scientific reports

Check for updates

OPEN

Enterprise chart question and answer method based on multi modal cross fusion

Xinxin Wang[1]✉, Liang Chen[2], Changhong Liu[3] & Jinyu Liu[4]

To enhance enterprises' interactive exploration capabilities for unstructured chart data, this paper proposes a multimodal chart question-answering method. Facing the challenge of recognizing curved and irregular text in charts, we introduce Gaussian heatmap encoding technology to achieve character-level precise text annotation. Additionally, we combine a key point detection algorithm to extract numerical information from the charts and convert it into structured table data. Finally, by employing a multimodal cross-fusion model, we deeply integrate the queried charts, user questions, and generated table data to ensure that the model can comprehensively capture chart information and accurately answer user questions. Experimental validation has demonstrated that our method achieves a precision of 91.58% in chart information extraction and a chart question-answering accuracy of 82.24%, fully proving the significant advantages of our proposed method in enhancing chart text recognition and question-answering capabilities. Through practical enterprise application cases, our method has shown its ability to answer four types of chart questions, exhibiting mathematical reasoning capabilities and providing robust support for enterprise data analysis and decision-making.

Enterprises have adopted data visualization techniques to integrate production and sales information from manufacturing workshops, enabling centralized display and effective coordination of information[1]. Data visualization techniques utilize various unstructured charts (such as line charts, bar charts, pie charts, scatter plots, etc.) to visually present and transmit data, providing users with a global overview of the data[2]. However, while charts can intuitively display the overall trends and proportional relationships in data, it is difficult to obtain precise numerical information solely through visual observation. For example, extracting specific numerical values from charts or comparing numerical differences between different elements requires more precise numerical support. Traditional methods of obtaining chart data primarily rely on SQL query statements to retrieve the required numerical information from databases. This approach not only requires users to possess professional SQL query skills, but also faces challenges such as data being scattered across different workshops and difficulties in directly accessing the raw data[3].

In response to the aforementioned issues, researchers have proposed chart question and answer (QA) technology in recent years[4]. The core of this technology lies in utilizing natural language processing techniques to accurately understand users' question intentions and automatically extract relevant information from chart data as answers. This approach not only satisfies users' need to access information without directly accessing enterprise databases, but also enables users without professional SQL query skills to quickly obtain key information from charts through natural language questions. The development of chart QA tasks is illustrated in Fig. 1. Early research primarily relied on manually defined rules and templates to match user questions with chart data. However, this approach was limited by its flexibility and scalability, making it difficult to handle complex and varied questions. With the advancement of deep learning techniques, researchers began utilizing deep learning models to address chart QA tasks. These models significantly improved their ability to understand chart data and questions by automatically learning data features. The IMG + QUES model proposed by Pal et al.[5] in 2012 was a pioneering work in this field. This model utilized convolutional neural networks (CNN) to extract chart features and combined them with long short-term memory networks (LSTM)[6] and multi-layer fully connected neural networks to generate answers. It achieved an accuracy of 59.41% on the Figure QA dataset (figure question answering)[7], laying the foundation for subsequent research. Subsequently, Kafle et al. proposed the SANDY model[8] and the DVQA dataset, addressing the limitation of only supporting binary judgment questions and introducing more complex answer types. Based on SANDY, the PReFIL model further improved the accuracy of

[1]School of Economics and Management, Shangluo University, Shangluo 726000, China. [2]The Shannxi Key Laboratory of Clothing Intelligence, Xi'an Polytechnic University, Xi'an 710048, China. [3]China Tobacco Chongqing Industrial Co.Ltd Qianjiang Cigarette Factory, Qianjiang, China. [4]Chongqing Vocational Institute of Tourism, Chongqing, China. ✉email: 595907410@qq.com

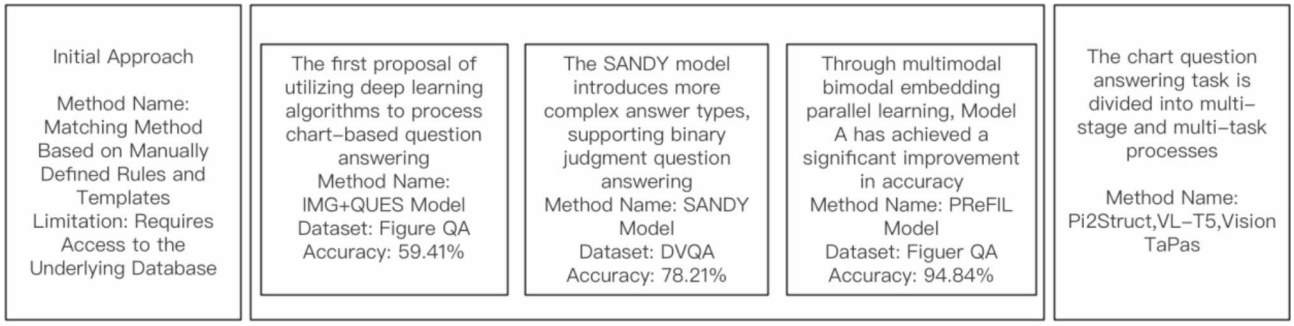


Fig. 1. Research status diagram of chart question answering tasks.

question answering by fusing question and image features to learn bimodal embeddings, achieving an accuracy of 80.88%[9].

In recent years, researchers have proposed multi-stage chart question-answering methods, such as the Pix2Struct[10] model proposed by Lee et al. in 2023. This model converts images into structured text descriptions and employs a similar approach to text information processing in LLM technology for chart question answering. Concurrently, the VL-T5[11] and Vision TaPas[12] models have further advanced the development of chart question answering technology by integrating chart visual information into text editors. Despite the significant progress made in chart question answering technology, there are still some shortcomings. Current research is primarily based on publicly available datasets (such as Figure QA and DVQA), which are generated by algorithms and exhibit high degrees of standardization and normalization. Additionally, chart question answering tasks are often viewed as classification problems, focusing primarily on simple information in charts with relatively limited question types and lacking complex reasoning problems.

Results

Dataset and experimental setup

In the experimental section of this chapter, three experiments are designed. Firstly, a comparison between CA-YOLOv5s and other one-stage object detection algorithms is conducted to intuitively verify the performance of CA-YOLOv5s on chart detection tasks, determining whether it can segment sub-charts within a complete data visualization dashboard, providing single chart data support for subsequent chart information extraction. Secondly, an ablation study is performed on the chart text extraction method, validating whether the proposed Extraction-OCR algorithm and Hourglass-OCR algorithm effectively improve the accuracy of chart text and chart numerical information extraction. Finally, a comparative experiment is conducted with the Revision model and the Chartsense model on nine chart types, comprehensively verifying the effectiveness of the proposed chart information extraction method for chart information extraction, providing strong support for practical applications. Through scientific and systematic experimental validation, the reliability and practicability of the proposed algorithms and methods can be ensured.

The dataset for chart data extraction consists of three datasets: the Figure QA dataset, the DVQA dataset, and the MECD dataset[13]. Among them, the Figure QA dataset and the DVQA dataset are publicly available datasets for chart question answering tasks. The Figure QA dataset includes five common chart types: line charts, dot plots, vertical bar charts, horizontal bar charts, and pie charts, providing rich data structures and visual patterns for the model. In contrast, the DVQA dataset only contains bar charts but provides stacked bar charts and bar charts with dark backgrounds that are not present in the Figure QA dataset. To better improve the accuracy of information extraction from individual charts in visual dashboards and subsequent chart question answering tasks, this paper expands the MECD dataset based on the needs of the project, including 19,020 charts from different manufacturing enterprises. To more comprehensively cover practical application scenarios, this paper further enhances the MECD dataset by capturing industry-standard charts and complete visual dashboards from platforms such as Datav and Power BI, thus enhancing the breadth and representativeness of the dataset. Additionally, 3,000 hand-drawn charts are collected by data collectors, further enriching the application scenarios and chart types for chart question answering and improving the diversity of the dataset. In total, there are nine chart types, including seven basic charts and one combined chart, specifically bar charts, bar graphs, line charts, scatter plots, pie charts, radial pie charts, rose charts, bar-line combination charts, and a group of visual dashboards.

Given the high similarity in features among commonly used chart types in dashboards, this paper divides the dataset into two categories for the chart segmentation experiments: single chart images and mixed images of visual dashboards. This division helps the segmentation model learn the image features of different chart types better, enabling it to complete the sub-chart segmentation task more effectively and reducing the impact of different chart types on the segmentation task, thereby improving the accuracy of the experiments. The chart data and visual dashboards in the manufacturing enterprise chart dataset (MECD) originate from real chart data generated during the daily production management of manufacturing enterprises, which significantly differs from the standard Figure QA dataset and DVQA dataset. The real enterprise dataset exhibits diversity, containing various chart features such as curved and tilted text content, as well as entire visual dashboards. The y-axis range of the Figure QA dataset and the DVQA dataset is uniformly divided into five groups from 1 to 100,

with an interval of 20. Although the DVQA dataset is more diverse in question-answering types compared to the Figure QA dataset, it consists solely of bar charts.

The experiments were conducted using the PyTorch deep learning framework, on a V-100 GPU processor, with an initial learning rate of 0.00001, and a batch size of 64. The model comprises three components: a 12-layer text-table encoder, a 12-layer visual transformer, and four multimodal cross-fusion reasoning layers. The model underwent 60 iterations of training on Figure QA and DVQA datasets, and 70 iterations on a real enterprise dataset, MECD.

