IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 12 May 2023, accepted 13 July 2023, date of publication 24 July 2023, date of current version 27 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3298050

SURVEY

Automatic Chart Understanding: A Review

ALI MAZRAEH FARAHANI[1], PEYMAN ADIBI[1], MOHAMMAD SAEED EHSANI[1], HANS-PETER HUTTER[2], AND ALIREZA DARVISHY[2]

[1]Artificial Intelligence Department, Faculty of Computer Engineering, University of Isfahan, Isfahan 81746-73441, Iran
[2]School of Computer Science, Zurich University of Applied Sciences (ZHAW), Winterthur 8401, Switzerland

Corresponding author: Alireza Darvishy (dvya@zhaw.ch)

This work was supported by the funding provided by School of Engineering, Zurich University of Applied Sciences.

ABSTRACT Automated chart analysis has vast potential to improve the accessibility of charts for a wider audience, e.g., people with visual impairments or other disabilities, by generating captions for chart images that can quickly convey the information being represented. Additionally, it can improve the performance of automatic document analysis systems, by enabling them to extract valuable information from the documents with graphical/visual scientific content. Although recent advancements in modality translation and multi-modal learning have led to the development of more or less successful image captioning and visual question answering methods, but most of them have been designed for general images, and cannot be successfully applied to specific areas such as medical images or scientific charts and graphs. Therefore, further research is necessary to develop automated chart analysis methods that can be effectively applied to these specific areas. In this paper, a comprehensive review of chart analysis methods is presented. The review covers a wide range of chart types, including line charts, bar charts, scatter plots, and includes an in-depth analysis of each method. Additionally, this paper provides a more extensive coverage of chart analysis methods compared to previous studies, making it a valuable resource for researchers and practitioners in the field. Various techniques can be categorized from different aspects, such as chart type, model architecture, learning algorithm, visual feature space, and language modeling. In this paper, different methods are classified from a more technical viewpoint, by considering the approach used for modeling the problem. A taxonomy is proposed which divides the methods into three major categories: rule-based, chart captioning, and chart question- answering approaches. The rule-based approach uses the classical knowledge representation methods for reasoning, which has been diminished by the emergence of deep learning models. Chart captioning provides a general summary of the information conveyed by a chart through recent modern learning methods but may miss some detailed information which may be of special interest. On the other hand, the question answering allows for a direct response to a more specific user question by combining image analysis and text understanding techniques. Finally, the existing challenges and the potential research directions of the interesting chart understanding problem are discussed.

INDEX TERMS Automated chart analysis, image captioning, visual question answering, modality translation, multi-modal learning.

I. INTRODUCTION

Non-textual components, such as images, diagrams, mathematical formulas, plots, and charts, are often included in documents to provide readers with additional insights and information. While visual representations can be easily understood by fully sighted individuals, they can pose accessibility challenges for people with visual impairments and are not easily searchable through text-based queries. This is especially true in the case of scientific literature, which is often inaccessible to visually impaired people [1], [2]. Converting charts, maps, formulas, etc., from images into text, (through e.g., optical character recognition (OCR), pattern recognition, or machine learning techniques), can make statistical, mathematical, and other scientific content accessible to all, facilitate search and indexing [3], [4], and

The associate editor coordinating the review of this manuscript and approving it for publication was Yudong Zhang.

enable use in digital environments where images are not easily supported. Providing textual captions for non-textual and especially multi-media content, is an essential step in Search Engine Optimization (SEO) and Web Content Accessibility (WCA) enhancement. The choice of annotation method will depend on the specific requirements and the desired level of accuracy and complexity.

Previously, to achieve this goal, authors and content creators needed to provide manual annotations by adding tags, hints, and alternative texts to non-textual components. Although these methods are still viable, the majority of documents are still lacking such annotations, and it would be impractical or impossible to tag them all manually.

With the recent advancements of artificial intelligence (AI) in image processing and computer vision, developing an automatic method for generating additional information for non-textual components has attracted a lot of attention among AI researchers. Early approaches attempted to find templates in the input image and associate them with some tags or sentences [5]. Follow-up efforts tried to exploit encoder-decoder architecture to generate captions for images [6]. In these methods, the encoder computes a latent representation of the input image, and the decoder can generate text based on the computed representation. Generating captions for images (also called Image Captioning or IC) is the task of generating a natural language description for an input image that describes its visual content. In other words, it simulates the human ability to recognize visual objects and their relationships with other visual elements. In image understanding, objects and their attributes are recognized, and a representation of the objects and their relationships are generated [7], [8]. After image understanding, generating well-formed sentences requires both syntactic and semantic understanding of the language [9]. Image understanding depends on the subject and image features. For example, understanding a scientific line chart differs from understanding an image of some people in a newspaper or a biomedical image. Figure 1 shows some examples of image captioning in different areas.

Question answering systems are becoming increasingly important in various fields, including natural language processing and computer vision. These systems are designed to automatically answer questions posed in natural language, and can be used to provide quick and accurate information, and improve decision-making. ChatGPT [13] is a state-of-the-art language model that has demonstrated incredible performance in question answering tasks. It uses a transformer architecture and has been trained on large amounts of text data, making it highly effective at understanding natural language and generating relevant responses. Also, it can be fine-tuned on specific tasks, such as question answering in a specific field, to further improve its performance. As a result, ChatGPT has the potential to revolutionize question answering systems and improve their applications in various fields. One example of a question answering task that combines image processing and natural language processing is Visual Question Answering (VQA). Unlike generating captions for images, VQA focuses on extracting particular information from input images and answering natural language questions about them. It is classified as a multi-modal task, as it requires the integration of information from multiple sources with different natures or modalities, specifically visual and linguistic domains, to answer a natural language question about an image [14], [15]. The visual aspect provides insight into the objects, scenes, and attributes in the image, such as appearance, location, and relationships, while the linguistic aspect supplies information regarding the question and answer, including the words and phrases used to describe the image and the desired output. A VQA system must understand both visual and linguistic information, and effectively combine them to generate a correct answer. Sometimes, VQA is referred to as visual Turing test [16] for image understanding.

Early VQA methods jointly transform image features and question features into a common latent space, then a classifier is used to determine what answer is related to this common latent representation. In early methods, most answers to the questions were simply ''yes'' or ''no'' which made the VQA task much simpler [6], [17], [18]. With the advancements of NLP methods and presenting sequence-to-sequence architectures, open-ended VQA methods were reintroduced [19], [20].

Although IC and VQA have a history from early 2000 [21], their focus has been on general images. Automatic analysis of specific domain images such as medical images, maps, and scientific charts is still arduous due to several challenges [22]. In many articles and papers, quantitative and qualitative information are represented as charts or plots which are not accessible to the visually impaired persons, search engines, etc. As a result, automatic understanding and captioning of chart images have become an active research topic in recent years. Finding chart images in a document and recognizing its type and style is a prerequisite for the analysis and explanation of its content, which has been reviewed in [12] and [23].

Chart understanding has unique properties which make it different from other image recognition problems, both from the modeling and solution design perspectives. For example, the following aspects of this problem compel the researchers to devise exclusive solutions for it:

1) Complex structure: Charts contain different elements such as title, axes legend, labels, and data points which may have different visual properties. This complexity makes the element identification and information extraction difficult.
2) Data-driven: Charts are used to represent quantitative data. The information conveyed by charts are not only visual but also numerical. Chart understanding methods need to extract and interpret this data.
3) Field-specific: To understand a chart, humans typically need to have some level of knowledge about the field the chart pertains to.

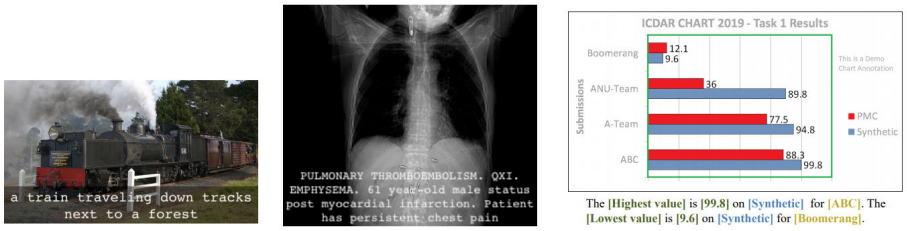


FIGURE 1. Different image captioning tasks: left) general image captioning [10], middle) biomedical image captioning [11] and right) scientific chart image captioning [12].

Surveying the information extraction methods from chart images and transforming the extracted information into natural language sentences is the focus of this paper. To the best of our knowledge, this paper presents a timely and up to date comprehensive survey which concentrated on chart image understanding problem for the first time. Previous related review papers [12], [23], [24] have covered more general and somehow different domains (e.g. chart detection and classification). Moreover, those have not completely reviewed the last recent advances on the specific domain of chart understanding problem (e.g. chart image captioning [29], [34], [59], [60], question answering and, reasoning [31], [32], [88], [91]).

Most charts are similar in shape and structure. Figure 2 shows four common types of charts. As can be seen in this figure, all charts have a title on their tops. Bar charts (A), line charts (D) and scatter plots (C) have an X-axis and Y-axis, which provide guidance in measuring the data. Pie charts (B) have circular shapes and labels. Extracting and parsing structural components inside chart images is crucial for the further analysis of the chart. Textual components are also fundamental in recognizing relations and measuring. Research indicates that a model optimized for one type of chart may not perform equally well on other chart types. For example, a model that excels at handling bar charts may not be effective for pie charts. Furthermore, the design and style of charts can vary. For instance, bar charts may come in both horizontal and vertical orientations, and scatter plots may incorporate special markers. The placement of the legends in charts that have multiple sets of data can vary all over the chart, creating a challenge for automated chart understanding techniques. Additionally, differences in font style and text orientation can also present difficulties for these methods.

This study provides a review of recent automatic chart image understanding techniques. The works are primarily chosen based on their relevance to the research topic, their credibility as peer-reviewed publications, and their recency. However, certain works are excluded due to their reliance on restricted datasets, their excessive use of simplifying assumptions, or their lack of significant relevance to the main topic of discussion. It is also worth mentioning that conducting meta-analysis was not feasible due to the differences in datasets, assumptions, and methodologies used in the selected works.

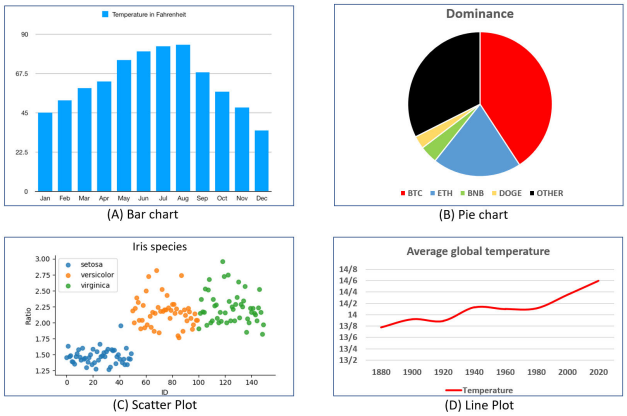


FIGURE 2. Samples of different chart types.

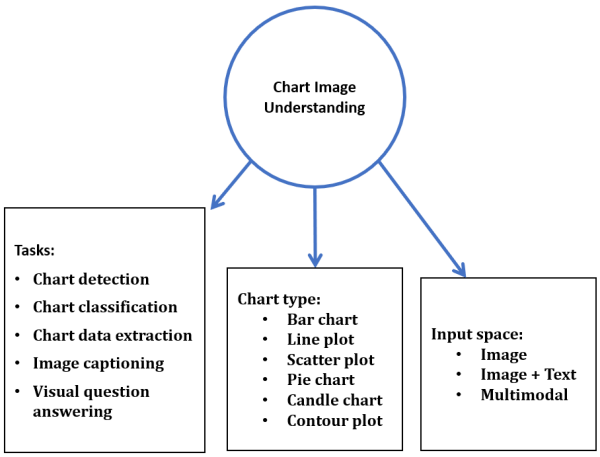


FIGURE 3. Different problem domains of automatic chart image understanding.

