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), pp. 3652–3658 (2019)

19. Yu, T., Zhang, R., Yang, K., Yasunaga, M., Wang, D., Li, Z., Ma, J., Li, I., Yao, Q., Roman, S., Zhang, Z., Radev, D.: Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 3911–3921 (2018)
20. Liu, R., Yuan, S., Dai, A., Shen, L., Zhu, T., Chen, M., He, X.: Few-Shot Table Understanding: A Benchmark Dataset and Pre-Training Baseline. In: Proceedings of the 29th International Conference on Computational Linguistics, pp. 3741–3752 (2022)
21. Post, M.: A Call for Clarity in Reporting BLEU Scores. In: Proceedings of the Third Conference on Machine Translation: Research Papers, pp. 186–191 (2018)
22. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 7871–7880 (2020)
23. Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., Raffel, C.: mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 483–498 (2021)
24. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q.V., Zhou, D.: Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In: Advances in Neural Information Processing Systems, vol. 35, pp. 24824–24837 (2022)
25. Narayan, S., Maynez, J., Amplayo, R.K., Ganchev, K., Louis, A., Huot, F., Sandholm, A., Das, D., Lapata, M.: Conditional Generation with a Question-Answering Blueprint. In: Transactions of the Association for Computational Linguistics, vol. 11, pp. 974–996 (2023)
26. Lin, C.-Y., Hovy, E.: Automatic Evaluation of Summaries Using N-gram Co-occurrence Statistics. In: Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, pp. 150–157 (2003)
27. Banerjee, S., Lavie, A.: METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In: Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, pp. 65–72 (2005)
28. Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., Artzi, Y.: BERTScore: Evaluating Text Generation with BERT. In: The Eighth International Conference on Learning Representations (2020)
29. Dhingra, B., Faruqui, M., Parikh, A., Chang, M.-W., Das, D., Cohen, W.: Handling Divergent Reference Texts when Evaluating Table-to-Text Generation. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 4884–4895 (2019)