Comparative experimental results

The evaluation criteria adopted in this paper's experiments include accuracy, recall, F1 score, AP (average precision per class), and mAP (mean average precision), where AP refers to the average precision for a single class and mAP is the mean of APs for all classes. The calculation methods are as shown in formulas 1–5.

$$Precision = \frac{true\ positives}{true\ positives + false\ positives} \tag{1}$$

$$Recall = \frac{true\ positives}{true\ positives + false\ negatives} \tag{2}$$

$$F1\ score = 2\frac{precision\ recall}{precision + recall} \tag{3}$$

$$AP = \int_0^1 p(R)\, dR \tag{4}$$

$$mAP = \frac{1}{n} \sum_{j}^{class\_number} Class\_AP_j \tag{5}$$

In the equations, true positives refer to the number of correctly detected positive samples, false positives refer to the number of negative samples that are wrongly detected as positive, and false negatives refer to the number of positive samples that are wrongly detected as negative. Precision is used to judge the accuracy of the model by calculating the ratio of correctly predicted positive samples to all predicted positive samples. Recall is obtained by calculating the ratio of correctly predicted positive samples to the total number of positive samples. In addition, F1 Score is another important indicator that combines precision and recall to evaluate the model's performance in the form of a harmonic mean. Furthermore, AP (average precision per class) is an evaluation metric for a single class, which combines information from precision and recall by calculating the area under the PR curve. AP serves as a standard to measure the performance of a classifier for a specific class. In multi-class object detection tasks, mAP (mean average precision), mentioned in formula (10), is the average of the AP values for all classes. It can comprehensively represent the overall performance of the detection model. By calculating the mean of AP values for each class, a comprehensive evaluation metric can be obtained to assess the model's overall performance across different classes.

Chart data extraction experiment sub-chart segmentation and recognition experiment

This chapter compares CA-YOLOv5s with current single-stage object detection algorithms such as CenterNet, RetinaNet, SSD, and the original YOLOv5s. It provides detailed detection accuracy values for eight different types of charts and the mean average precision (mAP) for each model, as shown in Table 1.

As can be seen from Table 1, the improved method CA-YOLOv5s achieves an AP value of 99.7% for bar graphs and pie charts, representing a 0.5–1.5% increase in accuracy compared to the YOLOv5s method. However, the recognition accuracy for bar charts is relatively lower, with an AP value of 96.7%. The recognition accuracy values for the seven basic chart types and one combined chart type are all above 90%, resulting in a final mAP value of 98.69%. This performance is superior to the CenterNet, RetinaNet, and SSD methods. Compared to the results of the YOLOv5s method, the average recognition accuracy of chart components in the CA-YOLOv5s method has increased from 97.31–98.69%.

	Bar chart	Line chart	Bar graph	Pie chart	Scatter plot	Radial pie chart	Rose diagram	Bar-line chart	mAP
CenterNet	85.0	94.0	95.8	97.0	94.3	94.6	89.3	94.6	93.08
RetinaNet	86.7	93.2	96.5	97.5	94.0	94.8	92.6	94.7	93.75
SSD	84.5	93.7	96.2	97.2	93.4	94.5	92.9	94.5	93.69
YOLOv5s	85.5	98.0	99.2	98.2	97.8	96.7	97.9	97.2	97.31
CA-YOLOv5s	96.7	99.2	99.7	99.7	98.4	98.6	98.9	98.3	98.69

Table 1. Comparison of our method with other approaches.

	Precision	Recall	F1 score
OCR	89.7%	88.6%	89.1%
Extraction-OCR	95.2%	94.4%	94.8%
OCR	87.7%	89.6%	88.6%
Hourglass-OCR	90.3%	91.4%	90.8%

Table 2. Ablation study on chart text recognition and numeric information extraction.

	Ours(Chart QA)			Revision			Chartsense		
Chart type	P	R	F1	P	R	F1	P	R	F1
Bar chart	94.70	92.70	93.69	75.60	84.08	79.61	82.30	86.10	84.16
H-bar chart	93.80	91.80	92.79	76.70	83.99	80.18	78.90	83.40	81.09
Line chart	94.30	91.30	92.79	67.40	77.19	71.96	74.30	79.60	76.86
Bar-line chart	89.70	92.40	91.03	60.20	72.05	65.59	67.90	73.40	70.54
Pie chart	92.30	91.10	91.70	64.30	75.46	69.43	79.70	81.30	80.49
Scatter plot	94.70	90.60	92.63	71.50	81.58	76.21	81.60	83.20	82.39
Radial pie chart	91.78	89.95	90.86	61.37	69.10	65.01	73.28	75.93	74.53
Rose diagram	86.21	87.38	86.76	54.74	56.89	55.68	67.83	69.87	68.75
Average value	92.23	91.06	91.68	65.80	75.03	69.80	76.30	79.60	77.60

Table 3. Performance comparison of chart data extraction.

Text recognition and numeric information extraction ablation experiments

During the phase of chart text extraction, this paper compares the proposed character-level text annotation algorithm Extraction-OCR and its enhanced version Hourglass-OCR, which incorporates a keypoint detection network, with traditional OCR techniques to verify the effectiveness of the improved models in extracting chart information. As shown in Table 2.

Based on the ablation study results in Table 2, the improvements made in this paper's model over the original OCR technology have effectively enhanced the accuracy of text extraction and numeric information extraction, providing data support for subsequent information reconstruction. However, the relative results for numeric information extraction are slightly poorer. The analysis indicates that the existence of overlapping key points in line and bar charts increases the difficulty of extracting numeric information.

Chart data extraction experiments

To demonstrate the effectiveness of the chart information extraction method proposed in this paper, we have conducted comparative experiments using the representative Revision[14] model and Chartsense model[15] proposed by previous scholars. The experimental results of each model are provided below, classified according to different chart types, as shown in Table 3.

Table 3 indicates that the chart data information extraction method proposed in this paper can successfully extract information from different charts, achieving an average accuracy of 92.23%. Compared with the Revision model, the average accuracy has improved by 26.43%, and compared with the Chartsense model, the average accuracy has increased by 15.93%. Specifically, the data extraction accuracy for bar charts and scatter plots has reached 94.70%. The excellent results of data extraction can be attributed to the more regular key points and fewer interfering factors in bar charts and scatter plots. For the combined chart of bar and line graphs, the recognition accuracy is only 91.7%, but there has been some improvement.

Chart QA experiment

To validate the effectiveness of the Chart QA model proposed in this paper for chart question answering tasks, we compare it with other previous works on chart question answering using open-source datasets Figure QA and DVQA, as well as a practical collaborative manufacturing enterprise dataset. Additionally, the validation model for the DVQA dataset uses an OCR version of the dataset. Table 4 displays the comparison of the experimental results on the FigureQA and DVQA datasets in terms of accuracy.

For the Figure QA dataset, it covers common chart types such as bar charts, line charts, pie charts, and line graphs. However, there are limitations on the types of user questions: 1) it only includes yes/no questions, and 2) it does not involve questions requiring numerical answers. Our model significantly outperforms the baseline model IMG+QUES, improving the overall accuracy by 31.96%. Compared to the baseline model SANDY (Oracle), the overall accuracy is increased by 29.35%. However, the accuracy is slightly lower than PReFIL. This is attributed to the fact that the FigureQA dataset mainly focuses on chart classification and simple yes/no questions, with relatively fewer numerical reasoning questions.

Regarding the DVQA dataset, it exhibits significant differences in appearance and style, capturing common styles found in scientific literature and the internet. Some of these differences include variations in bar charts

Models	FigureQA		DVQA (OCR)		MECD	
	Val1	Val2	Test-familiar	Test-novel	Val	test
IMG + QUES	59.41%	57.14%	32.01%	32.01%	22.38%	23.03%
SANDY (Oracle)	62.02%	59.54%	56.48%	56.62%	36.15%	38.02%
PReFIL	94.84%	93.26%	80.88%	80.04%	49.43%	51.46%
VL-T5	83.59%	82.38%	85.78%	84.47%	54.89%	56.31%
Vision TaPas	81.21%	89.74%	86.93%	86.77%	61.37%	62.05%
Pix2Struct	89.83%	88.62%	89.78%	90.13%	77.26%	76.95%
OURS (Chart QA)	91.37%	91.45%	94.43%	94.37%	80.56%	82.24%

Table 4. Chart QA experiment results.

and the number of groups, the presence or absence of grid lines, and differences in bar color, width, spacing, direction, and texture. The inclusion of questions related to numerical values and chart structure greatly enhances the practicality of chart question answering. However, the DVQA dataset is limited to bar charts, resulting in some deficiencies in chart type diversity.

On the DVQA dataset (divided into familiar and novel test sets), our model outperforms the baseline model IMG + QUES on the familiar test set, improving the overall accuracy by 62.42%. Compared to the baseline model SANDY (Oracle), the overall accuracy is increased by 37.95%. On the novel test set, our model also surpasses the baseline model IMG + QUES, improving the overall accuracy by 62.36%. Compared to the baseline model SANDY (Oracle), the overall accuracy is increased by 37.75%, representing a 3–7% improvement compared to recent models.

Based on the characteristics of the aforementioned datasets, after incorporating a certain amount of real collaborative manufacturing dataset MECD into the experiments, we found that the drawbacks of the baseline model and the PReFIL model became apparent with the addition of the new dataset and new reasoning question–answer pairs. Since traditional chart question answering models mostly focus on chart classification as the final result, lacking numerical reasoning ability, the accuracy drops significantly. However, after incorporating the real collaborative manufacturing dataset MECD into the public datasets, our proposed method achieves an accuracy of 82.24%.

The combined results of the four experiments indicate that the chartQA method proposed in this study has achieved a significant improvement in accuracy for chart question answering tasks. This improvement is primarily attributed to the decomposition of the chart question answering task, which utilizes a multi-stage processing flow to deeply explore chart information and precisely capture image features. Specifically, the method combines a character-level text annotation algorithm with a sandglass network to extract detailed information from charts, successfully converting unstructured chart data into structured table data. This step greatly improves the processing efficiency and accuracy of subsequent question answering tasks. Additionally, by introducing an aggregation operation head into the model, it endows the model with a certain level of computational ability, enabling it to handle more complex and diverse chart questions. This not only supplements and extends traditional chart question types but also provides users with more comprehensive and accurate answers.