Figure 3 presents different aspects which can be used for classification of these techniques, including: 1) Tasks: In addition to IC and VQA, chart detection, classification, and data extraction are also considered here. 2) Chart Type: Several usual types have been shown in Figure 2, but more chart types can be considered like scatter plots, bar charts, pie charts, and candlestick charts. 3) Input Space: Based on the input space different feature learning types as well as different multi-modal feature fusion types are considered.

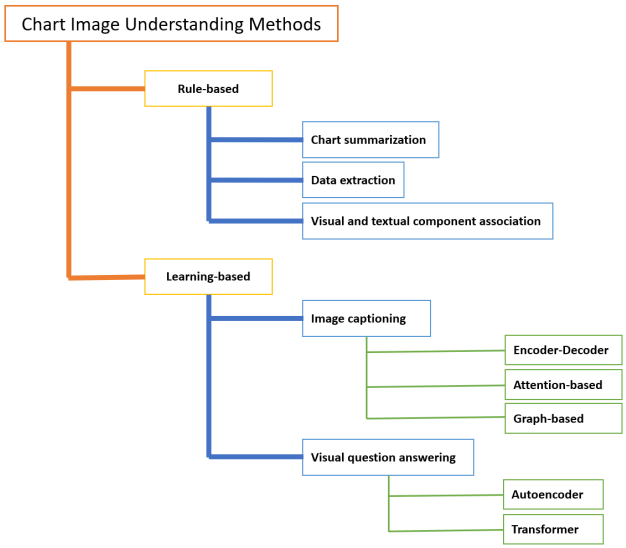


FIGURE 4. The taxonomy for chart image understanding, proposed in this paper.

In this paper, we propose to categorize chart image understanding methods based on a technical perspective, which takes into account the problem modeling approach adopted by each method. Figure 4 shows the proposed taxonomy for categorization of chart understanding methods. Early methods primarily rely on rules, whereas novel methods mostly exploit deep learning approaches. Learning-based methods are divided into two subcategories: image captioning and visual question answering. Each subcategory encompasses distinct technical approaches. Section II covers rule-based methods for chart data extraction and caption generation, while Section III evaluates image captioning methods, highlighting their advantages and limitations. The discussion on Chart Question Answering (CQA) techniques can be found in Section IV. It should be noted that throughout this paper, the term ''Chart Question Answering (CQA)'' is used in reference to VQA on scientific chart images. Section V examines commonly used datasets and Section VI covers evaluation metrics for captioning and question-answering tasks in image understanding methods. The challenges faced in chart understanding tasks are briefly discussed in Section VII. Potential future research directions are discussed in Section VIII. Finally, the concluding remarks are given in Section IX.

II. RULE-BASED METHODS

Charts and plots are usually created based on tabular data. Some methods attempt to ''reverse engineer'' chart images and extract the source tabular data. Although tabular data extraction methods do not generate natural language descriptions, extracted data can be used for further interpretation and analysis to generate accurate and informative captions. Existing tabular data extraction methods are mostly based on predesigned rules, heuristics, and hand-crafted features such as image edges, HOGs [37], SIFT [38] descriptors, etc.

A. DATA EXTRACTION

Automatic data extraction from chart images is difficult because graphical components (e.g., lines, axes, tick marks, legends, etc.) and textual components (e.g., chart titles, axis labels, tick values, etc.) are needed to be interpreted differently and consistently [39]. As mentioned above, one of the first points that needed to be considered is chart type. Different chart types have different graphical and textual components and can be completely different visually. For example, pie charts do not have axis and ticks, but line charts do. These variations in chart components make the design of data extraction rules specific per chart type. For instance, a rule can be ''X-axis label is on the bottom-center of the image'', or ''closed rectangular shapes are bars'' in a bar chart. Specific rules simplify the structural parsing of the chart image but make the method heavily dependent on chart type and even style. For this reason, several works use interactive methods to reduce this bias.

For example, PlotDigitizer [41] is a semi-automatic tool that takes an image as input and requires the user to specify the X and Y axes and a data series line. With the knowledge of the axes and a line point, PlotDigitizer employs the auto-trace method [42] to follow the points on the line and extract the data. Surveys have revealed that while semi-automatic techniques can be highly precise, they cannot be employed on a large scale or in scenarios where human intervention is not possible. Thus, we concentrate on fully automatic data extraction methods.

In ReVision [43] a two-step method is proposed in which the first step is to identify the chart type. For this purpose, the input image is split into small patches and 100 patches are sampled randomly. Then, K-Means clustering is performed on randomly sampled patches to obtain a set of ''centroid'' patches that correspond to the most frequently occurring patch types. Centroid patches are then passed to an SVM classifier to determine the type of input chart image. The second step is data extraction which is distinctive for each chart type. In ReVision, only pie and bar charts are regarded for data extraction. For example, in bar charts, background removal is done, and rectangular shapes are found using connected component analysis (CCA). Assuming that the Y axis is on the left side of the chart, a mark detection process is performed to find marks and values on the Y axis. Figure 5 shows the procedure of extracting bars from a chart. After finding bars and marks, the relative height of each bar is calculated with respect to the largest value (corresponding to the top-most mark) on the Y axis. The relative heights can then be converted to values based on marks' values. ReVision has reported 90% accuracy in chart type classification and 79% accuracy in bar chart data extraction on Prasad et al.'s [44] dataset.

A similar approach is adopted in [45] which extracts data carried by the chart. In this method, a classifier determines the chart type. Based on the detected chart type, a set of rules is chosen for graphical component detection. For this

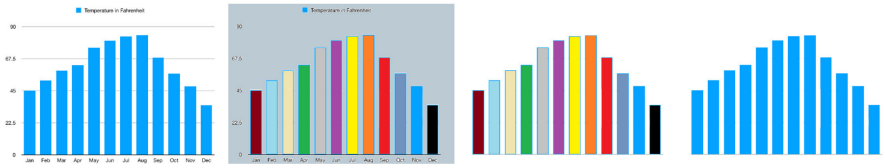


FIGURE 5. Procedure of parsing bar charts proposed in ReVision [43].

purpose, they exploit edge structure models to represent the geometrical and topological structure (e.g., line segments or circular arcs) of image edges. Also, a text detection algorithm is used to localize and crop textual elements, which are then passed to the tesseract [46] OCR tool. Finally, data extraction is done by assigning the names and numerical values to the detected chart components.

ChartVi [47] is another method that automatically interprets chart images to help people with visual impairments better understand them. It works by acquiring data from images and generating concise summaries. ChartVi uses handcrafted methods for data acquisition and OCR for text extraction, followed by a summary generation method on the extracted data. Similar to other data extraction methods, the challenge is to extract all the information needed to generate semantically correct summaries.

Another automatic data extraction method was proposed by Sreevalsan-Nair et al. [48] which utilizes tensor fields to detect the spatial location of corners of the top line of bar charts and scatter points in scatter plots. Then, the locations of the corners are sorted and fed into a scanline algorithm in ascending order of x to determine missing points. The range of y-intervals at each x value gives the univariate distribution of the chart data. For scatter plots, the location of scatter points provides a bivariate distribution of the chart. Although their model could perform well in ideal situations, dependency of tensor fields on user-defined image characteristics such as image resolution, bar width, borders, etc., makes the model vulnerable to variations in design/templates.

B. VISUAL AND TEXTUAL COMPONENT ASSOCIATION

Visual and textual component association refers to the process of connecting and relating visual elements, such as points, axes, marks, etc., with the textual information extracted from OCR. This process involves identifying and extracting both the visual and textual components from a given chart image, and then associating them in a meaningful way. There are typically two levels of association: logical-level and semantic-level. Logical-level association involves recognizing the role of each textual component based on its spatial location, while semantic-level association involves extracting data values from the chart components. For example, in a bar chart, the heights of the bars represent the value of the data, and the labels along the horizontal axis describe the categories being measured. The visual component, which is the heights of the bars, must be accurately aligned with the textual component that is the labels on the horizontal axis, to convey the correct information. Similarly, in a line chart, the data points are connected by lines to show trends over time, and the labels along the horizontal and vertical axes describe the units of measurement. The lines connecting the data points, as the visual component, must be accurately associated with the labels on the axes, as the textual component, to provide accurate context for the data. The goal of visual and textual component association is to gain a deeper understanding of the information contained in the chart image and to make it more accessible and searchable.

In [49] textual and graphical components are detected using CCA based on a set of filters. Then, textual components are recognized through OCR. Understanding of the chart image is achieved by associating the extracted texts and graphical components. For example, in the logical level in a bar chart, the height of a bar that makes a cross on the x-axis label and y-axis label can be used as a reference value. Then, in the semantic level, heights of other bars can be measured relative to this reference bar.

Given the crucial role of texts inside chart images, the work done in [50] aims to enhance text extraction and association. OCR engines, such as Tesseract [46], are utilized to identify and extract text from charts. However, Tesseract OCR has limitations in detecting smaller text regions. To address this issue, the authors propose a preprocessing step that detects and locates text regions before they are passed to the OCR engine, thereby increasing recognition accuracy. This research primarily focuses on bar charts and pie charts.

It is apparent that no data extraction method can work well on all types of charts, because each chart type needs a specific set of heuristics. Finding the best heuristics for model design is difficult. Also, a specific chart type can comprise very different designs or templates that make the heuristics inefficient. However, it is also improper to directly apply an end-to-end solution since these methods usually deal with a specific type of charts. For this reason, several methods try to use feature extraction instead of using rules for chart component detection.

Scatteract [36] is a method that attempts to detect chart components using a general detection model. In this method, three Convolutional Neural Networks (CNN) are trained

to detect tick marks, tick values and points, respectively. For each tick mark, the closest tick value is found and sent to an OCR tool. Finally, a robust regression is performed to determine the mapping from pixel coordinates to chart coordinates, for each pixel. Similarly, Choi et al. [51] proposed a method with three CNNs for chart classification, text extraction, and object detection, respectively. Objects including labels, legends, lines, and rectangles are then passed to a data extraction pipeline to reconstruct data values and visual encodings.

The methods reviewed so far work on scatter plots, bar charts, and pie charts. One of the most challenging chart types for automatic data extraction is the line chart. The continuous nature of line charts makes it necessary to define how many data points should be extracted. Also, the axes scales can be non-linear (e.g., logarithmic) and it is common in line charts to have more than one line (data series). In case of non-linear charts, recognizing the scale is crucial for accurate data extraction. In case of multi-series line chart, each line needs to be separated and handled individually. For example, in [39], CCA is used to tackle this problem. Another work [40] proposed a curve separation procedure in which different colors in chart image are counted and a histogram is created. Histogram bins with high-frequency values correspond to the curves. Then a method called curve legend association is performed to link each curve to a data series. This association is based on the line colors or data marks.

To the best of our knowledge, there is no unified end-to-end method capable of automatic data extraction from different chart types. Generating accurate descriptions for charts requires precise knowledge of the source data. However, it is possible to generate abstract or high-level descriptions for chart images using rules.

C. CHART SUMMARIZATION

Chart summarization is the process of condensing the information presented in a chart or graph into a brief, concise and easily understandable form, to make complex information more accessible and understandable, without sacrificing important details or accuracy. This can involve summarizing the data presented, highlighting important trends or patterns, and presenting a simplified version of the chart that conveys the most important information in a clear and straightforward way. In other words, the goal of chart summarization is to identify the knowledge conveyed by the chart to extract important concepts and integrate them into coherent natural language sentences.

For instance, Greenbacker et al. [52] developed a line chart summarization method. Figure 6 shows a summarization example for a line chart. Since a line chart can consist of short fluctuations, Greenbacker et al. proposed a graph segmentation method that generalizes line charts into sequences of falling, rising, and stable segments, where a segment is a series of connected data points. Once the chart has been converted into a sequence of trends, several candidate captions are generated. An example of two candidate captions is shown below:

- There is a rising trend from <param1> to <param2>.
- There is a significant sudden jump in value between <param1> and <param2> which may or may not be sustained.

Once the candidate captions have been generated, the next step is to identify the parameters (<param#>). Unlike the reviewed methods which use OCR to identify the chart's parameters, the work of [52] exploits the captions and article's text to find ''Verb in caption evidence'' and ''Adjective in caption evidence''. One major disadvantage of this method is the low diversity of the generated captions. Also, this method is heavily dependent on captions and the texts provided for the chart inside the document. Thus, it can only work on chart images inside a document.

In rule-based methods, the limitations on the chart type, style, and candidate captions propel the researchers to more generalized methods. Recent advancements in image and natural language processing have opened up new possibilities for achieving better solutions. Convolutional Neural Networks (CNNs) have demonstrated remarkable performance in image processing and machine vision tasks. On the other hand, Recurrent Neural Networks (RNNs) and Transformers are well-suited for processing sequence-based data (such as natural language sentences).

FIGURE 6. "This line graph shows a big jump in Bitcoin price in September 2021. The graph has many peaks and valleys between July 2021, to July 2022 but maintains an average price of around 40K dollars. However, in October 2021 the price jumps sharply to around 60K dollars before dropping quickly to around 40K dollars by January 2022." An example of line chart summarization [52].

III. SCIENTIFIC CHART IMAGE CAPTIONING

Considering the importance of image captioning methods for general images, we first review some of the popular general methods. Most successful scientific chart captioning methods make use of the ideas employed in the general image analysis methods. As mentioned in Section I and shown in Figures 3 and 4, we can categorize image captioning methods into different groups based on their approaches and the architectures. In the following, we categorize general image captioning methods into three subcategories based on their model architecture. In the last subsection, several methods are presented that have applied one of the reviewed architectures onto chart images.

A. ENCODER-DECODER ARCHITECTURE

We can consider image captioning as a modality translation task. Image captioning attempts to translate image modality to a text modality. Thus, several researchers have used encoder-decoder models for this task. In encoder-decoder-based methods, a model is used to translate visual information to an intermediate representation, and then another model is used to translate the intermediate representation to text/description. A CNN which is usually pre-trained on a large dataset is used to encode visual data to a fixed-size feature vector. After that, an RNN is used as a language model which takes the feature vector and generates a sentence that describes the input image. This type of image captioning method was first introduced by Kiros et al. [53]. Figure 7 shows the architecture of a simple encoder-decoder model.

One advantage of encoder-decoder-based models is that both encoder and decoder networks can be trained together. This kind of training, also known as end-to-end learning, makes both encoder and decoder networks cohesive and avoids adapting independent components. Though encoder-decoder-based methods are straightforward and simple, they have continued to outperform other image captioning methods to date [54].

Despite the advantages of encoder-decoder-based methods, they suffer from high levels of bias, which propels the model to generate low-diversity sentences [55]. In other words, the model cannot generate sentences that didn't exist in the training corpora. Another drawback of encoder-decoder-based methods is that when the extracted feature vector is fed into the decoder model, it only affects the first layer of the decoder. As a result, only the initial words of the output sentence are dependent on the feature vector. Consequently, the significance of initial words becomes less and less, as the sentence gets longer. This problem is similar to the vanishing gradient problem, in which the image information only affects a number of initial layers of the decoder model.

As mentioned before, RNNs are powerful in modeling natural languages due to their capability of representing a variable-length sequence. However, because of the vanishing gradient problem, traditional RNNs are difficult to train, especially when the expected output sequence should be long. Long Short-Term Memory (LSTM) architecture can help to avoid vanishing gradient problems due to its gating structure and memory mechanism. For example, Donahue et al. [56] proposed a stacked version of LSTM in which every LSTM unit takes the encoded feature vector and the previous word as its input. In this manner, the encoded feature vector affects all the LSTM units, and each output word becomes dependent on the visual features extracted by the encoder module.

Although LSTM can help in addressing vanishing gradient problem, it still ignores long-distance dependencies. In other words, the effect of initial words becomes weaker on later words. In 2017 another technology boost came with the advent of transformers [26] which mimic the human ability in concentrating on the important parts of a sequence. Concentrating on different parts in a sequence is referred to as attention mechanism which solves the problem of long-term memory and gradient vanishing in RNNs.

B. ATTENTION-BASED ARCHITECTURE

The attention mechanism is the process of enhancing more important parts of input data and reducing the importance of the rest parts. The attention mechanism can be applied at both sentence and image levels. At sentence level, attention mechanism allows the decoder to notice different words of the current word sequence. At image level, the attention mechanism divides the input image into N regions. The importance of each region changes when the transformer generates a word. In this manner, the attention mechanism helps the model to focus on the relevant parts of the image.

Although the transformer model was officially proposed in 2017, the main idea of transformers, namely the attention mechanism, was already in use before then. For example, an image captioning method based on attention mechanism was proposed by Xu et al. [25] in 2015. Their model is similar to encoder-decoder-based methods, with a slight difference: Unlike encoder-decoder models in which the output of a CNN is used as the final feature vector of the encoder, in [25] a context vector is generated using the features learned at lower layers of the encoder network. The idea of using lower convolutional layers is that the use of the last layer of CNN may cause the loss of details. They also proposed two different techniques of attention: 1) hard attention and 2) soft attention mechanisms. In the hard attention mechanism, one feature map from a convolutional layer is used. In soft attention, a combination of all feature maps from different convolutional layers is used as the output feature vector. The vector that specifies the weight of each feature map is called the context vector that determines the importance of each feature map in each cycle of the recurrent neural network.

Another attention-based method was proposed by Jin et al. [57], that can extract abstract concepts based on the semantic relationship between visual and textual information. In their method, the input image is first analyzed and divided into multiple regions at multiple scales from which visual features are extracted. Extracted visual features are then fed into an RNN, which predicts the sequence of regions and words.

C. GRAPH-BASED ARCHITECTURE

Similar to the attention mechanism that considers the relationships between different objects in images and different parts of a sequence, several methods use graphs to model relations between objects. For example, a method called scene-graph generation [28] attempts to represent the input image with a graph. The graph comprises two types of nodes, 1) object nodes which represent objects (e.g., car,

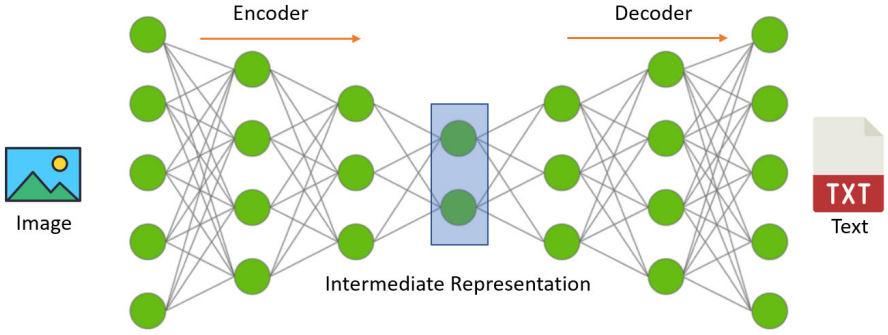


FIGURE 7. Architecture of encoder-decoder-based image captioning models.

dog, table), and 2) relation nodes which represent a relation term (e.g., holding, behind, playing). Objects can have direct connections with each other or an indirect connection through a relation node. Unlike object detection tasks in which the model predicts the object classes, scene-graph generation additionally predicts relations between objects. By analyzing paths in the generated scene-graph, it is possible to produce some sentences constituent. Each constituent describes a fact about the scene, e.g., ''a man in a car'', ''a cat playing with a ball''. Finally, short pseudo-sentences can be combined using a language model to form a proper description.

Another model proposed by Li and Jiang [27] uses hierarchical attentions. The first step is to generate a scene-graph for the input image. In addition, bounding boxes of the detected objects are extracted and then fed into a CNN to extract visual features of each object in the input image. In the second step, by analyzing the scene-graph, triples (three lexeme sequences) are extracted. Each triple describes a relationship between two objects (e.g., ''man near motorcycle'', ''man wearing shirt''). In the third step, triples are embedded into fixed-length vectors. The fourth step is the attention mechanism that determines which embedded triple vector and visual feature vector should be fed into the LSTM for generating the next word.

Generating a short sentence for each triplet suffers from the drawback of redundancy of information, while also resulting in a lengthy and non-specific final text. To address this issue, it has been suggested that text summarization techniques be employed to generate a concise version of the input text while preserving its salient details.

Although text summarization methods can solve redundancy problem, it makes the description more abstract and less detailed. Lundgrad and Satyanarayan [64] carried out a study on 90 individuals with normal vision and 30 individuals with visual impairments. The results showed that while many image captioning techniques produce abstract captions that may be pleasing to those with normal vision, they are not useful for individuals with visual impairments, especially if they have not previously seen a visual representation. Their work also shows that some captioning methods raise ethical concerns. For instance, one of their subjects noted that ''if a description were to only describe a visualization's encodings, then the reader wouldn't get any insight from these texts, which not only increases the readers' burden but also conveys no effective information about the data'' [64].

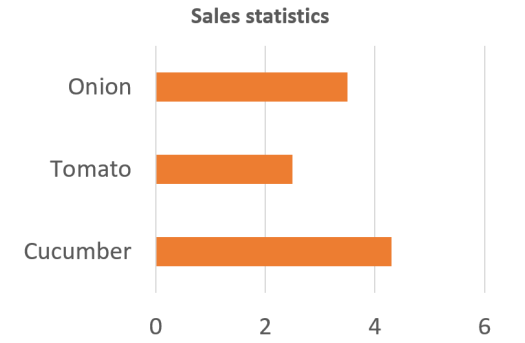
Another challenge in image captioning is the evaluation of generated captions. Existing quality evaluation metrics such as BLEU [65], ROUGE [66], and, CIDEr [67] are based on machine translation tasks which are not proper for captioning tasks [68]. Due to these reasons, one can suggest using VQA frameworks. VQA on chart images is discussed in secion IV.

D. APPLICATIONS ON CHART IMAGES

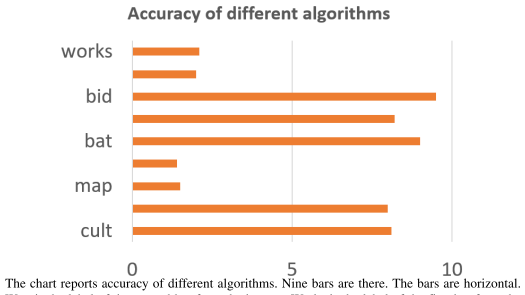
The methods reviewed earlier in this section were all general image captioning methods. Almost all recent successful methods use deep learning paradigms. Due to the complexity and huge number of parameters in deep neural networks that should be learned, a large number of training data is needed. Datasets such as MSCOCO [10] that have thousands of image-caption pairs have made the training of deep image captioning models possible. However, for scientific chart images, there is no such public dataset available till now [12]. To deal with the lack of large-scale datasets for scientific chart image captioning, some interpretation and investigation processes are needed. In the following, methods for scientific chart image captioning are discussed.

Unlike general images, scientific charts often have text within their images. Textual components thus play an important role in chart understanding. Text detection and recognition can benefit automatic chart understanding methods. For instance, FigJAM [29] generates annotations for textual components inside a bar chart which are extracted using an OCR tool. Given a textual component (e.g., chart title, label name, value, chart type, min/max, etc.), the goal is to generate a sentence which is called a ''caption unit''.

Assume that there is a horizontal bar chart with a bar labeled ''F1''. By giving ''type'' as the input to the model, the output caption unit should be something like ''This is a horizontal bar chart'', or by giving ''F1'' to the model,



The chart compares sales stats among different products. Three bars are there. The bars are horizontal. Cucumber is the label of the first bar from the bottom. Tomato is the label of the second bar from the bottom. Cucumber sold the most units. The item Tomato sold less units than Onion.