Manufacturing industry application case

Collaborative manufacturing enterprise chart QA example

A network alliance of collaborative manufacturing companies has brought together multiple manufacturing enterprises, each with multiple workshops equipped with various types of CNC lathes, vertical (horizontal) machining centers, and other types of processing equipment. This networked collaborative manufacturing system integrates manufacturing information from various enterprises, monitoring and analyzing critical performance indicators of the manufacturing process within the alliance, such as capacity, production progress, and machining quality. This ensures that common production tasks are completed on time and meet quality standards.

Manufacturing information provided by different enterprises is mainly presented in the form of charts, and users can upload these charts. The system utilizes the chart QA model to parse the uploaded charts, directly extract key information, and provide answers to related questions. This significantly enhances the accessibility and interactivity of data in the collaborative manufacturing process between enterprises. This intelligent approach makes cooperation between enterprises more efficient. Users only need to ask questions about the charts, and the system can quickly parse the data content in the charts, providing users with accurate answers. Through this intuitive approach, users can obtain the desired numerical answers without accessing the database, helping data analysts make informed decisions in conjunction with background knowledge. The proposed Chart QA method is applied to a quality data integration and visualization analysis platform for manufacturing enterprises. Figure 2 illustrates the chart QA results generated based on user input for these charts in the manufacturing enterprise context.

The identified types of charts in this paper total six, as shown in Fig. 2, namely, bar chart, horizontal bar chart, line chart, scatter plot, pie chart, and a combined chart of bar and line. There are four types of user queries. The first type is related to visual trends, such as the question in Fig. 2E: "Which production material has the largest proportion in the pie chart?" Analyzing manufacturing data and raw material usage allows determining which

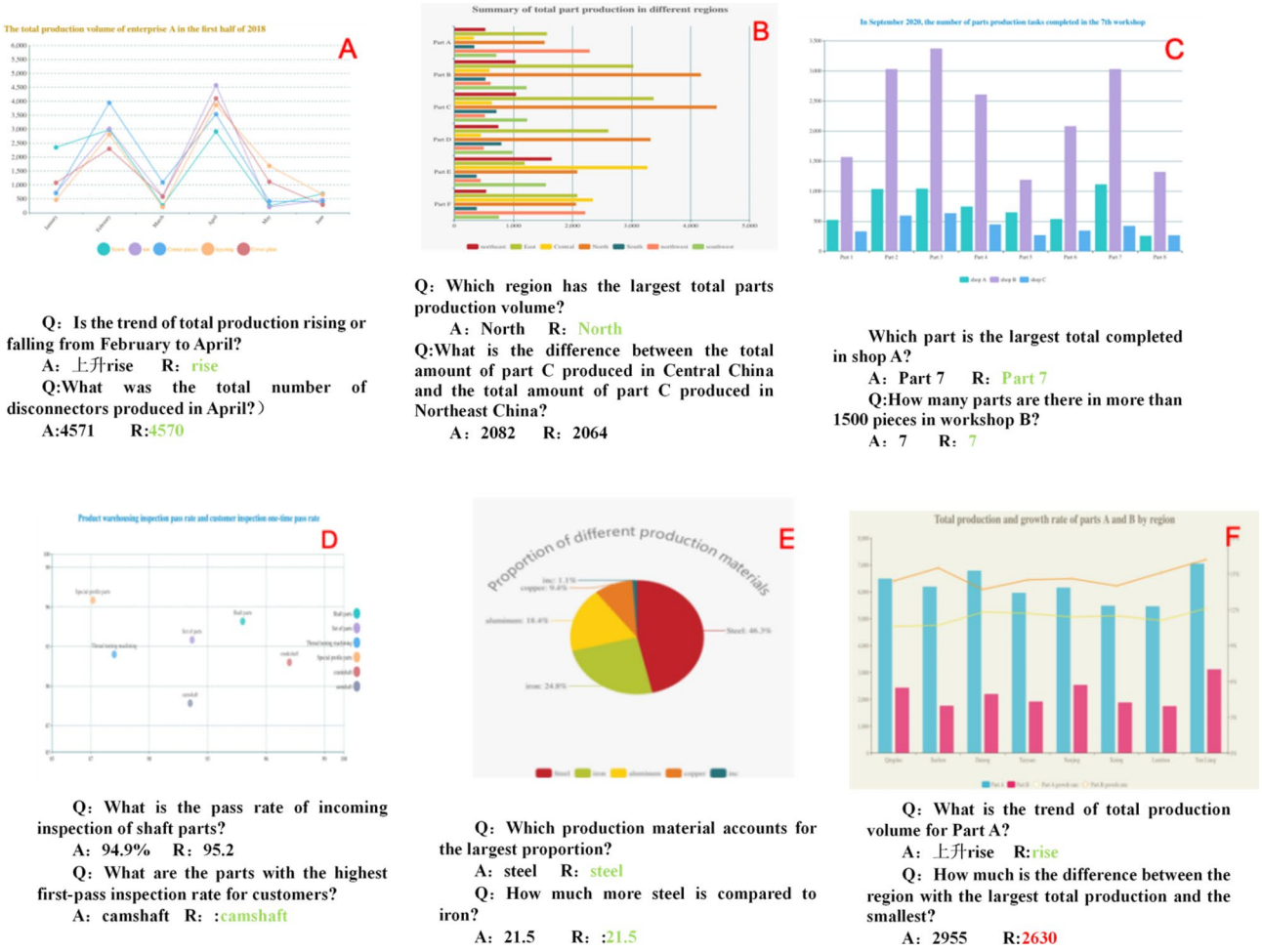


Fig. 2. Chart QA results for different types of charts.

raw material has the largest proportion. This helps companies understand the most critical raw materials in the production process, facilitating better supply chain and inventory management to ensure an adequate supply of raw materials and optimize procurement costs. The second type involves numerical extraction queries, like the question in Fig. 2A: "What is the total number of isolation switches produced in April?" If the production total in April deviates from the previous plan, companies can adjust manufacturing plans promptly to adapt to market changes or demand fluctuations. The third type pertains to structural composition queries, such as the question in Fig. 2C: "How many types of parts in Workshop B have production quantities exceeding 1500 units?" If one workshop exceeds the average planned production, redistributing production tasks to other workshops can distribute resources more evenly, avoiding overload in Workshop B and ensuring full utilization of other workshops. The fourth type involves aggregation and inference queries, for example, the question in Fig. 2F: "How much is the difference between the regions with the highest and lowest production volumes?" Differences in production volumes may reflect varying market demands in different regions. Companies can adjust sales and marketing strategies based on these differences to better meet regional demands. The numerical reasoning answers presented in Fig. 2, with an allowable 5% error compared to standard answers, are considered correct. Correct answers are indicated in green, while incorrect answers are shown in red. The analysis of errors indicates that the combination of line and bar charts makes it challenging to locate overlapping key points during key point extraction, leading to a lack of clear capture of chart feature information and causing deviations in numerical reasoning.

Discussion

Traditional chart interpretation methods often rely on manual efforts, which are inefficient and prone to errors. More crucially, due to the need for underlying data protection in enterprises, it is difficult to directly access the raw data through methods such as SQL queries, which limits the data support capabilities of multi-chart question answering systems. Therefore, the development of a system that can automatically understand and answer questions about charts has certain practical application value. This system focuses primarily on two core issues: first, how to accurately identify and extract textual and numerical information from charts without directly accessing the underlying data; second, how to accurately answer inferential questions that require aggregation

operations. To address these two issues, this article designs a chart question answering model based on multi-modal cross-fusion. The model realizes automated processing of chart data and question answering functionality through three stages: sub-chart segmentation and recognition, chart text annotation and information extraction, and answer generation, as shown in Fig. 3.

During the preparatory phase, the segmentation of data visualization dashboards is often performed manually in practical applications, but this method has a time cost issue. Dashboards are often composed of multiple sub-charts, each with different layouts, characteristics, and details, requiring careful identification of the boundaries of each sub-chart and precise cutting one by one during manual segmentation. This process is not only inefficient, especially when dealing with a large number of dashboards, resulting in a sharp increase in time cost, but also prone to errors, negatively impacting the accuracy of segmentation. To reduce time cost and improve segmentation efficiency, an automated segmentation method based on object detection technology, namely sub-chart recognition and segmentation, is introduced. The core goal of this phase is to accurately segment the entire dashboard image input by the user through the application of object detection technology. This step is based on the recognition of different image features on the dashboard and the analysis of spatial position information to achieve accurate segmentation and classification of sub-charts, providing a single-chart data basis for subsequent chart text annotation and data information extraction.

In the first phase, aiming at the problem of effectively recognizing irregular curved text in new types of charts, this paper focuses on capturing text and numerical information from charts. To achieve this goal, the following work has been mainly carried out: first, fine-grained annotation at the character level is performed on chart text to ensure accurate recognition of text regions; second, chart text extraction technology is applied to identify and extract key information from the chart, including but not limited to legends, axis labels, and chart titles; finally, a keypoint detection network is utilized to effectively capture keypoints associated with graphics and numerical values. These key pieces of information not only help the model analyze the distribution of numerical information in the chart, but also can further infer detailed numerical information of the chart by combining chart coordinates and legend information. After this series of processing, the obtained chart information is finally converted into structured tabular data, providing strong data support for the subsequent answer generation phase.