The chart reports accuracy of different algorithms. Nine bars are there. The bars are horizontal. Wart is the label of the second bar from the bottom. Works is the label of the first bar from the bottom ... Bid has the highest accuracy. The accuracy of the algorithm cult is smaller than bid. The accuracy of the algorithm map is smaller than cult. The accuracy of the algorithm cult is larger than bat ...

FIGURE 8. An example of FigJAM [29] caption generation for two bar charts with different complexity.

the output should be "F1 is the label of the first bar from the bottom". To achieve this goal, a pre-trained ResNet-50 is used [29] to extract visual features from the input chart image. The extracted visual features are then weighed based on the given textual component ("F1" in the example). This method which is similar to attention mechanism [26] forms an input for the sentence generator module which is an LSTM network. The LSTM network acts like a decoder that takes the extracted visual features and a word (e.g., "type", "F1"), and then generates a caption unit. Figure 8 shows an example of FigJAM results for two bar charts.

As can be seen in Figure 8, for a bar chart with several bars, the number of caption units increases rapidly. Although FigJAM can generate accurate caption units for each bar, it lacks the capability of combining caption units and summarization. Since the main objective of charts are to encode information into a compact form and to highlight the most important parts of information, it's not proper to generate a sentence for every single data point.

This problem exists in other methods as well. For instance, Chen et al. [58] also proposed a model for caption generation. In their method, three different feature vectors are generated: 1) visual features, that are extracted using a ResNet model, 2) label maps, that consist of textual components inside the chart images and are extracted using an OCR tool, and 3) relation maps, that specifies the relations between visual features and label maps. After forming three feature vectors, they are passed to an LSTM network through three attention mechanisms. The relations are changed based on the last word the LSTM network generates. Figure 9 shows FigCAP [58] architecture.

A method based on encoder-decoder architecture is proposed in [59] for line charts. Unlike other methods which accept an image as their input, [59] assumes that the chart data is extracted in a preprocessing or data extraction step. Therefore, they gathered a training dataset by crowdsourcing that includes captions, chart images, and their corresponding tabular data. Both encoder and decoder are LSTM networks in which the encoder network is fed with time series values and the decoder is fed with caption words. The aim of their model is to generate captions based on the input time-series values.

A summarization method for Area Under the Curve (AUC) plots is proposed by Safder et al. [60], which extracts semantics such as text, plotted data, and lines from the input chart image and estimates a function for each line using a line fitting method. In the next step, their method searches for related texts and captions in the full-text document. Finally, it combines the chart semantics from parsing image and the text which is extracted from full-text document to generate specialized summaries.

A method called Chart-to-Text proposed in [61] uses a transformer model for sentence generation. This method assumes that a preprocessing step has been performed and numerical and textual data are extracted from the chart image. Then, the model tries to generate a summary based on the extracted data. They have also provided a benchmark dataset [62] covering various topics and chart types and tested state-of-the-art (SOTA) neural models on it for generating summaries.

The method proposed in [63] involves extracting and identifying visual marks, visual channels, and text information from the charts using a multilayer perceptron classifier. A 1-D convolutional residual network is then used to analyze the relationships between visual elements and recognize significant features, with both data and visual information as input. Finally, a caption of a visual chart is generated through a template-based approach, effectively covering main visual features, and supporting major feature types in common charts.

Another captioning method is proposed in [34] that works on bar charts. They first classify the input chart image and specify its variant (horizontal/vertical, histogram, single/multi-series bar chart, etc.). Then, object detection and text detection processes are performed on the image. The next step is to extract tabular data which is done in a semi-automatic manner. If the chart is a histogram, a distribution fitting method is used. Finally, a summary is generated for the input chart.

IV. CHART QUESTION ANSWERING

Chart question answering is a specific sub-task of VQA, which focuses on generating answers for questions

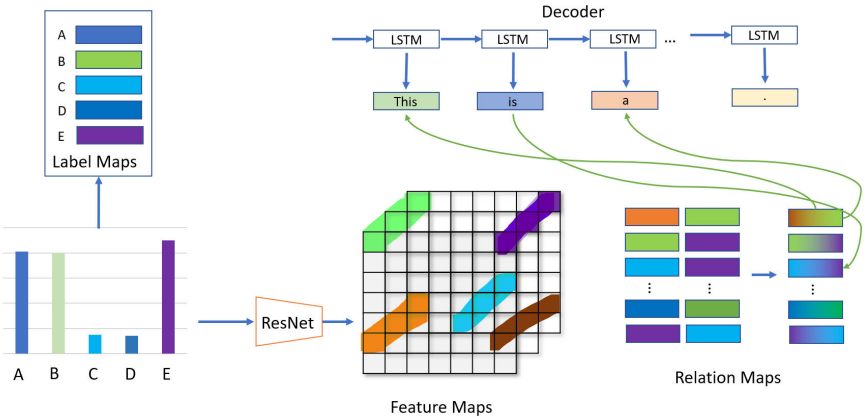


FIGURE 9. Architecture of FigCap [58].

related to charts, graphs, and other data representations. In CQA, a machine learning model is trained to recognize chart-specific information and answer questions about the data points, relationships, and patterns within the chart. VQA, on the other hand, is a multi-modal task that involves combining visual and natural language information to answer questions about images. VQA can involve tasks such as object recognition, grounding, reasoning, and counting. While image captioning models focus on generating a natural language description for an image, VQA models must also consider the input question and search for information relevant to the question.

A. VISUAL QUESTION ANSWERING

As mentioned earlier, visual question answering is a multi-modal task that combines vision and language. Unlike image captioning in which a model generates a free-form natural language description for an input image, VQA gets involved in specific object recognition, reasoning, etc. tasks. Generally, an IC model chooses the most relevant sentence for the input image, but VQA needs to consider the question and search for information that is relevant to it. Some questions may require reasoning about data points and their relationships.

Recent studies showed that recognizing links between question words and image regions (or objects) is a key factor for enhancing interaction between modalities [69]. Although attention mechanism can assist in linking image regions to question words, research has shown that attention-based models do not look at the same regions as humans [70]. For this reason, co-attention models [71], [72] have been proposed which contain complete interaction between question words and image regions. Co-attention mechanism leverages visual and textual attentions simultaneously. In this way, the co-attention network selectively attends to a question word and a region in the image. It should be noted that co-attention neglects the internal relations between image regions or question words to decrease computational complexity [71].

Another set of approaches for modeling relations is graph-based e.g. using multimodal graph-based data fusion methods [73], [74]. In graph-based methods image regions and question words are represented by graph nodes and the nodes that are related together connected by graph edges. In this manner, inter-modality and intra-modality relations can be modeled using a single graph.

For example, in Mucko [75] an object detection process is used to recognize image objects. Then, a graph representation is produced in which the nodes are corresponding to the objects and the edges represent relations. Edges between graph nodes are obtained based on an external knowledge base. Also, another graph is built based on the training question/answers in the dataset. In other words, the edges in the second graph are obtained based on training data. To combine the two graphs, Graph Convolutional Networks (GCN) [76] are used. After combining the two graphs, a vector representation of the graphs and input question is obtained based on graph convolution process which is used for answering the input question. Figure 10 shows an illustration of Mucko [75] method.

Even though image regions or question words might have relations with each other, some VQA methods do not consider intra-modality relations. The method proposed in [72] attempts to infer inter- and intra-modality relations. In other words, the relations between an image region with other image regions and question words are obtained in a unified way. Figure 11 shows an illustration of the model proposed in [72].

Several VQA models specialized to consider texts inside general images [77], [78], [79]. In text-based visual question answering (TextVQA), a model must answer questions about visual scenes with text reading ability. For instance, a solution called the Text-Instance Graph (TIG) network has been proposed [79], which models the relationships between visual object instances and texts obtained by an OCR module using a graph structure. The TIG also has a dynamic network to extend the perception space and handle complex logic in questions.

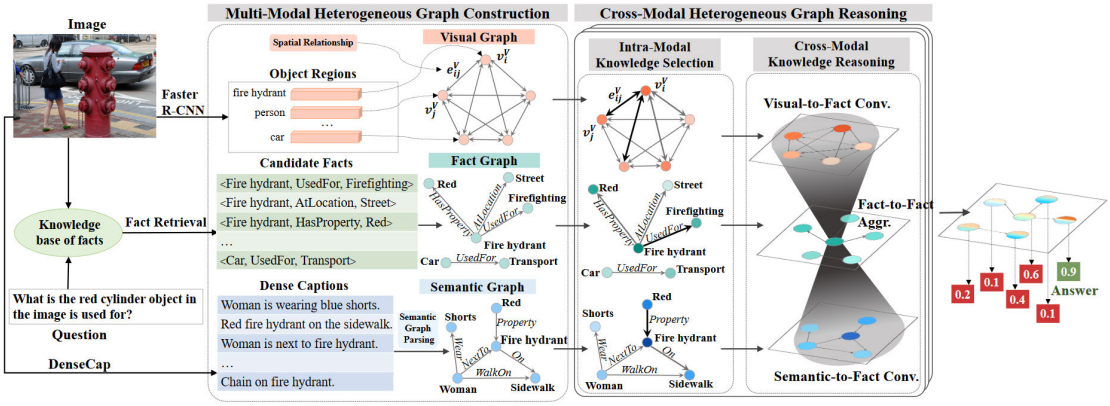


FIGURE 10. An illustration of Mucko method [75]. Two semantic graphs are created based on training data and an external knowledge base. Two graphs are combined using graph convolution operation. The resultant graph is representing a semantic space that can be used for question answering. This figure is directly taken from [75].

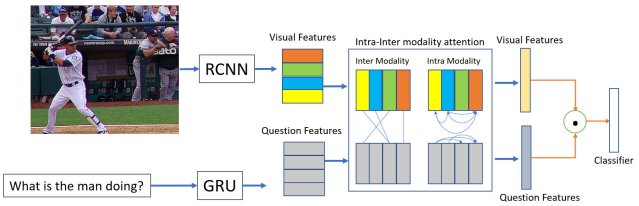


FIGURE 11. Inter and intra-modality attention mechanism proposed in [72].

Although recent works on general-domain visual question answering such as [80], [81], and [82] have achieved high performances, they are not capable of handling domain-specific images. In CQA, models must recognize the structure of the chart and extract information from it to answer questions. This requires not only a good understanding of the visual information, but also an understanding of the relationships between data points and the underlying structure of the chart.

Kafle et al. [30] show that general VQA methods are only capable of answering simple structure questions and perform poorly on data retrieval and reasoning. They presented DVQA dataset containing bar chart images and several question-answer pairs for each bar chart. The majority of VQA methods work as classification systems [30]. A classifier trained on a static and predefined vocabulary is incapable of answering questions that are not encountered during training. This problem, which is referred to as the Out of Vocabulary (OOV) problem, can be solved by taking textual elements in charts into account [83]. OCR can help to solve the OOV problem by providing the text contained within a chart or graph to the chart question answering model. By extracting the text, OCR can provide the chart question answering model with a complete vocabulary of words and phrases that are present in the chart or graph. This can ensure that the model has access to all of the relevant information needed to answer questions about the chart or graph.

B. AUTOENCODERS

Autoencoders are simply an encoder and a decoder coupled together to reproduce the input at the output. The encoder processes the input data and generates a compact representation that is used as the input to the decoder, which generates the output data. The utilization of autoencoder architectures in VQA involves combining text and visual modalities. Autoencoders can be trained to convert information from multiple modalities, such as text and images, into a unified latent representation. This representation can be utilized for various purposes, such as classification, generation, or visualization. Through the training process, the autoencoder learns to retain important information from each modality while disregarding irrelevant details. This results in a shared representation that can be leveraged to tackle multi-modal tasks by harnessing the complementary nature of different modalities.

Kafle et al. [30] proposed two models, MOM and SANDY, with the aim of addressing chart-specific answer generation problems. These models attempt to generate answers that are specific to charts and diagrams, which can be difficult due to the complex nature of these visual representations. To achieve this, the Stacked Attention Network (SAN) [84], which is an autoencoder is utilized. The SAN calculates a weighted sum of vector representations for different regions of an image, as well as the question words that are encoded. This approach allows the model to assign more weight to the image regions that are most relevant to the question words, while assigning less weight or even zero weight to irrelevant regions. This results in a more focused and accurate answer generation process, which is the goal of the MOM and SANDY models.

MOM [30] uses dual-network architecture, a classification network that generates a "generic answer", and an OCR network that is responsible for chart-specific answers that must be read from the bar chart. More precisely, the OCR network predicts the bounding box containing the correct label, and then extracts its text. In other words, the OCR network generates a dynamic per-image vocabulary. Although MOM

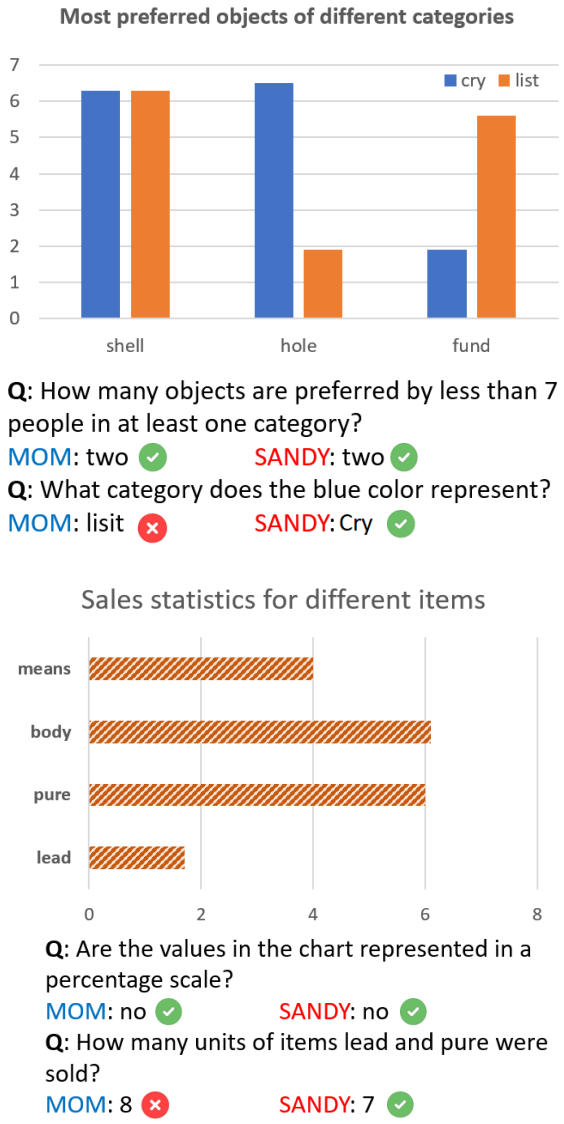


FIGURE 12. Results of MOM and SANDY [30] models for two bar charts.

generates chart-specific answers, its LSTM question decoder cannot handle chart-specific words. SANDY uses a dynamic encoding model that explicitly encodes chart-specific words in the questions. Figure 12 shows the example results of MOM and SANDY. Although SAN-based methods such as SANDY perform well on binary answering (yes/no, true/false, etc.), they face challenges in cases where the answers are not from a fixed vocabulary.

Another chart question-answering method that aims to deal with the OOV problem is proposed by Methani et al. [31]. They presented a dataset called PlotQA by gathering more than 200 thousand plots from real-world sources with 28.7 million question-answer pairs based on crowd-sourced question templates. They show that previous methods can achieve less than 10% accuracy on PlotQA dataset because of the OOV problem. They also proposed a two-stage method for question answering. The first stage is a question classifier that specifies whether the input question can be answered from a fixed vocabulary or needs a more complex vocabulary. The second stage contains two models, a CNN+LSTM model (similar to SANDY [30]) which is designed for fixed vocabulary questions, and a modular pipeline which is for questions that need complex vocabulary.

Simple questions such as "What is the value of accuracy for KNN?" can be answered by looking up the correct label or axis to retrieve its value. However, more complex questions which involve different data on the chart or require some counting and reasoning can be challenging. A method proposed in [85] attempts to handle complex question-answering tasks. It first extracts visual features and the data from the input chart and creates a data table. In the next step, it transforms the input question by replacing visual elements in the chart with non-visual references to data. Finally, the extracted data table and the transformed question are sent to a data table question-answering method called Sempre [85], which is trained on a table question-answering dataset.

C. ATTENTION AND TRANSFORMERS

Despite the capability of deep networks in vision tasks, designing a model that combines accurate visual elements with the ability to reason over them is challenging. There are methods that exploit a knowledge base alongside their models to perform reasoning tasks. For example, ChartNet [86] uses a type of attention mechanism called Modular Attention for Compositional Reasoning (MAC) in Compositional Attention Network (CAN) [87] for visual reasoning on bar and pie charts. The CAN network consists of three units. The first unit transforms the input image and question sentence into a distributed vector representation. The second unit decomposes the input question into a series of operations. Operations are simply looking up into the input image and retrieving some information. The third unit is responsible for the classification of the outcome of the last operation.

Another CQA method is proposed by Singh and Shekhar [32] which uses a transformer model for answer generation. In their method, the question and the input image are fed into two embedding models. The output of the question embedding model and the output of the image embedding are then sent to a transformer model for reasoning and answer generation.

LEAF-Net [88] uses a Mask-RCNN [89] to detect and classify the chart objects. Then, textual objects are sent to an OCR module to convert them into strings. A tuple of <element type, element order> is then assigned to each text string to determine its type and its order. For example, if a textual object is of type x-axis, its order is determined based on its distance from the left corner of the image. For the input question, its words are matched with the extracted text strings from the image. For each match, a vector representation is used based on the GloVe [90] method. Finally, the extracted image objects

and word representations from the question and textual object are fed into an attention-based model to generate an answer.

A similar method is proposed in [91] consists of two stages. The first stage is responsible for detecting and classifying image objects. It also computes a vector representation for each visual object and a vector representation for each word in the input question. The second stage is responsible to fuse the vector representations of visual objects and words of the question. After the fusion stage, the common representation is sent to a classifier to determine the final answer.

Even though finding matches between image regions and question words is critical, there are situations in which no matches can be found. For instance, in a bar chart, a question can be "What does blue color represent?" whereas no word "blue" is present in the image. Similar to [85], the method proposed in [92] tries to translate visual contents in the image into word representations. They called this process "visual to non-visual" conversion. After converting all visual elements to non-visual encodings, they are passed to the Sempre model to generate an answer for the input question. Finally, the question and the generated answer are sent to an explanation generation model which generates a sentence describing how the answer was generated from the chart.

FigureNet [93] works on categorical charts like bar and pie charts. One key feature of the FigureNet model is its attention to colors and orders. For instance, in vertical bar charts, the model identifies plot components from left to right, and for pie charts in an anti-clockwise direction. This model takes the input image and calculates the probability of colors for each chart component. For the input question, an LSTM network is used to calculate a vector representation for the question with a slight difference. For words that describe a color, a specific one-hot vector is used instead of a trainable vector representation. In the fusion stage, the vector representation of the question and image are concatenated and fed into a fully connected layer to classify the correct answer.

Despite the impressive results of deep learning-based methods in both captioning and QA tasks, training such models needs a large amount of data. Providing enough training data is a challenge by itself. In addition, the huge number of network parameters makes the training costly in terms of memory and processing power. In the next section, we review some publicly available data sets. In summary, Table 1 shows a comparison of the reviewed chart captioning and chart question-answering methods based on methodology and type. In addition, Table 2 summarizes the reviewed methods and states the advantages and disadvantages of each method briefly.

TABLE 1. Comparison of the reviewed methods based on the chart types they supported and their methodology.

Method	Task	Chart Type
ReVision [43]	Data Extraction	Bar, Pie
Mishchenko et. al. [45]	Data Extraction	Bar, Pie, Line
Sreevalsan et. al. [48]	Data Extraction	Bar, Scatter
Huang et. al. [49]	Data Extraction	Bar
Scatteract [36]	Data Extraction	Scatter
Choi et. al. [51]	Data Extraction	Bar, Pie, Line
Choudhury et. al. [40]	Data Extraction	Line
FigJAM [29]	Captioning	Bar, Pie
Chen et. al. [58]	Captioning	Bar, Pie, Line
Greenbacker et. al. [52]	Captioning	Line
Spreafico et. al. [59]	Captioning	Line
Iqra et. al. [60]	Captioning	Line
Chart-To-Text [61]	Captioning	Bar, Line
Daggubati etl. al. [34]	Captioning	Bar
MOM [30]	QA	Bar
SANDY [30]	QA	Bar
Methani et. al. [31]	QA	Bar, Line, Scatter
Pasupat et. al. [85]	QA	Bar
ChartNet [86]	QA	Bar, Pie
STL-CQA [32]	QA	Bar, Pie
Leaf-Net [88]	QA	Bar, Pie
Levy et. al. [91]	QA	Bar, Line
FigureNet [93]	QA	Bar, Pie

V. DATASETS

Automatic chart captioning and question-answering tasks require datasets for training and evaluation. Most works use their own private and often small datasets. Most datasets are synthetically created because providing annotations for charts is time-consuming and costly. Additionally, most existing public datasets have limitations in the number of samples and diversity of chart images. In this section, we present several publicly available datasets that can be used in CQA and chart captioning tasks. Table 3 compares these datasets based on the chart types they contain.

A. DVQA

DVQA dataset [30] is a bar chart question-answering dataset presented in 2018 to test different aspects of bar chart understanding in question-answering frameworks. DVQA dataset contains 3,487,194 question-answer pairs related to 300,000 synthetically created bar charts. The questions are divided into three categories (from each one several examples are given): i.

1) Structure: How many bars? Is the chart horizontal/vertical? How many groups? etc.
2) Data: What is the label of the third bar from the left? Are the values in logarithmic scale? etc.
3) Reasoning: Which "algorithm" has the highest accuracy? Did "I1" sold less units than "I2"? etc.

B. FigureQA

This dataset [94] is a visual reasoning corpus of over 1 million question-answer pairs and contains 100,000 images from different charts including line plots, dot-line plots, bar charts, and pie charts. There are 15 question types that address properties like minimum, maximum, median, intersection, etc. The answers are Yes or No. All charts are created synthetically using Bokeh plotting library [98].

TABLE 2. Advantages and disadvantages of several of the reviewed methods.

Method	Advantages	Disadvantages
Greenbacker et. al. [52]	Simple, Does not require large training data	Dependent on document text, Low diversity captions, Unable to handle non-linear scale, OOV problem, Multi-stage method, No reasoning, Works only on line plots
FigJAM [29]	Simple architecture, Recognizes chart elements, Extracts demanded information, End-to-end learning	OOV problem, Generates many short captions, No summarization, Strongly dependent on OCR, Works only on bar charts
Iqra et. al. [60]	Recognizes chart elements, Estimates each data series curve, Generates summary, Computes Area-Under-the-Curve (AUC)	Requires document's text, Unable to handle non-linear scale, OOV problem, Poor reasoning, Works only on line plots
MOM [30]	End-to-end learning, Simple classification task, Capable of reasoning to some extent Multi-modal analysis	Single-word answers, Does not generate captions, Requires large training data, Does not consider inter-modality relations, OOV problem, Works only on bar charts
SANDY [30]	End-to-end learning, Simple classification task, Constructs dynamic vocabulary, Capable of reasoning to some extent, Multi-modal analysis	Single-word answer, Does not generate caption, Requires large training data, Does not consider inter-modality relations, OOV problem
Methani et. al. [31]	End-to-end learning, Handles OOV problem, Simple, Handles different chart types	Huge vocabulary (28m words), Poor reasoning, Low-diversity questions, Does not consider inter-modality relations, Single-word answer, Low interpretability
STL-CQA [32]	End-to-end learning, Infers inter-modality relations, Capable of reasoning, Interpretable,	Single-word answer, Relatively complex architecture, Does not consider inter-modality relations,
Charles et. al. [58]	Generates abstract and high detailed captions, Infers inter and intra-modality relations, Capable of reasoning, Handles different chart types,	Two-stage architecture, Relies on OCR performance, Requires large training data, No summarization,

C. LEAF-QA

This dataset [88] contains 250,000 images of different chart types, constructed from real-world open data sources. There are more than 2 million question-answer pairs about the structure, relations, and semantics of the charts. One advantage of this dataset is the use of paraphrases for question-answer pairs which avoids models from memorizing templates [88]. Another advantage of this dataset is the dense annotation for each chart image that includes masks for each chart element (e.g., legend, axes, etc.).