In the second phase of answer generation, a multi-modal cross-fusion approach is used to integrate chart features, text information, and generated table information, with the addition of an aggregated operation reasoning head with weighted information to answer questions involving data reasoning. This multi-modal information fusion method helps the question-answering system better understand the relationship between user

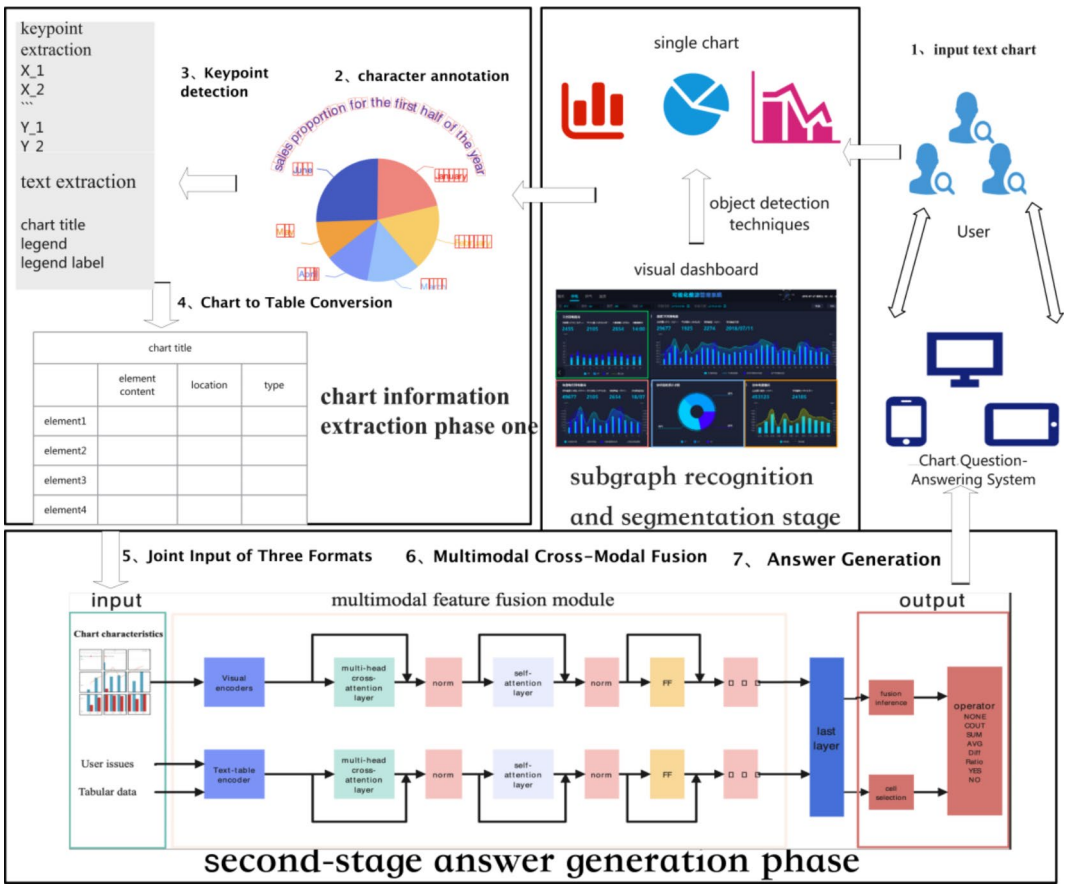


Fig. 3. The overall scheme of the chart Q&A.

questions and chart information, thus enhancing the comprehensive understanding and reasoning capabilities of the chart question-answering system and satisfying enterprises' needs for chart data reasoning analysis.

Methods

Chart information extraction

Chart text annotation

Text in charts carries crucial information, and precise localization of text is essential to ensure accurate recognition of all textual content in the charts. However, the diverse forms of text in charts, which may include tilt, curvature, and varying sizes, pose challenges for traditional text detection algorithms based on word-level annotations. These diversities make it difficult for these algorithms to precisely capture and locate text regions, thereby affecting the completeness of text content recognition in charts.To address these issues, this paper proposes a character-level text detection technique. By employing a character-level annotation method, this technique aims to improve the accuracy and completeness of chart text content extraction. Specifically, instead of treating text as an entire word, the technique decomposes text into character-level annotations to accommodate the diversity of text forms, thereby enhancing the accuracy of text content recognition. The introduction of this method can significantly improve the quality of chart text information extraction, providing strong support for in-depth understanding and analysis of chart data.

To convert the original word-level annotations into character-level annotations, this paper takes a series of steps. Firstly, the areas containing text lines are cropped from the original charts. Then,based on the text line-level label information, the cropped text lines are input into a character-level annotation network. In the training data,each word sample W is represented by R(w), with its bounding box denoted as BB (Bounding Box), and the length of the word is L(w). Based on this, the confidence of sample w is calculated using formula (6). In this formula, Sconf(w) represents the confidence of word w, which measures the accuracy of the character segmentation program in predicting the word's length. When the predicted length $l^c(w)$ is close to the true length l(w), the confidence approaches 1; conversely, when the difference between the two is large, the confidence decreases.

$$s_{conf}(w) = \frac{l(w) - \min(l(w), |l(w) - l^c(w)|)}{l(w)} \tag{6}$$

The pixel-level confidence S_C in the chart is calculated as shown in formula (7).If a pixel p is located within the bounding box of a word w, its confidence is equal to the confidence of that word,S_{conf}(w);otherwise, its confidence is set to 1.This confidence is used as a weight in the loss function,guiding the model to pay more attention to pixels within the word bounding box and with accurate predictions during training,thereby improving the model's accuracy.

$$S_c(p) = \begin{cases} s_{conf}(w) & p \in R(w) \\ 1 & \text{otherwise} \end{cases} \tag{7}$$

The loss function L is defined as in formula (8), which calculates the sum of squared Euclidean distances between the true scores and predicted scores for all pixels, using the pixel-level confidence Sc(p) as a weight. This way, the loss function can measure the overall prediction performance of the model and guide the model optimization during training.

$$L = \sum_p S_c(p) \cdot \left(||S_r(p) - S_r^*(p)||_2^2 + ||S_a(p) - S_a^*(p)||_2^2 \right) \tag{8}$$

where Sr*(p) represents the predicted center position score, Sa* represents the predicted inter-character affinity score, Sr represents the true center position score, and Sa(p) represents the true inter-character affinity score. When using character-level data for training, Sc(p) can be directly set to 1. Through the segmentation process, each individual character region obtained has corresponding training data labels. When training each image, character-level center position scores and affinity scores need to be generated to better determine the position information of text centers and the relationship between texts. The center position score represents the probability that a certain pixel is the center of a character, while the affinity score represents the probability of spatial relationship between adjacent characters. The loss function L can measure the overall prediction performance of the model and guide the model optimization during training. Formulas 6–8 are interrelated and together constitute the basis of model training for character segmentation and text detection tasks. By continuously optimizing the loss function L, the model can continuously improve its prediction performance, thereby more accurately detecting and recognizing text information from charts.

This paper employs Gaussian heatmap encoding to represent the probability of character centers. The specific process is illustrated in Fig. 4. Firstly, each character is precisely located by annotating a quadrilateral on the image. Then, by using the diagonals of the character-level annotated quadrilaterals, two triangles can be obtained. Constructing a bounding box connecting the two triangles' centroids represents the connection between two characters. Before generating labels for character-level annotations and connecting bounding boxes, a 2D Gaussian heatmap is created with a shape matching the corresponding character or connecting bounding box. The Gaussian heatmap, centered at each position, signifies the probability of that position being the center of the text, decreasing in probability as the distance from the center increases. Through perspective transformation, the

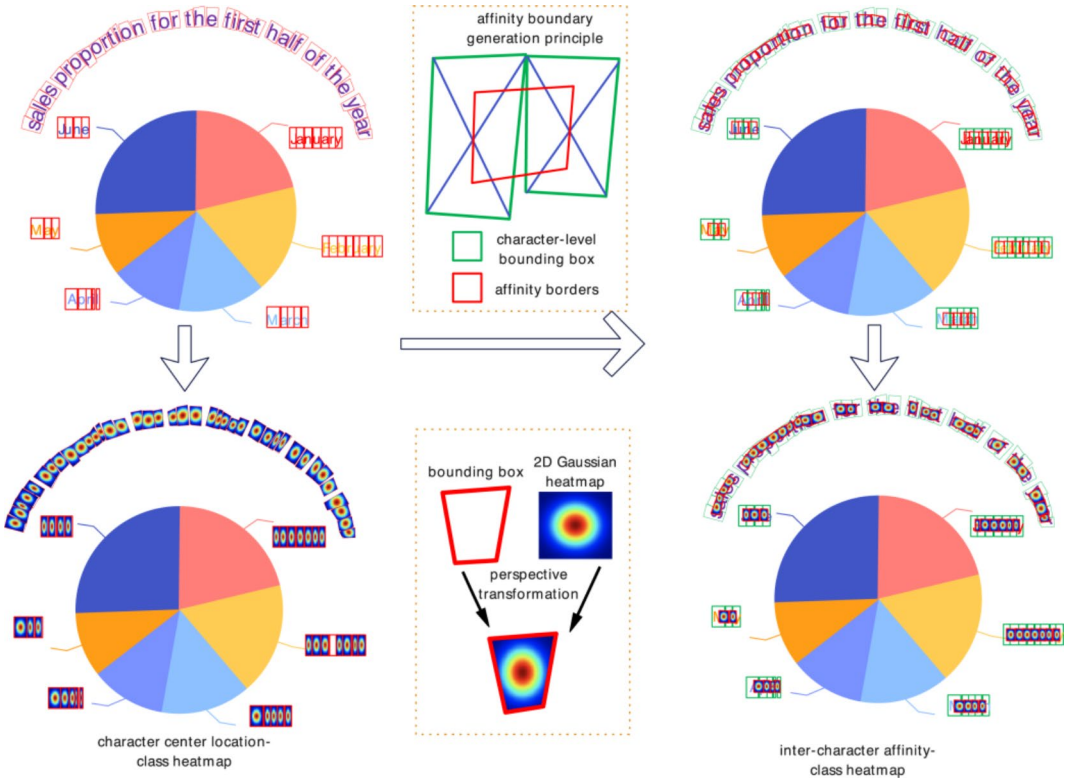


Fig. 4. Chart text character level callouts.

generated 2D Gaussian heatmap is mapped onto the shape of the corresponding character-level annotation or connecting bounding box, ensuring a perfect correspondence. Finally, the mapped Gaussian heatmap is pasted onto the respective background, forming labels for training data. This process provides probability information about the text center for each position in the image. This method resolves issues such as curvature and slanting that traditional text detection methods may struggle with in charts. During the text recognition phase, an OCR engine (Luo et al. 2021) is used to recognize the chart information annotated at the character level.

Chart to table transformation

To better facilitate the tasks of chart comprehension and numerical reasoning, it is necessary to convert unstructured charts into semi-structured tabular forms.This paper proposes a key point detection model to optimize the detection process of chart graphics. This method can detect multiple types of charts using a single model. After extracting the information of these key points, combined with data such as chart type, axes, and legends, the model can accurately extract numerical information from the chart. The numerical extraction process is mainly divided into three steps. The first step employs a corner network to extract key points. The second step extracts the numerical range, and the third step combines the chart text information obtained through OCR technology to determine the specific numerical value represented by each key point in the chart[16].

In the first step, a corner network is used to perform the extraction of key points. The hourglass network[17], as the core component of the corner network, effectively captures the location information of key points in the image through a series of downsampling and upsampling processes. The output of the hourglass network is a probability feature map, where pixels at key point locations are highlighted, providing powerful input for subsequent prediction modules. In the initial stage, the corner network is responsible for extracting key point information from the image. The core component of the corner network is the hourglass network, which adopts downsampling and upsampling steps to obtain the location information of key points in different charts. Through the upsampling and downsampling operations of the hourglass network, not only the key features of the image are captured, but also the accuracy of the network's positioning of key points is improved. The output of the hourglass network is a carefully compiled probability feature map, where the locations of key points are highlighted, providing powerful data support for subsequent prediction modules.

Next, these probability feature maps are passed to two key prediction modules: the top-left corner prediction module and the bottom-right corner prediction module. To locate the key points in the image, corner pooling operations are performed on these two prediction modules. Corner pooling operations, based on the implementation of max pooling, also focus on the boundary points of the chart, enabling precise capture of key points. After corner pooling, the prediction modules further refine the features using convolutional layers and generate three types of feature maps: heatmaps, embeddings, and offsets. These maps work together to provide necessary information for determining the precise locations of the top-left and bottom-right points. The heatmaps represent the probability distribution of each pixel point as a potential key point, the embeddings are

used to distinguish different key points, and the offsets are responsible for fine-tuning the locations of the key points.

The main responsibility of the heatmap is to predict the position information of the top-left and bottom-right points in the key point region. Its number of channels C is equivalent to the number of categories in the training set, reflecting the occurrence probability of various key points. The embedding map is used to pair the top-left and bottom-right key points of the same target, achieving accurate matching of key points by minimizing the distance between feature maps of the same key point group and increasing the distance between feature maps of different targets. Finally, the offset map is used to correct the position of the key points to ensure the accuracy of positioning.

The second step involves determining the data range of the chart, which is crucial for more effectively utilizing the detected key point positions from the image pixel space to interpret specific numerical relationships. This method is primarily used to process line charts, bar charts, and scatter plots. Given that the sum of all sectors in a pie chart defaults to 100%, the data range extraction step is omitted. For radial pie charts, the data range is not typically read directly from the chart; instead, the relative size of each classification can be inferred from the angular size and labels of the inner sectors. When extracting the data range, the focus is on identifying numbers related to the y-axis. To effectively isolate these y-axis labels, it is assumed that these numbers usually appear on the left side of the plot area.

Therefore, it is sufficient to locate the chart area and then easily filter out the y-axis labels based on their position. The determination of the chart area is also defined by the top-left and bottom-right corner points. The area of the chart can be precisely located following the key point detection process in the first step, or a more precise spatial positioning of the chart can be achieved using the CA-YOLOv5s model in the preparatory stage. Once the position of the chart area is determined and the OCR results are obtained, the data range estimation Table 5 is then used. The goal of this algorithm is to obtain the data range of the chart. In this algorithm, the plot area is first identified using the detected corner points, and then the recognized numbers located on the left side of the plot area are found. Finally, the data range and pixel range are calculated using the top and bottom numbers to map the key points to actual data values. This process aims to achieve automatic extraction and interpretation of chart data in the context of computer vision and text analysis. Inside the function, the nearest number candidates rmin to the bottom-left corner and rmax to the top-left corner of the plot area are first found through OCR results. Then, the numbers of rmin and rmax are extracted and stored in rmin_num and rmax_num. By calculating the Y-axis scale Yscale, which represents the proportion of values in the pixel space, the Y-axis range is finally calculated using Yscale, the bottom, top, and the position information of the candidate points. Combining Yscale with the chart's textual information and key point positions, the numerical information of different key points in the chart is automatically estimated.

Through data extraction techniques, it is possible to obtain complete chart data that includes basic chart information and numerical information. As different chart types have their own characteristics, the methods of data extraction and key point matching will also vary. Taking a bar chart as an example, it is commonly used in manufacturing to showcase the production output or sales figures of different product lines. The primary focus is on extracting corner point information, which helps the model accurately determine the position of each bar, and the height of the bar directly corresponds to the production output or sales figures of each product line.

Looking at line charts, they are often used in manufacturing to demonstrate changes in production volume or quality indicators of a product over time. The key points of a line chart are mainly the turning points of each line, and the positions of these turning points reflect the changing trends and specific values of production or quality indicators. Scatter plots are also common in manufacturing, and they are used to analyze the relationship between different variables. The key points of a scatter plot are the individual points on the chart, and the positions of these points directly correspond to the specific values in the dataset.

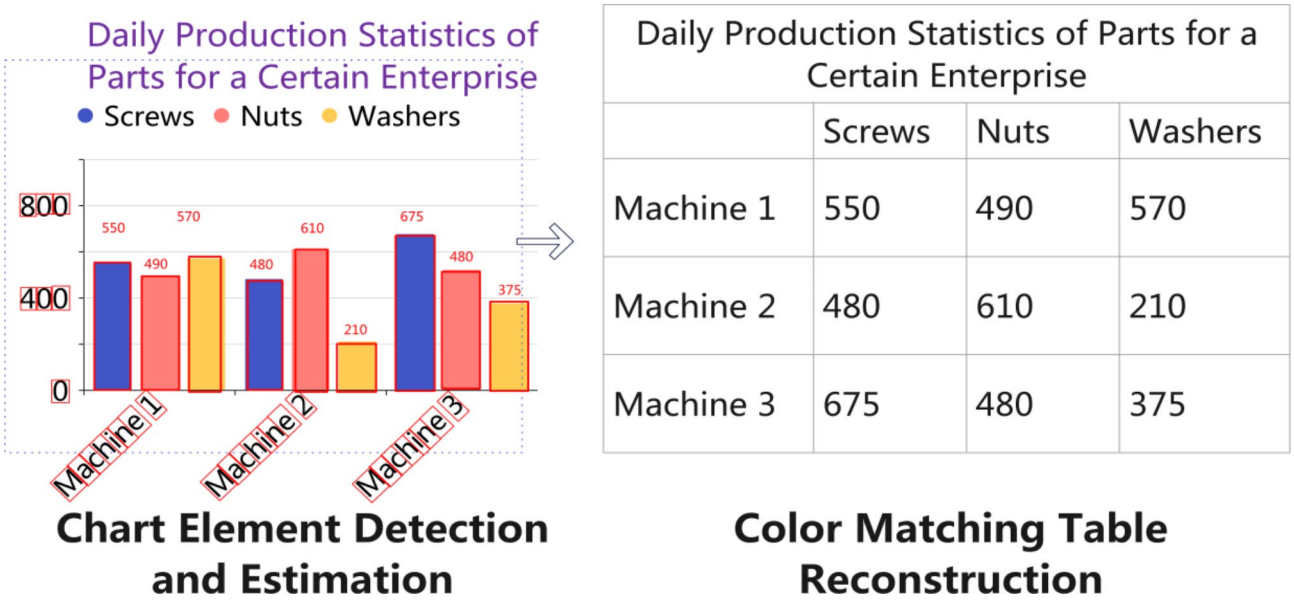
By extracting graphical information in the above-mentioned ways and integrating it with textual information, complete data related to the chart can be obtained. Rose plots (or polar bar charts) display numerical values for multiple categories in polar coordinates, where each category is usually represented by one or more bar lines. The endpoint of these lines extending from the center can be captured, and their length analyzed to represent the numerical value.