D. FigureSeer

This dataset [95] is similar to FigureQA and contains 60,000 figure images annotated by crowd-workers (a large number of people whom each contribute a small amount of labor) with the focus on answering linguistic questions about the underlying data (the data that can be inferred from the chart image visually). One advantage of FigureSeer is that its plots come from real-world data and are not synthetically created. However, it does not cover question answering that requires reasoning (e.g., estimating AUC, recognizing repeating patterns, etc.).

E. PlotQA

The focus of this dataset [31] is on reasoning tasks and addressing the out-of-vocabulary problem. For this reason, they provided 28.9 million question-answer pairs for 224,377 plots. The plots are created from real-world data sources

TABLE 3. Comparison of publicly available datasets for chart image captioning and CQA tasks.

Dataset	Bar	Pie	Line	Scatter	Box	Size
DVQA [30]	✓	✗	✗	✗	✗	300K
FigureQA [94]	✓	✓	✓	✗	✗	100K
FigureSeer [95]	✓	✗	✓	✓	✓	60K
LEAF-QA [88]	✓	✓	✓	✓	✓	250K
PlotQA [31]	✓	✗	✓	✓	✗	224K
FigCap [29]	✓	✓	✓	✗	✗	100K
SciCap [96]	✓	✓	✓	✓	✓	400K
LineCap [97]	✗	✗	✓	✗	✗	3528

and the questions are generated based on 74 templates. Question-answer pairs are about structural understanding, data retrieval, and reasoning. Answers are divided into three categories: i) yes/no, ii) fixed vocabulary, and iii) out of vocabulary. The fixed vocabulary subset consists of questions and answers that their words are from a limited and fixed vocabulary. On the other hand, the OOV subset contains Q/A pairs with a broader set of words.

F. FigCap

The previously mentioned datasets are all designed for question-answering applications. FigCAP [58] on the other hand is a captioning dataset which is used for generating natural language descriptions for a given chart image. This dataset consists of horizontal and vertical bar charts, line plots, and dotted line plots. Image-caption pairs are divided into two subsets called FigCAP-H and FigCAP-D. Images in both subsets are the same, but the captions are different. Captions in FigCAP-H are about structure and general information of the chart. On the other hand, FigCAP-D contains captions with detailed information and reasoning about the chart images. In FigCAP both images and captions are created synthetically based on the FigureQA dataset. Thus, the image-caption pairs can be generated as many as needed.

G. SciCap

SciCap is a large-scale figure caption dataset [96] based on computer science-related papers in arXiv published between 2010 and 2020. This dataset contains more than 400,000 figures extracted from 290,000 papers. Unlike the other datasets which focus on the underlying data in figures (the data that can be inferred from the chart image), SciCap focuses on captions.

H. LineCap

A set of experimental tests done by Mahinpei et al. [97] on baseline chart captioning models shows that most captioning models perform well in describing single-lined figures, however struggle with complex trends and multi-lined charts. They repeat the same description for all lines in multi-lined charts, even if the lines showed different trends. The authors have introduced a new dataset called LineCap which consists of 3,528 figures with a focus on line charts.

VI. EVALUATION CRITERIA

Evaluating and measuring the quality of a generated caption or explanation for an image is a challenge. For cases where the images and captions are numerous, human evaluation is costly or even impractical. On the other hand, automatic evaluation metrics such as BLEU [65], CIDEr [67], and ROUGE [99] are mostly based on n-gram matching. For instance, BLEU4 metric takes a geometric mean from the precision scores of 1-gram to 4-gram as follows:

$$\mathrm{AP} = \prod_{n=1}^{4} (P_n)^{\frac{1}{4}} = (P_1)^{\frac{1}{4}} (P_2)^{\frac{1}{4}} (P_3)^{\frac{1}{4}} (P_4)^{\frac{1}{4}} , \quad (1)$$

where AP is the Average Precision, and P_1 to P_4 are the matching scores of 1-gram to 4-gram, respectively. One main issue in (1) is that if the generated sentence has only one word, its 1-gram score P_1 becomes 1 which propels the sentence generator model to generate short sentences. To overcome this deficiency, a penalty factor is used to decrease the score of short sentences. Using the penalty factor the $BLEU_N$ score can be calculated by (2):

$$BLEU_N = \text{Penalty} \times \mathrm{AP} = \exp(1 - \frac{k}{l}) \times \prod_{n=1}^{N} (P_n)^{\frac{1}{N}} , \quad (2)$$

where k is the length of ground truth sentence and l is the length of the generated sentence. In some literature, a logarithmic version of BLEU metric is used which is computed based on (3):

$$logBLEU_N = \min(1 - \frac{k}{l}, 0) + \sum_{n=1}^{n} \frac{logP_n}{N} , \quad (3)$$

Although N-gram-based metrics are simple and easy to understand, they only consider word matches and don't consider the meaning of words, while humans use different words with the same meaning. These metrics also ignore the importance of words in a sentence. This issue causes the correlation between human judgment and automatic metrics very low [68].

A more recent metric called SPICE which is specifically proposed for image captioning by Anderson et al [100] is based on similarity of scene graphs. In case of semantic quality, SPICE correlates better with human judgment as compared to previous metrics. Scene graph encodes the semantic representation of a sentence by building a dependency graph containing objects, attributes, and relations. Concepts are then represented by tuples consist of <object, relation, attribute>. After extracting tuples for both target sentence and predicted sentence, SPICE is computed based on F1-Score of matched tuples. SPICE also considers semantic similarity of words by utilizing WordNet [101] to match synonyms.

In VQA tasks, two approaches are used for model designing and evaluation. Several methods use sentence generator or word sequence generator modules as the last stage of their model. In this case, the answer can be a

free-form and open-ended sentence. Other methods use a classifier as the final stage of their model. The model should pick the best matching answer from a fixed set of predefined answers. For methods that generate free-form sentences, image captioning metrics such as BLEU can be employed. For classification-based approaches, common metrics such as accuracy, recall, and F1-Score can be utilized.

VII. CHALLENGES

As this literature review reveals, there are several challenges in chart captioning and question-answering tasks.

A. CHART TYPE VARIABILITY

Graphical and textual components vary in different chart types which makes the design of a general automatic chart understanding model difficult. Each type of chart requires a different approach for interpretation and answering questions. Therefore, the model needs to be flexible and able to handle various chart types accurately. Our investigations show that there is no unified method capable of understanding different chart types accurately. That is why many methods use a chart type classification stage in their pipeline to select the best model for the input chart type. In addition, a chart type can have different design patterns (e.g., visual effects, shadows, fonts, etc.) which makes it more intricate for machine learning models.

B. ACCURATE OBJECT AND TEXT DETECTION

Object and text detection is one of the challenges in IC and CQA methods. CQA systems aim to answer questions based on the information presented in a chart or graph. Therefore, accurate detection of the various elements in the chart, such as axes, legends, titles, data points, and labels, is crucial for the system to correctly interpret the chart and generate a meaningful answer. Object detection involves identifying the different components of a chart and labeling them appropriately. This process can be complicated by factors such as low image quality, complex chart designs, occlusion, and overlapping elements. Inaccurate detection can lead to incorrect labeling and interpretation of the chart, resulting in incorrect or meaningless answers. Text detection, on the other hand, involves recognizing the text in the chart and accurately extracting it. This can be challenging, as the text may be presented in various fonts, sizes, and orientations, and may be occluded by other elements in the chart. Accurate text detection is essential for understanding the content of the chart and generating meaningful answers to questions.

C. ACCURATE OCR

The crucial role of text in chart understanding pushes models to use an OCR module or sub-model in their pipeline. Chart images have a sparse distribution of short text strings in different fonts, orientations, colors, and sizes [12]. Several of the reviewed methods have used OCR tools like Tesseract [46] for text detection, while other methods used custom text detection algorithms based on connected component analysis, texture analysis, and CNNs. Since most chart images are created digitally, text recognition can be done using standard OCR tools like Tesseract [46].

D. DATA EXTRACTION

In many cases, data extraction is the most challenging task especially when it comes to detailed caption generation or answering non-comparative questions (e.g., Where does "Series A" reach the maximum value?). Although in bar and pie charts the data extraction is easier, in line charts and scatter plots where the chart has several series, overlapping/occlusion, non-linear scales, etc., the data extraction becomes more difficult.

E. QUESTION UNDERSTANDING

Many chart captioning methods focus on understanding the text content of chart images which is not enough for complex cases. Answering complex questions involves deep analysis of the information given inside a chart. The system needs to recognize the type of question asked, such as comparison, trend, or relationship, and identify the relevant objects or texts inside the chart image to generate a meaningful answer. For example, the question "Which line has the largest Area-Under-the-Curve in the input chart?" cannot be answered by the textual contents or extracted data alone. It rather needs reasoning that extracts concepts from the chart.

F. ACCURATE DESCRIPTION GENERATION

Generating natural language descriptions is another challenge that requires an accurate language model capable of converting extracted data or visual features to sentences. The language model should be capable of handling the out of vocabulary problem because it is common in real-world chart images to have unseen labels or new abbreviations for data marks and series. Relevancy is another challenge, especially in QA tasks, where the model needs to recognize the relatedness of the input question to the input image. If the input question is completely irrelevant to the image, the model should be able to reject it.

G. LIMITED DATA

CQA and IC methods require large amounts of data for training the models. The training data needs to be cleaned and annotated carefully. However, the availability of annotated data is often limited, which can hinder the performance of the system.

VIII. POTENTIAL FUTURE RESEARCH DIRECTIONS

We have reviewed a structured classification of IC and CQA tasks and arranged the current solutions and evaluation techniques based on the identified problem space. This analysis has enabled us to recognize the significant gaps in the current literature and the potential opportunities for future research in the attractive field of automatic chart image understanding. As a result, in this section we discuss the potential future research directions in this area.

One problem that IC and CQA can be improved for which is the ability of comprehendig complex charts. Many charts contain multiple layers of information, and understanding the relationships between data points and variables in the same level or different levels can be challenging. Researchers could focus on developing algorithms that can analyze and comprehend these complex charts, allowing for more accurate and nuanced understanding and analysis.

Another potential research objective for CQA is to improve the ability of models to understand and analyze natural language questions that are not directly related to the chart or graph. This would require developing algorithms that can analyze the semantic meaning of the question, and then map it to the relevant data points on the chart or graph. Improving the ability of chart question answering systems to understand complex natural language questions can enhance their usefulness and applicability in a wide range of domains.

Paying attention to multimodal nature of chart understanding problem, and leveraging the complementary information in its textual and visual domains, especially by taking advantage of aligning the geometric information in these domains (like [73], [102], [103]), can lead to more successful IC and CQA models in the future. The importance of this approach will be more obvious when taking the positional relationships of the chart image elements into account, using state-of-the-art geometric (deep) learning techniques, like new graph neural networks [104], [105].

Developing newer and more general benchmark datasets will also be an important activity to improve the capabilities of future chart understanding systems, especially the ones that are based on data demanding learning algorithms. Benchmark datasets are essential for evaluating the performance of chart question answering systems. Researchers could focus on developing standardized datasets that can be used to evaluate the performance of different algorithms and compare their effectiveness.

IX. CONCLUSION

In this paper, we reviewed the automatic chart understanding methods, and discussed their approaches to achieve their objectives. According to our investigations, the performance of chart understanding models is directly dependent on object detection and recognition as well as visual feature extraction. For this reason, early methods were mostly based on predefined rules and hand-crafted features with the focus on extracting tabular data from chart images. Additionally, early methods were mostly designed based on multi-stage manners in which each stage works independently. In these methods, each stage designed and tuned for a specific task that makes the entire model less general. Although hand-crafted features work properly in specific narrow tasks, they perform poorly when the image varies in style or even colors. With the advancements of deep learning-based methods and end-to-end training, designing rules is left to the model itself, and thus the model could achieve a more robust and general knowledge for understanding and interpreting the charts.

Furthermore, recognizing relations between image regions and question words are vital in accurate image understanding and QA, respectively. Chart understanding models as multimodal data analysis problems, should consider inter-modality and intra-modality relations, in the visual and textual modalities. Utilizing attention mechanisms, relation maps, and graphs are examples of modeling these inter and intra-modality relations. With the invention of Graph Convolutional Networks (GCN) and their variants, it is expected to see more accurate methods soon.

CONFLICT OF INTERESET

The authors declare that there are no known conflicts of interest associated with this paper.

REFERENCES

[1] A. J. Rajkumar, J. Lazar, J. B. Jordan, A. Darvishy, and H.-P. Hutter, "PDF accessibility of research papers: What tools are needed for assessment and remediation?" in *Proc. 53rd Annu. Hawaii Int. Conf. Syst. Sci.*, 2020, pp. 1–10.

[2] A. Darvishy, T. Leemann, and H.-P. Hutter, "Two software plugins for the creation of fully accessible pdf documents based on a flexible software architecture," in *Computers Helping People With Special Needs*. Linz, Austria: Springer, Jul. 2012, pp. 621–624.

[3] A. Mirkazemy, P. Adibi, S. M. S. Ehsani, A. Darvishy, and H.-P. Hutter, "Mathematical expression recognition using a new deep neural model," Under 2nd Rev. Neural Netw., 2022. [Online]. Available: https://ssrn.com/abstract=4245142

[4] F. M. Schmitt-Koopmann, E. M. Huang, H.-P. Hutter, T. Stadelmann, and A. Darvishy, "FormulaNet: A benchmark dataset for mathematical formula detection," *IEEE Access*, vol. 10, pp. 91588–91596, 2022.

[5] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth, "Every picture tells a story: Generating sentences from images," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2010, pp. 15–29.

[6] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3128–3137.

[7] G. Srivastava and R. Srivastava, "A survey on automatic image captioning," in *Proc. Int. Conf. Math. Comput.* Singapore: Springer, 2018, pp. 74–83.

[8] S. Cao, G. An, Z. Zheng, and Z. Wang, "Vision-enhanced and consensus-aware transformer for image captioning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 10, pp. 7005–7018, Oct. 2022.

[9] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: Lessons learned from the 2015 MSCOCO image captioning challenge," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 652–663, Apr. 2017.

[10] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. 13th Eur. Conf. Comput. Vis.*, vol. 8693, Zurich, Switzerland, Sep. 2014, pp. 740–755.

[11] A. Selivanov, O. Y. Rogov, D. Chesakov, A. Shelmanov, I. Fedulova, and D. V. Dylov, "Medical image captioning via generative pretrained transformers," 2022, *arXiv:2209.13983*.

[12] K. Davila, S. Setlur, D. Doermann, B. U. Kota, and V. Govindaraju, "Chart mining: A survey of methods for automated chart analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 11, pp. 3799–3819, Nov. 2021.

[13] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training—GPT," OpenAI, 2018.

[14] A. A. Yusuf, F. Chong, and M. Xianling, "An analysis of graph convolutional networks and recent datasets for visual question answering," *Artif. Intell. Rev.*, vol. 55, no. 8, pp. 6277–6300, Dec. 2022.

[15] C. Chen, D. Han, and C.-C. Chang, "CAAN: Context-aware attention network for visual question answering," *Pattern Recognit.*, vol. 132, Dec. 2022, Art. no. 108980.

[16] D. Geman, S. Geman, N. Hallonquist, and L. Younes, "Visual Turing test for computer vision systems," *Proc. Nat. Acad. Sci. USA*, vol. 112, no. 12, pp. 3618–3623, Mar. 2015.

[17] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: A collaboratively created graph database for structuring human knowledge," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Jun. 2008, pp. 1247–1250.

[18] C. L. Zitnick, R. Vedantam, and D. Parikh, "Adopting abstract images for semantic scene understanding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 4, pp. 627–638, Apr. 2016.

[19] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3156–3164.

[20] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. NIPS*, 2013, pp. 3111–3119.

[21] C. Wang, Z. Zhou, and L. Xu, "An integrative review of image captioning research," *J. Phys., Conf. Ser.*, vol. 1748, no. 4, 2021, Art. no. 042060.

[22] J. Pavlopoulos, V. Kougia, and I. Androutsopoulos, "A survey on biomedical image captioning," in *Proc. 2nd Workshop Shortcomings Vis. Lang.*, 2019, pp. 26–36.

[23] K. C. Shahira and A. Lijiya, "Document image classification: Towards assisting visually impaired," in *Proc. IEEE Region 10 Conf. (TENCON)*, Oct. 2019, pp. 852–857.

[24] F. Bajić and J. Job, "Review of chart image detection and classification," *Int. J. Document Anal. Recognit.*, vol. 1, pp. 1–22, Jan. 2023.

[25] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2048–2057.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[27] X. Li and S. Jiang, "Know more say less: Image captioning based on scene graphs," *IEEE Trans. Multimedia*, vol. 21, no. 8, pp. 2117–2130, Aug. 2019.

[28] X. Chang, P. Ren, P. Xu, Z. Li, X. Chen, and A. Hauptmann, "A comprehensive survey of scene graphs: Generation and application," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 1–26, Jan. 2023.

[29] X. Qian, E. Koh, F. Du, S. Kim, J. Chan, R. A. Rossi, S. Malik, and T. Y. Lee, "Generating accurate caption units for figure captioning," in *Proc. Web Conf.*, Apr. 2021, pp. 2792–2804.

[30] K. Kafle, B. Price, S. Cohen, and C. Kanan, "DVQA: Understanding data visualizations via question answering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5648–5656.

[31] N. Methani, P. Ganguly, M. M. Khapra, and P. Kumar, "PlotQA: Reasoning over scientific plots," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 1527–1536.

[32] H. Singh and S. Shekhar, "STL-CQA: Structure-based transformers with localization and encoding for chart question answering," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2020, pp. 3275–3284.

[33] T. Eiter, N. Higuera, J. Oetsch, and M. Pritz, "A neuro-symbolic ASP pipeline for visual question answering," *Theory Pract. Log. Program.*, vol. 22, no. 5, pp. 739–754, Sep. 2022.

[34] S. C. Daggubati, J. Sreevalsan-Nair, and K. Dadhich, "BarChartAnalyzer: Data extraction and summarization of bar charts from images," *Social Netw. Comput. Sci.*, vol. 3, no. 6, pp. 1–19, Oct. 2022.

[35] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2018, pp. 4171–4186.

[36] M. Cliche, D. Rosenberg, D. Madeka, and C. Yee, "Scatteract: Automated extraction of data from scatter plots," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Skopje, Macedonia: Springer, 2017, pp. 135–150.

[37] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 886–893.

[38] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, vol. 2, Sep. 1999, pp. 1150–1157.

[39] W. Huang and C. L. Tan, "A system for understanding imaged infographics and its applications," in *Proc. ACM Symp. Document Eng.*, Aug. 2007, pp. 9–18.

[40] S. R. Choudhury, S. Wang, P. Mitra, and C. L. Giles, "Automated data extraction from scholarly line graphs," in *Proc. Int. Workshop Graph. Recognit.*, 2015.

[41] *Plot Digitizer*. Accessed: Mar. 20, 2023. [Online]. Available: http://plotdigitizer.sourceforge.net/

[42] *Auto-Trace*. Accessed: Mar. 20, 2023. [Online]. Available: http://autotrace.sourceforge.net/

[43] M. Savva, N. Kong, A. Chhajta, L. Fei-Fei, M. Agrawala, and J. Heer, "ReVision: Automated classification, analysis and redesign of chart images," in *Proc. 24th Annu. ACM Symp. User Interface Softw. Technol.*, Oct. 2011, pp. 393–402.

[44] V. S. N. Prasad, B. Siddiquie, J. Golbeck, and L. S. Davis, "Classifying computer generated charts," in *Proc. Int. Workshop Content-Based Multimedia Indexing*, Jun. 2007, pp. 85–92.

[45] A. Mishchenko and N. Vassilieva, "Chart image understanding and numerical data extraction," in *Proc. 6th Int. Conf. Digit. Inf. Manage.*, Sep. 2011, pp. 115–120.

[46] A. Kay, *Tesseract: An Open-Source Optical Character Recognition Engine*, vol. 159. Houston, TX, USA: Belltown Media, 2007.

[47] P. Mishra, S. Kumar, M. K. Chaube, and U. Shrawankar, "ChartVi: Charts summarizer for visually impaired," *J. Comput. Lang.*, vol. 69, Apr. 2022, Art. no. 101107.

[48] J. Sreevalsan-Nair, K. Dadhich, and S. C. Daggubati, "Tensor fields for data extraction from chart images: Bar charts and scatter plots," in *Topological Methods in Data Analysis and Visualization VI*. Springer-Verlag, 2021, pp. 219–241.

[49] W. Huang, C. L. Tan, and W. K. Leow, "Associating text and graphics for scientific chart understanding," in *Proc. 8th Int. Conf. Document Anal. Recognit. (ICDAR)*, 2005, pp. 580–584.

[50] A. P. Deshpande and C. Mahender, "Summarization of graph using question answer approach," in *Proc. ICT4SD*, 2020, pp. 205–216.

[51] J. Choi, S. Jung, D. G. Park, J. Choo, and N. Elmqvist, "Visualizing for the non-visual: Enabling the visually impaired to use visualization," *Comput. Graph. Forum*, vol. 38, no. 3, pp. 249–260, Jun. 2019.

[52] C. Greenbacker, P. Wu, S. Carberry, K. F. McCoy, and S. Elzer, "Abstractive summarization of line graphs from popular media," in *Proc. Workshop Autom. Summarization Different Genres, Media, Lang.*, 2011, pp. 41–48.

[53] R. Kiros, R. Salakhutdinov, and R. Zemel, "Multimodal neural language models," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 595–603.

[54] R. Kiros, R. Salakhutdinov, and R. S. Zemel, "Unifying visual-semantic embeddings with multimodal neural language models," *Trans. Assoc. Comput. Linguistics*, pp. 1–13, 2015.

[55] P. Sountsov and S. Sarawagi, "Length bias in encoder decoder models and a case for global conditioning," 2016, *arXiv:1606.03402*.

[56] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, T. Darrell, and K. Saenko, "Long-term recurrent convolutional networks for visual recognition and description," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2625–2634.

[57] J. Jin, K. Fu, R. Cui, F. Sha, and C. Zhang, "Aligning where to see and what to tell: Image caption with region-based attention and scene factorization," 2015, *arXiv:1506.06272*.

[58] C. Chen, R. Zhang, E. Koh, S. Kim, S. Cohen, and R. Rossi, "Figure captioning with relation maps for reasoning," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 1537–1545.

[59] A. Spreafico and G. Carenini, "Neural data-driven captioning of time-series line charts," in *Proc. Int. Conf. Adv. Vis. Interfaces*, Sep. 2020, pp. 1–5.

[60] I. Safder, H. Batool, R. Sarwar, F. Zaman, N. R. Aljohani, R. Nawaz, M. Gaber, and S.-U. Hassan, "Parsing AUC result-figures in machine learning specific scholarly documents for semantically-enriched summarization," *Appl. Artif. Intell.*, vol. 36, no. 1, pp. 1–27, Dec. 2022.