The model needs to associate the extracted data values with text labels for table construction. When colors are used to distinguish between data and text label relationships, first, the system utilizes Open CV image processing techniques, such as color space conversion and color threshold segmentation, to accurately identify the colors of different elements in the chart. For chart types like bar charts and line charts, the system pays special attention to the colors of bars or lines and converts these colors into RGB values for direct numerical comparison. Then, the system turns to the legend area, analyzes the text labels in depth, and determines the colors associated with each

Algorithm 1: Value Estimation

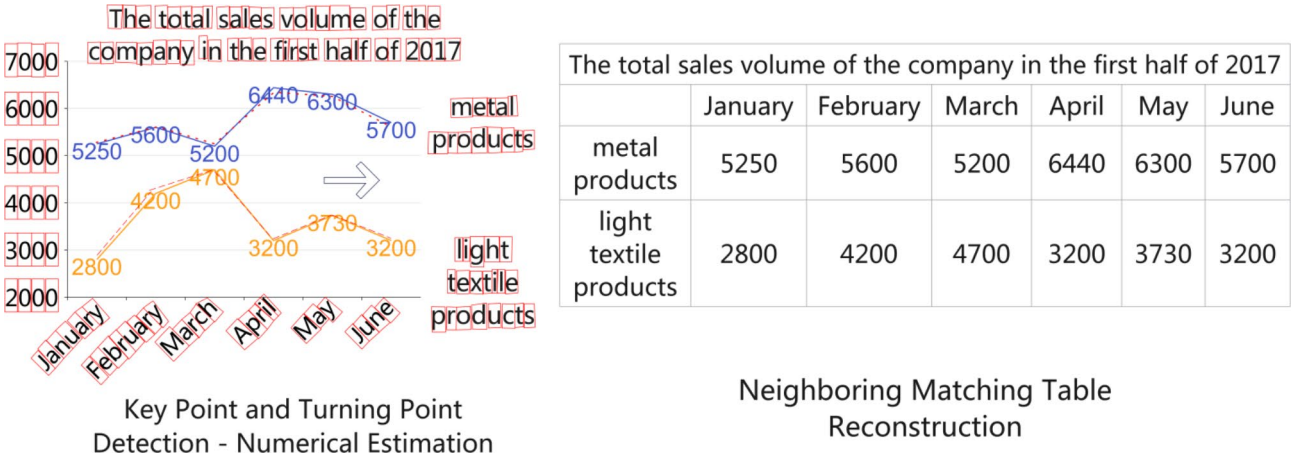
Input: Chart Area Coordinates: Top, Left, Bottom, OCR Recognition Results R
Output: Y-axis Scale Information Yscale,
rmin, rmax ∈ R // These two variables will store the minimum and maximum label objects near the y-axis from the OCR recognition results
y1, y2 ∈ R // These two variables will store the y-coordinates of rmin and rmax in the image
rmin_num = number(rmin_text) // Store the extracted numerical data from rmin into rmin_num
rmax_num = number(rmax_text) // Store the extracted numerical data from rmax into rmax_num
Yscale = (rmax_num - rmin_num) / (y2 - y1) // Calculate the Y-axis scale information Yscale

Table 5. Data range estimation table.



Daily Production Statistics of Parts for a Certain Enterprise			
	Screws	Nuts	Washers
Machine 1	550	490	570
Machine 2	480	610	210
Machine 3	675	480	375

Fig. 5. Principle of chart color matching.



The total sales volume of the company in the first half of 2017						
	January	February	March	April	May	June
metal products	5250	5600	5200	6440	6300	5700
light textile products	2800	4200	4700	3200	3730	3200

Fig. 6. The principle of proximity.

label. This step typically involves detecting color blocks or lines next to legend labels and converting their colors to RGB values for subsequent numerical comparisons.

The system performs precise color matching operations. By comparing the colors of data elements (such as bars in a bar chart or lines in a line chart) with the colors in the legend label area, the system can precisely determine which data elements are associated with which legend labels based on the similarity or difference in RGB values. For example, if a bar chart has red and blue bars, as well as red and blue legend labels, by examining the colors of the bars, the system can associate the values of the red bars with the red labels and the values of the blue bars with the blue labels, as shown in Fig. 5. However, when all markers have the same color or there is no clear color association, and when the relative positions of labels or markers and corresponding data values are clear, a proximity-based heuristic method is used. This method matches data values and labels by measuring their relative positions. For instance, in a line chart, if a label is located next to a certain line, the system can associate that label with the value of the adjacent line, as shown in Fig. 6.

Chart question answering

We have obtained textual content and keypoint information from the chart, and transformed this information into tabular data, providing the foundational information for subsequent chart question answering tasks. However, having this information alone is not sufficient to complete chart question answering tasks. Since chart question answering tasks require an understanding of the chart's structure and data relationships, it is necessary to infer accurate answers by integrating tabular information with image features. This section will explore how to

effectively combine the reconstructed tabular information with chart features to achieve better inference results for chart question answering tasks.

Sub chart segmentation algorithm

The primary objective of the chart question answering task is to obtain the corresponding answers given a chart and related textual questions through an algorithm. It has been observed that in enterprises, alongside abundant individual chart data, there are cases involving hand-drawn charts and visual dashboards[18] Dashboards often contain numerous sub-charts, and it is necessary to filter out sub-charts that meet user requirements to reduce unnecessary redundant information for subsequent chart visual encoding. This paper introduces a Spatial and Coordinate-based Cooperative Attention Mechanism[19] (Coordinate Attention, CA) on the basis of the YOLO network[20]. By capturing cross-channel feature information, direction perception, and position-awareness information, this mechanism allocates weight relationships among different pixels in the spatial domain. It enhances the focus on features relevant to the user's desired chart, reduces interference from irrelevant features, improves the attention and precise positioning capabilities of targets, and also addresses the recognition and classification issues of hand-drawn charts. Using the aforementioned sub-chart positioning algorithm, the dataset's dashboard data types were segmented based on different chart positions, annotated through titles on different sub-charts, and achieved the functionality of segmenting the entire dashboard into individual sub-charts.

For individual sub-charts obtained through the YOLO object detection algorithm, this paper replaces traditional convolutional neural networks with the Vision Transformer (ViT)[21]. The Vision Transformer divides the image into 2D image blocks and employs the self-attention mechanism of the Transformer[22] to capture global relationships between image blocks. This allows the Vision Transformer to better understand the overall structure and contextual information of the chart, demonstrating stronger reasoning capabilities in chart question answering tasks. For image classification tasks, the Vision Transformer introduces a special token, commonly referred to as the [CLS] token, in the output sequence. This [CLS] token generates a representation in the last layer of the Transformer encoder, representing the features of the entire image.

As mentioned earlier, to better understand the overall structure of the chart, the chart was segmented. The relative spatial positions of segmented chart blocks typically carry crucial information. For instance, in a bar chart, the relative positions and heights of different bars, or in a line chart, the positions of turning points, are essential for understanding the meaning of the chart and answering questions. Therefore, this paper adds a position embedding module to endow the model with the ability to perceive the relative spatial positions of segmented chart blocks. By learning the relative positions of different elements in the chart, the model gains a more accurate understanding of the spatial relationships between various elements. Through pre-training on a large-scale image dataset, the Vision Transformer learns universal image representations with excellent feature transferability. In chart question answering tasks, the pre-trained Vision Transformer model can be used as a feature extractor. The chart is input into the Vision Transformer to obtain visual feature encoding. These rich visual features help users better understand the chart structure, leading to superior results in chart question answering tasks.

Text table feature encoding

This section focuses on how to effectively encode the reconstructed tabular data with textual questions and perform aggregate operations on the table data. To better understand the relationship between table information and user queries, the table is flattened and concatenated with the question, as illustrated in Fig. 6.

In Fig. 7, token embeddings capture the semantic information of each input token (word, character, etc.) to enable the model to understand and compare different tokens in a vector space. Positional embeddings indicate if the answer to the data is related to the previous data, preserving contextual information. Paragraph embeddings serve to distinguish between questions and tables by concatenating the question and relevant table, with the question marked as 0 and the table marked as 1. The formation of paragraph embeddings is a gradual refinement process.During the input preprocessing stage of the model, questions and tables are labeled as different segments and assigned unique identifiers. These identifiers are then converted into fixed-length embedding vectors in the embedding layer of the model. These embedding vectors are randomly initialized at the start of model training

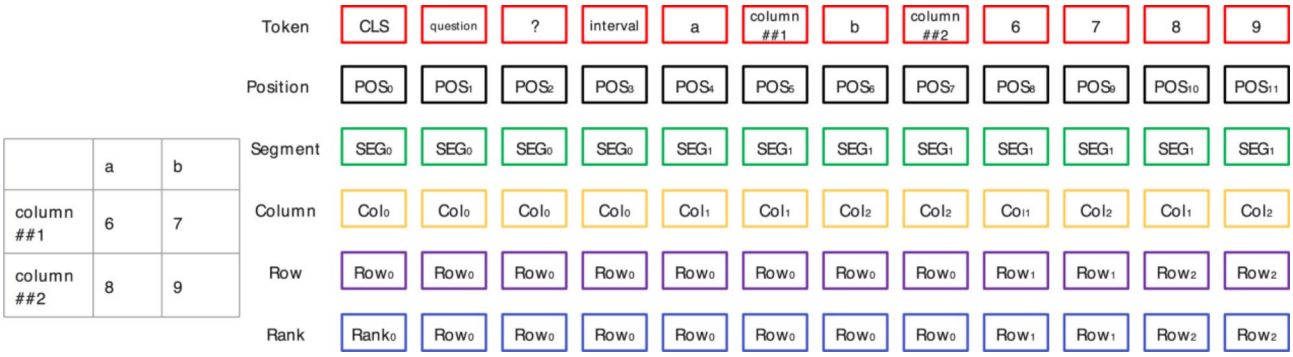


Fig. 7. Text table expansion encoding schematic.

but gradually learn semantic information to distinguish different segments (such as questions and tables) as the model iterates and optimizes on the training data. Through backpropagation and parameter updates, the embedding vectors are continuously adjusted until they can accurately capture and express the semantic features of the input segments.