[61] J. Obeid and E. Hoque, "Chart-to-text: Generating natural language descriptions for charts by adapting the transformer model," in *Proc. 13th Int. Conf. Natural Lang. Gener. (INLG)*, Dublin, Ireland, Dec. 2020, pp. 138–147.

[62] S. Kantharaj, R. T. Leong, X. Lin, A. Masry, M. Thakkar, E. Hoque, and S. Joty, "Chart-to-text: A large-scale benchmark for chart summarization," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics*. Cedarville, OH, USA: Association for Computational Linguistics, 2022, pp. 4005–4023.

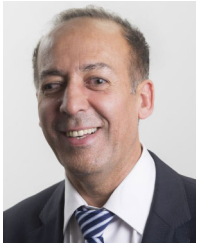
[63] C. Liu, L. Xie, Y. Han, D. Wei, and X. Yuan, "AutoCaption: An approach to generate natural language description from visualization automatically," in *Proc. IEEE Pacific Vis. Symp. (PacificVis)*, Jun. 2020, pp. 191–195.
[64] A. Lundgard and A. Satyanarayan, "Accessible visualization via natural language descriptions: A four-level model of semantic content," *IEEE Trans. Vis. Comput. Graphics*, vol. 28, no. 1, pp. 1073–1083, Jan. 2022.
[65] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguistics*, 2001, pp. 311–318.
[66] C.-Y. Lin and E. Hovy, "Automatic evaluation of summaries using N-gram co-occurrence statistics," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics Human Lang. Technol.*, 2003, pp. 150–157.
[67] R. Vedantam, C. L. Zitnick, and D. Parikh, "CIDEr: Consensus-based image description evaluation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4566–4575.
[68] T. Baltrusaitis, C. Ahuja, and L.-P. Morency, "Multimodal machine learning: A survey and taxonomy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 2, pp. 423–443, Feb. 2019.
[69] W. Zhang, J. Yu, H. Hu, H. Hu, and Z. Qin, "Multimodal feature fusion by relational reasoning and attention for visual question answering," *Inf. Fusion*, vol. 55, pp. 116–126, Mar. 2020.
[70] A. Das, H. Agrawal, L. Zitnick, D. Parikh, and D. Batra, "Human attention in visual question answering: Do humans and deep networks look at the same regions?" *Comput. Vis. Image Understand.*, vol. 163, pp. 90–100, Oct. 2017.
[71] J.-H. Kim, J. Jun, and B.-T. Zhang, "Bilinear attention networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1571–1581.
[72] P. Gao, Z. Jiang, H. You, P. Lu, S. C. H. Hoi, X. Wang, and H. Li, "Dynamic fusion with intra- and inter-modality attention flow for visual question answering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6639–6648.
[73] M. Behmanesh, P. Adibi, J. Chanussot, C. Jutten, and S. M. S. Ehsani, "Geometric multimodal learning based on local signal expansion for joint diagonalization," *IEEE Trans. Signal Process.*, vol. 69, pp. 1271–1286, 2021.
[74] J. Cao, X. Qin, S. Zhao, and J. Shen, "Bilateral cross-modality graph matching attention for feature fusion in visual question answering," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Feb. 7, 2022, doi: 10.1109/TNNLS.2021.3135655.
[75] Z. Zhu, J. Yu, Y. Wang, Y. Sun, Y. Hu, and Q. Wu, "Mucko: Multi-layer cross-modal knowledge reasoning for fact-based visual question answering," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 1097–1103.
[76] M. Welling and T. N. Kipf, "Semi-supervised classification with graph convolutional networks," in *Proc. 5th Int. Conf. Learn. Represent. (ICLR)*, Toulon, France, Apr. 2017, pp. 1–14.
[77] C. Gao, Q. Zhu, P. Wang, H. Li, Y. Liu, A. Van den Hengel, and Q. Wu, "Structured multimodal attentions for TextVQA," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 12, pp. 9603–9614, Dec. 2022.
[78] A. Singh, V. Natarajan, M. Shah, Y. Jiang, X. Chen, D. Batra, D. Parikh, and M. Rohrbach, "Towards VQA models that can read," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8317–8326.
[79] X. Li, B. Wu, J. Song, L. Gao, P. Zeng, and C. Gan, "Text-instance graph: Exploring the relational semantics for text-based visual question answering," *Pattern Recognit.*, vol. 124, Apr. 2022, Art. no. 108455.
[80] Z. Wan and H. He, "AnswerNet: Learning to answer questions," *IEEE Trans. Big Data*, vol. 5, no. 4, pp. 540–549, Dec. 2019.
[81] F. Liu, J. Liu, Z. Fang, R. Hong, and H. Lu, "Visual question answering with dense inter- and intra-modality interactions," *IEEE Trans. Multimedia*, vol. 23, pp. 3518–3529, 2021.
[82] L. Peng, Y. Yang, Z. Wang, Z. Huang, and H. T. Shen, "MRA-Net: Improving VQA via multi-modal relation attention network," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 318–329, Jan. 2022.
[83] A. Mishra, S. Shekhar, A. K. Singh, and A. Chakraborty, "OCR-VQA: Visual question answering by reading text in images," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 947–952.
[84] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola, "Stacked attention networks for image question answering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 21–29.
[85] P. Pasupat and P. Liang, "Compositional semantic parsing on semi-structured tables," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Lang. Process.* Cedarville, OH, USA: Association for Computational Linguistics, 2015, pp. 1470–1480.
[86] M. Sharma, S. Gupta, A. Chowdhury, and L. Vig, "ChartNet: Visual reasoning over statistical charts using MAC-networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–7.
[87] D. A. Hudson and C. D. Manning, "Compositional attention networks for machine reasoning," in *Proc. 6th Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–20.
[88] R. Chaudhry, S. Shekhar, U. Gupta, P. Maneriker, P. Bansal, and A. Joshi, "LEAF-QA: Locate, encode & attend for figure question answering," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, Mar. 2020, pp. 3512–3521.
[89] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2961–2969.
[90] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.
[91] M. Levy, R. Ben-Ari, and D. Lischinski, "Classification-regression for chart comprehension," in *Computer Vision—ECCV 2022*. Cham, Switzerland: Springer, Oct. 2022.
[92] D. H. Kim, E. Hoque, and M. Agrawala, "Answering questions about charts and generating visual explanations," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, Apr. 2020, pp. 1–13.
[93] R. Reddy, R. Ramesh, A. Deshpande, and M. M. Khapra, "FigureNet : A deep learning model for question-answering on scientific plots," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.
[94] S. Ebrahimi Kahou, V. Michalski, A. Atkinson, A. Kádár, A. Trischler, and Y. Bengio, "FigureQA: An annotated figure dataset for visual reasoning," 2017, *arXiv:1710.07300*.
[95] N. Siegel, Z. Horvitz, R. Levin, S. Divvala, and A. Farhadi, "FigureSeer: Parsing result-figures in research papers," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 664–680.
[96] T.-Y. Hsu, C. L. Giles, and T.-H. K. Huang, "SciCap: Generating captions for scientific figures," 2021, *arXiv:2110.11624*.
[97] A. Mahinpei, Z. Kostic, and C. Tanner, "LineCap: Line charts for data visualization captioning models," in *Proc. IEEE Vis. Vis. Anal. (VIS)*, Oct. 2022, pp. 35–39.
[98] Bokeh Development Team. *Bokeh: Python Library for Interactive Visualization*. [Online]. Available: https://bokeh.pydata.org/en/latest/
[99] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Proc. Workshop Text Summarization ACL*, Barcelona, Spain, 2004, pp. 74–81.
[100] P. Anderson, B. Fernando, M. Johnson, and S. Gould, "SPICE: Semantic propositional image caption evaluation," in *Proc. 14th Eur. Conf. Comput. Vis.*, Amsterdam, The Netherlands, Oct. 2016, pp. 382–398.
[101] G. A. Miller, "WordNet: A lexical database for English," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, 1995.
[102] A. Pournemat, P. Adibi, and J. Chanussot, "Semisupervised charting for spectral multimodal manifold learning and alignment," *Pattern Recognit.*, vol. 111, Mar. 2021, Art. no. 107645.
[103] M. Behmanesh, P. Adibi, J. Chanussot, and S. M. S. Ehsani, "Cross-modal and multimodal data analysis based on functional mapping of spectral descriptors and manifold regularization," 2021, *arXiv:2105.05631*.
[104] M. Behmanesh, P. Adibi, S. M. S. Ehsani, and J. Chanussot, "Geometric multimodal deep learning with multiscaled graph wavelet convolutional network," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Oct. 25, 2022, doi: 10.1109/TNNLS.2022.3213589.
[105] J. Gaskell, N. Campioni, J. M. Morales, D. Husmeier, and C. J. Torney, "Inferring the interaction rules of complex systems with graph neural networks and approximate Bayesian computation," *J. Roy. Soc. Interface*, vol. 20, no. 198, Jan. 2023, Art. no. 20220676.

ALI MAZRAEH FARAHANI received the B.S. degree in software engineering from TVU, Iran, in 2013, and the M.S. degree in artificial intelligence from the Shahid Bahonar University of Kerman, Iran, in 2016. He is currently pursuing the Ph.D. degree in artificial intelligence with the University of Isfahan, Isfahan, Iran.

PEYMAN ADIBI received the Ph.D. degree from the Faculty of Computer Engineering, Amirkabir University of Technology, Tehran, Iran, in 2009. Since 2010, he has been with the Artificial Intelligence Department, Faculty of Computer Engineering, University of Isfahan, Isfahan, Iran, where he is currently an Associate Professor and the Head of the Intelligent and Learning Systems (ILS) Research Laboratory. His current research interests include machine learning and pattern recognition, computer vision and image processing, computational intelligence and soft computing, multimodal and geometric data analysis, and their applications.

MOHAMMAD SAEED EHSANI received the B.Sc. and M.Sc. degrees in communication engineering from the School of Electrical and Computer Engineering, College of Engineering, University of Tehran, Tehran, Iran, in 1988 and 1991, respectively, the Ph.D. degree in computer science from the School of Engineering, Kennedy Western University, Los Angeles, CA, USA, in 2004, and the D.L. degree (Hons.) from Cambridge University, Cambridge, U.K., in 2015. He is currently a member of the Artificial Intelligence Department, Faculty of Computer Engineering, University of Isfahan, Isfahan, Iran. His areas of research include pattern recognition, neuro-fuzzy, rule-base control, next-generation networks (NGNs), big data, and cybernetics. He is a Fellow Member of the American Biographical Institute (ABI)/International Biographical Center (IBC) and a member of IEEE's technical committees.

HANS-PETER HUTTER received the Doctor of Technical Science degree in electrical engineering from ETH Zürich (ETHZ), in 1997. In 1997, he worked on hybrid HMM/ANN approaches to speech recognition over telephone lines. He joined the UBS Ubilaboratory as a Postdoctoral Researcher, where he worked on a European project for HMM-based speaker identification over the telephone. At the same time, he was a Co-Lecturer at ETHZ in two speech processing modules. In 1997, he joined the Zurich University of Applied Sciences (ZHAW), Winterthur, where he was a Professor in computer science on various projects in the area of speech recognition and user-centered design of graphical and voice user interfaces. In 2005, he founded the ZHAW School of Engineering, Institute of Applied Information Technology (InIT), together with his colleagues, and was the Head of the Institute, until 2010. At the same time, he was also the Head of the Human-Information Interaction Group, InIT, which he is still leading today.



ALIREZA DARVISHY is currently a Professor in ICT accessibility and the Head of the ICT Accessibility Laboratory at Zurich University of Applied Sciences, Switzerland. He serves as an Independent Reviewer for European research projects, such as the Active Assisted Living (AAL) Program. He is a Principle Investigator of the "Accessible Scientific PDFs for All" Project, funded by the Swiss National Science Foundation.