Column embeddings enable the model to understand the semantic meanings of different columns in the table. Row embeddings help the model comprehend different rows of the table. The formation of order embeddings focuses on handling sortable data in the table. During preprocessing, the model first identifies sortable columns in the table and assigns a unique sort identifier to each cell in these columns. These identifiers reflect the sort position of the cell within a specific column. In the embedding layer, these sort identifiers are converted into embedding vectors, which are also randomly initialized at the start of training. However, as the model continues to learn from the training data, these embedding vectors gradually learn to distinguish between different sort positions. Through backpropagation and parameter adjustments, order embedding vectors are continuously optimized to help the model understand and compare sort relationships between cells, especially for sortable data such as numbers and dates. This allows the model to reason based on sort relationships, such as using natural numbers 1-n for numbers and dates, with 1 representing the smallest value and subsequent numbers representing increasing order, while other non-sortable types are assigned 0. These additional embeddings refine the representation of tokens, enabling the model to better utilize context, deepen its understanding of dialogue structure and relationships between tokens, and thus enhance the system's ability to capture complex dialogue dynamics and information interactions.

The above methods transform table data into a format similar to natural language sentences and learn the structural and semantic information of table data through pre-trained models. This enables the model to understand the relationship between questions and table content and accurately find the relevant row labels as answers. For question-answering models with table data, the answers to questions often come from one or more table cells or subsets of table cells. Additionally, for scalar answers, the results may be derived from aggregation operations on multiple table cells[23]. Therefore, based on the above considerations, two heads are designed: one for selecting the involved table cells and another for selecting the aggregation operator to be applied to these table cells, as shown in Fig. 8.

This article extends the BERT model to enable aggregation operations[24]. If the corresponding answer exists in the table, the model first selects a column from the table and then a cell from the selected column to find the answer. However, when the question requires a scalar value that is not directly present in the table, for example, when calculating the sum or average of a particular column, an aggregation operation is necessary. This article utilizes the probability of each token being selected and the table values to estimate the probabilities of each aggregation operator. Finally, the model's expected result is given by using the results of all aggregation operations, as calculated by the following formula (9). In this formula, S_{pred} represents the final scalar value predicted by the model, which is the result of the summary or aggregation of table data. The index variable I iterates over all possible aggregation operators OP_i. $(\hat{p}_a(op_i))$ is the probability predicted by the model for the selection of the i-th aggregation operator opi. This probability reflects the model's judgment on which operator is most suitable for the current task. $(compute(op_i, p_s, T))$ is the result of applying the aggregation operator opi to the selected column ps in the table T. This calculation process is based on the selected operator and column data, producing a scalar value as the output.

$$S_{pred} = \sum_{i=1} \hat{p}_a(op_i)\ compute(op_i, p_s, T) \tag{9}$$

Here, Jscalar represents the value of the Huber loss function, which quantifies the discrepancy between the model's predicted value (a) and the true value. The variable a stands for the difference or error between the model's predicted value and the true value. δ is a hyperparameter in the Huber loss function that controls the switching point from squared loss to absolute loss. By adjusting the value of δ, one can strike a balance between the sensitivity of the loss function to outliers and its overall performance.

$$\mathcal{J}_{\text{scalar}} = \begin{cases} 0.5 \cdot \text{a}^2 & \text{a} \leq \delta \\ \delta \cdot \text{a} - 0.5 \cdot \delta^2 & \text{otherwise} \end{cases} \tag{10}$$

By undertaking these two pivotal steps, the model can proficiently identify pertinent cells within the table, predicting the appropriate aggregation operator. This achieves semantic parsing of tabular data, enhancing the model's understanding and processing capabilities.

Multimodal feature encoding

To fully utilize information from images, text-tables, and questions, a multimodal feature fusion module[25] is employed. This fusion module consists of four blocks, each containing a visual branch and a text-table branch. Specifically, the image is first fed into a visual encoder ViT, which transforms it into a series of embedding vectors H = {h_{Lcls}, h_{L1}, …, h_{L8}}, where L represents the number of layers in ViT, h_{Lcls} is the embedding vector for the special [CLS] token, and h_{L1} to h_{L8} correspond to the embedding vectors for each cell in the table. Concurrently, the question text and data table are input into a table-text encoder, which encodes the question and table content into a series of embedding vectors Z = {z_{Lcls}, z_{L1}, …, z_{Lm}}. Here, L also denotes the number of layers, m is the total number of tokens in the question and table, z_{Lcls} is the embedding vector for the special [CLS] token of the question, and z_{L1} to z_{Lm} represent the embedding vectors for each token in the table and question, as illustrated in Fig. 9.

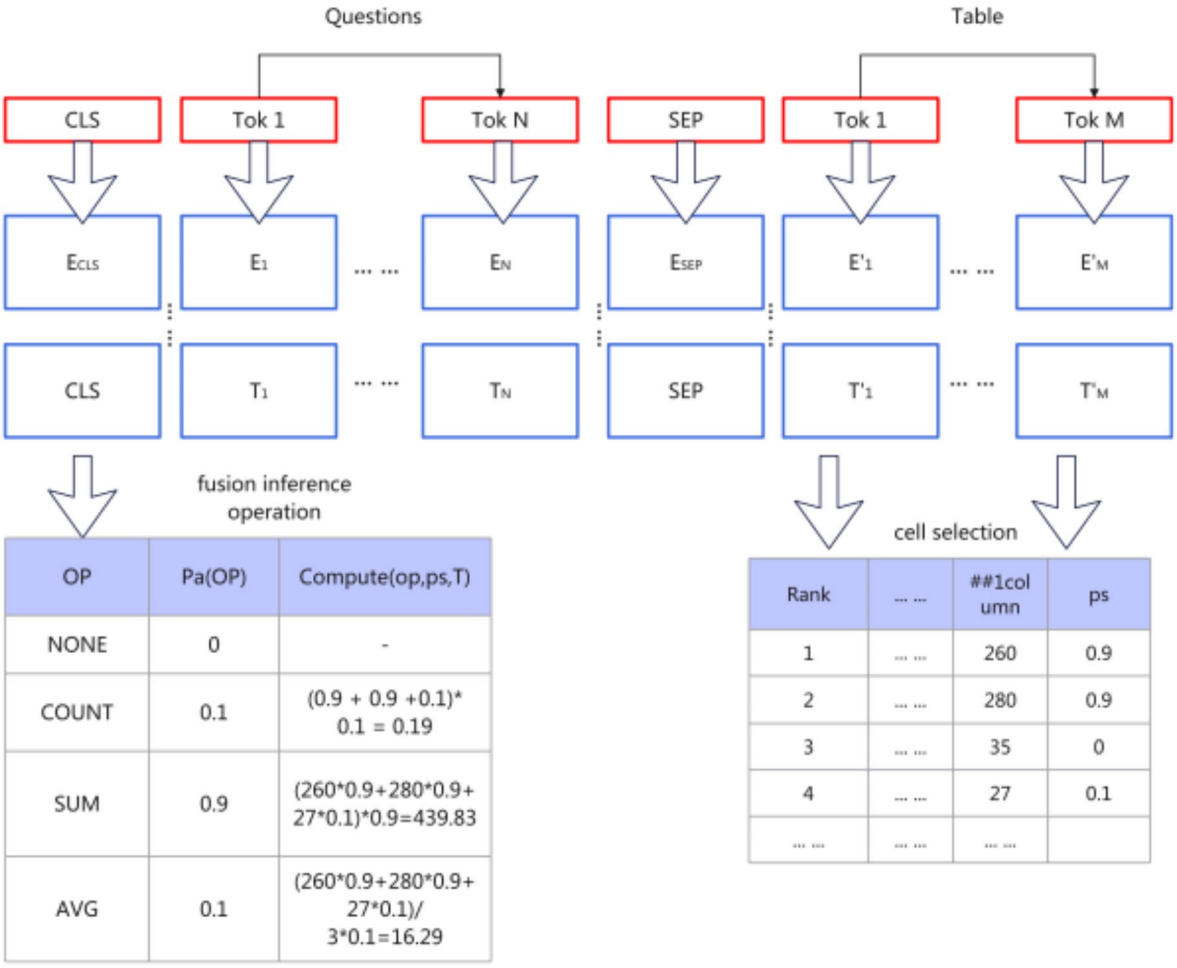


Fig. 8. How a fusion inference head with weighted values works.

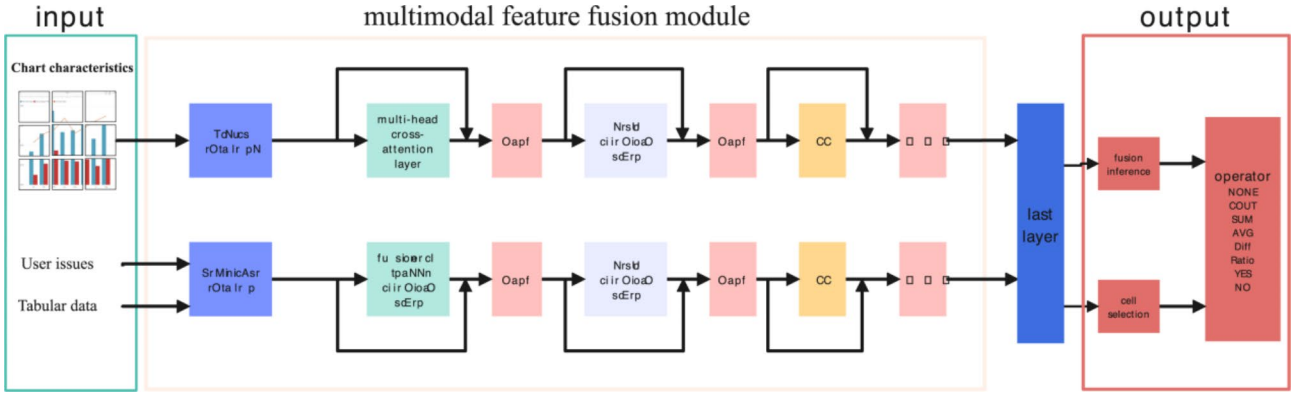


Fig. 9. The input of the multimodal feature fusion module consists of visual encoding and text-table encoding.

Subsequently, these encoded vectors H and Z are fed into a cross-modal encoder. The cross-modal encoder comprises four blocks, each containing a visual branch and a text-table branch. In the first block, the visual branch utilizes the visual features H as query vectors, while the text-table branch employs the text-table features Z as key and context vectors. This design allows the visual branch to focus and fuse information from the text-table branch, enhancing the model's understanding of the relationship between the question and the chart. In the second block, the roles of the two branches are reversed, with the text-table branch using Z as query vectors and the visual branch using H as key and context vectors, enabling bidirectional interaction and fusion. To fully interact visual and text-table features, this process is repeated for four blocks. Within each block, cross-features are first generated through a multi-head cross-attention layer. These cross-features are then processed by a self-attention layer and a fully connected layer to capture complex inter-feature relationships[26]. Residual connections are applied in each block, ensuring smooth gradient propagation during information transmission, which helps mitigate the gradient vanishing problem in deep neural networks. Additionally, layer normalization is used to normalize the output of each block, improving model stability and training efficiency. In the final layer of the text-table branch, aggregation operations and a cell selection head are appended. These operations further process the text-table data to prepare for outputting the final answer.

Data availability

Our research utilizes three datasets to validate the effectiveness of the proposed methodology. Firstly, we employ two publicly available datasets from the domain of chart-based question answering tasks: the FigureQA dataset, accessible at https://www.microsoft.com/en-us/research/project/figureqa-dataset/download/, and the DVQA dataset, accessible at https://github.com/kushalkafle/DVQA_dataset/blob/master/LICENCE.However, this arti cle has made available a portion of the declassified data used at https://github.com/wangxinxin-linan/MECD for everyone to use. Additionally, we incorporate a real manufacturing enterprise chart dataset named MECD. Due to the inclusion of extensive real data pertaining to enterprise production, manufacturing, and sales, these datasets cannot be made publicly available. Therefore, the datasets generated and/or analyzed during the current study are not publicly accessible. However, they can be obtained from the corresponding author upon reasonable request.

Received: 11 February 2024; Accepted: 16 December 2024
Published online: 06 January 2025

References

1. Xiong, Y., Tang, Y. L., Kim, S. & Rosen, D. W. Human-machine collaborative additive manufacturing. *J. Manuf. Syst.* **66**, 82–91. https://doi.org/10.1016/j.jmsy.2022.12.004 (2023).
2. Kalamaras, I. et al. An interactive visual analytics platform for smart intelligent transportation systems management. *IEEE Trans. Intell. Transp. Syst.* **19**, 487–496. https://doi.org/10.1109/tits.2017.2727143 (2018).
3. Obeid, J. & Hoque, E. Chart-to-text: Generating natural language descriptions for charts by adapting the transformer model. In *13th International Conference on Natural Language Generation, INLG 2020, December 15, 2020 - December 18, 2020* 138–147 (Virtual, Association for Computational Linguistics (ACL), 2020). https://doi.org/10.48550/arXiv.2010.09142.
4. Tian, X., Lin, Y., Song, M., Bao, S., Wang, F., He, H., Sun, S. & Wu, H. Q-TOD: A query-driven task-oriented dialogue system. In *2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, December 7, 2022 - December 11, 2022* 7260–7271 (Association for Computational Linguistics (ACL), 2022). https://doi.org/10.18653/v1/2022.emnlp-main.489.
5. Pal, A., Chang, S. & Konstan, J. A. Evolution of experts in question answering communities. In 6th *International AAAI Conference on Weblogs and Social Media, ICWSM 2012, June 4, 2012 - June 7, 2012* 274–281 (AAAI Press, 2012). https://doi.org/10.1609/icwsm.v6i1.14262.
6. Duan, J. S., Zhang, P. F., Qiu, R. H. & Huang, Z. Long short-term enhanced memory for sequential recommendation. *World Wide Web* **26**, 561–583. https://doi.org/10.1007/s11280-022-01056-9 (2023).
7. Kahou, S. E., Michalski, V., Atkinson, A., Kadar, a., Trischler, A. & Bengio, Y. Figureqa: An annotated figure dataset for visual reasoning. arXiv. https://doi.org/10.48550/arXiv.1710.07300 (2017).
8. Kafle, K., Price, B., Cohen, S. & Kanan, C. (2018). DVQA: Understanding data visualizations via question answering. In *31st Meeting of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2018, June 18, 2018 - June 22, 2018* 5648–5656 (IEEE Computer Society, 2018). https://doi.org/10.1109/cvpr.2018.00592.
9. Kafle, K., Shrestha, R., Price, B., Cohen, S. & Kanan, C. Answering questions about data visualizations using efficient bimodal fusion. In *2020 IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2020, March 1, 2020 - March 5, 2020* 1487–1496 (Institute of Electrical and Electronics Engineers Inc., 2020). https://doi.org/10.1109/wacv45572.2020.9093494.
10. Lee, K., Joshi, M., Turc, I. R. et al. Pix2struct: Screenshot parsing as pretraining for visual language understanding. In *International Conference on Machine Learning* 18893–18912 (PMLR, 2023).
11. Cho, J., Lei, J., Tan, H. et al. Unifying vision-and-language tasks via text generation. In *International Conference on Machine Learning* 1931–1942 (PMLR, 2021).
12. Dong, H., Wang, H., Zhou, A. et al. TTC-QuAli: A text-table-chart dataset for multimodal quantity alignment. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining* 181–189 (2024).
13. Chen, L. & Zhao, K. An approach for chart description generation in cyber–physical–social system. *Symmetry* **13**(9), 1552 (2021).
14. Savva, M., Kong, N., Chhajta, A., Fei-Fei, L., Agrawala, M. & Heer, J. Revision: Automated classification, analysis and redesign of chart images. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology* 393–402 (2011).
15. Jung, D., Kim, W., Song, H., Hwang, J.-i., Lee, B., Kim, B. & Seo, J. Chartsense: Interactive data extraction from chart images. In *Proceedings of the 2017 Chi Conference on Human Factors in Computing Systems* 6706–6717. https://doi.org/10.1145/3025453.3025957 (2017).
16. Luo, J., Li, Z., Wang, J. & Lin, C.-Y. ChartOCR: Data extraction from charts images via a deep hybrid framework. In *2021 IEEE Winter Conference on Applications of Computer Vision, WACV 2021, January 5, 2021 - January 9, 2021* 1916–1924 (Virtual, Institute of Electrical and Electronics Engineers Inc., 2021). https://doi.org/10.1109/wacv48630.2021.00196.
17. Xu, T. & Takano, W. Graph stacked hourglass networks for 3D human pose estimation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2021, June 19, 2021 - June 25, 2021* 16100–16109 (Virtual, IEEE Computer Society, 2021). https://doi.org/10.1109/cvpr46437.2021.01584.
18. Ma, R. et al. Ladv: Deep learning assisted authoring of dashboard visualizations from images and sketches. *IEEE Trans. Vis. Comput. Graph.* **27**(9), 3717–3732. https://doi.org/10.1109/tvcg.2020.2980227 (2020).

19. Redmon, J., Divvala, S., Girshick, R. & Farhadi, A. You only look once: Unified, real-time object detection. In *29th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, June 26, 2016 - July 1, 2016* 779–788 (IEEE Computer Society, 2016). https://doi.org/10.1109/cvpr.2016.91.
20. Hou, Q., Zhou, D. & Feng, J. Coordinate attention for efficient mobile network design. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2021, June 19, 2021 - June 25, 2021* 13708–13717 (Virtual, IEEE Computer Society, 2021). https://doi.org/10.48550/arXiv.2103.02907.
21. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J. & Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929. (2020).
22. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. & Polosukhin, I. J. A. Attention is all you need. **30**. https://doi.org/10.48550/arXiv.1706.03762 (2017).
23. Sun, N., Yang, X. & Liu, Y. TableQA: a Large-Scale Chinese Text-to-SQL Dataset for Table-Aware SQL Generation. arXiv. https://doi.org/10.1145/2047196.2047247 (2020).
24. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. J. Bert: Pre-training of deep bidirectional transformers for language understanding. https://doi.org/10.48550/arXiv.1810.04805 (2018).
25. Tan, H. & Bansal, M. J. Lxmert: Learning cross-modality encoder representations from transformers. https://doi.org/10.18653/v1/D19-1514 (2019).
26. Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan, F. S. & Shah, M. J. A. Transformers in vision: A survey. **54**, 1–41. https://doi.org/10.1145/3505244 (2022).

Acknowledgements

This work was supported by the National Natural Science Foundation of China (51675108) and Key Scientific Research Program of the Education Department of Shaanxi Province, China. (22JS021).

Author contributions

In this study, W.X. was primarily responsible for the overall implementation and design of the chart-based question answering method, as well as collecting and organizing real-world enterprise datasets to validate the effectiveness of the chart-based question answering task. C.L. oversaw the overall design of the method and provided guidance and support in the direction of innovation. L.C. was responsible for providing real-world chart data from enterprises and validating the method through practical enterprise application scenarios. These three authors played crucial roles in the study, collectively driving the progress and outcomes of the research. L.J. organized the datasets and question-answer pairs for this article.

Declarations

Competing interests

The authors declare no competing interests.

Ethical approval

The issue of inaccurate recognition of curved text in novel charts has been effectively addressed through the application of character-level text annotation methods.
The visual dashboard underwent segmentation and organization using an object detection algorithm. This step was taken to provide chart support for subsequent multi-table cross-referencing. The enhancement of mathematical computational capabilities in chart-based question and answer processes was achieved by introducing an inference fusion head. This additional component significantly contributes to the system's ability to perform intelligent analysis and respond to queries.

Additional information

Correspondence and requests for materials should be addressed to X.W.