# AutoCaption: An Approach to Generate Natural Language Description from Visualization Automatically

Can Liu[1] *     Liwenhan Xie[1] †     Yun Han[1] ‡     Datong Wei[1] §     Xiaoru Yuan[1,2] ¶

1) Key Laboratory of Machine Perception (Ministry of Education), and School of EECS, Peking University, Beijing, China
2) National Engineering Laboratory for Big Data Analysis and Application, Peking University, Beijing, China

## ABSTRACT

In this paper, we propose a novel approach to generate captions for visualization charts automatically. In the proposed method, visual marks and visual channels, together with the associated text information in the original charts, are first extracted and identified with a multilayer perceptron classifier. Meanwhile, data information can also be retrieved by parsing visual marks with extracted mapping relationships. Then a 1-D convolutional residual network is employed to analyze the relationship between visual elements, and recognize significant features of the visualization charts, with both data and visual information as input. In the final step, the full description of the visual charts can be generated through a template-based approach. The generated captions can effectively cover the main visual features of the visual charts and support major feature types in commons charts. We further demonstrate the effectiveness of our approach through several cases.

## 1 INTRODUCTION

Visualization designers create natural language caption (NLC) to interpret visualization for a better propagation among users. NLC is helpful in conveying crucial information on visualizations in data news and academic papers. However, the visualization community pays little attention to generate NLC automatically, which is helpful when lots of visualization do not have NLC. The automatic generation of NLC would be useful for data analysts like data news designer and stock analysts. The reason why there is no such algorithm to generate NLC directly is that natural language involves a higher level of human cognition, while traditional methods are hard to handle the complex task to summarize a visualization. Nowadays, the development of deep learning technology brings the possibility to deal with this problem. Deep learning methods have shown their ability in a similar task like image captioning [16, 19], which is similar to visualization captioning task. Automatic image captioning also brings significant benefits to image understanding and retrieval. Based on this, deep learning technology should have the potential to solve the automatic generation of NLC in visualization. However, to generate natural language from a visualization chart requires a long term exploration. As the first step of exploration, we present a machine learning-based pipeline to generate natural language caption automatically from a visualization.

Our approach supports charts in vector format (e.g., SVG format), which is a standard format in nowadays social media and online news feeds. Additionally, our approach can be extended to bitmap format charts leveraging Object Detection and OCR techniques.

It is easy to recognize visual elements because they are regular shapes like rectangles and circles. The learning model detects crucial features of the chart, each of which is defined by feature type and grouped elements. A template-based method converts these features into natural sentences. Then we construct a caption for the chart leveraging the extracted weighted facts from the model.

To validate the effectiveness of this model, we experiment with bar charts, line charts, and scatter plots of various forms. From the result of validating dataset and tests on specific synthetic data, it is shown that the model can extract crucial features. We then deploy the pre-trained model onto a demo system and conduct a survey to people from different areas. The system supports accepting an uploaded chart. The system supports the exploration of each generated sentence in the caption with their corresponding elements highlighted in the chart display view.

In this work, the primary contributions are three-fold:

- We introduce a novel approach to generate captions for information charts with a deep learning-powered scheme, which learns the noteworthy features in accordance with human perception.
- We design a new model that is extensible to different charts with various suites of summary templates.
- Our prototype system demonstrates a potential usage of our approach in education and data overviews combining with practical annotations.

## 2 RELATED WORK

Our work draws on prior researches in the area of chart interpretation, automation in visualization, and deep learning for caption generation in natural images.

It has long been the interest of the computational linguistics community to generate a text summary for an information graphic automatically, as this would promisingly improve the data accessibility of multimodal documents [3]. Due to the challenge of a wide diversity in chart styles, most works focus on a certain type of chart modeling its context with rules and then exploring approaches to generate summaries, such as line charts [18], bar charts [4], and pie charts [7]. In most works, rules are obtained through a series of perception cognition experiments [6]. The influential work from Mittal et al. [10] propose a method to describe the visual mapping in natural language, but they require users to give information about the mapping relationships. Al-Zaidy et al. [1] use a semantic-graph representation for bar charts and then generate the summary by the "protoform" method [20]. Bryan et al. [2] use pre-defined scores for types of summaries to generate annotations for temporal charts. Hullman et al. also provides rule-based annotations for stock visualization [9]. There is no previous approach to learn the chart features automatically. However, there is an increasing interest in automatic knowledge discovery from the database [5, 15]. Additionally, many industrial products like Arria[1], Google Sheets[2], etc. employ Natural

---

*e-mail: can.liu@pku.edu.cn
†e-mail: liwenhan.xie@pku.edu.cn
‡e-mail: yun.han@pku.edu.cn
§e-mail: datong.wei@pku.edu.cn
¶e-mail: xiaoru.yuan@pku.edu.cn (corresponding author)

---

[1]Arria: https://www.arria.com/core-tech/
[2]Google Sheets: https://www.google.com/sheets/about/

Language Generation (NLG) techniques to suggest such insights. In general, they use a suite of algorithms to find out statistically important results and generate natural language with a pre-defined template [12].

In addition to the efforts above, several works devoted to facilitating chart understanding with additive annotations, where automatic chart interpretation is also necessary. Srinivasan et al. [14] obtains the inferential data facts from a set of heuristics, and then suggest visualizations and embellishments to users after analyzing relationships between common descriptions, charts, and annotations.

## 3 METHOD

Two things are important to generate a caption of visualization, which are getting the content and presenting the content using natural language. We first have a step that parses the mapping relationship. For this step, there have been many previous works to parse a visualization into underlying data [11, 13]. Our main contribution is not revealing the underlying data, so we adopt existed method to handle this part [11]. Data information is recovered by parsing the visual information of data elements with the mapping relationship.

The next steps are to extract the feature content and generate the sentence. The process can be divided into two parts– **Chart feature detection.** We proposed a deep-learning model which takes the visual and data information as input and finally generate the chart feature. Each chart feature has a chart feature type and divides the elements into several groups. **Sentence generation.** We provide a default template for each type of chart feature. We use the output elements of the model to calculate the filling template components.
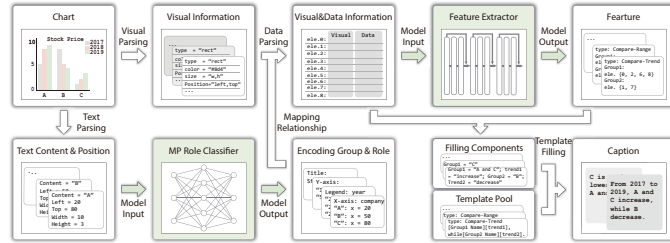


Figure 1: The work flow of our approach: A feature extraction model takes the visual and data information as input and outputs the chart feature with **type** and grouped elements. The chart feature type decides which summary template to use, while grouped elements decide the content of the filling components needed by the templates. The caption with several descriptions is generated.

### 3.1 Item Definition

To describe the overall pipeline better, we first define the following concepts.

- **Data elements** is the elements in the charts that have the corresponding data item in the underlying data. Take the bar charts as an example. The bars are the data elements. The data elements can be presented as $V = \{e_{1m}, e_{2m}, ...e_{nm}\}$ where $e_{im} = \{attrs_{visual}\}$, $attrs_{visual}$ is the visual information, i.e., position, color, opacity, etc.

- **Chart feature** is the content of a sentence. A chart feature has a feature type and several groups of data elements. The type corresponds to a certain template, while The related elements of a feature are composed of grouped elements; each group is a subset of data elements. Formally, $Feature = \{\{G_i\}, type\}$, where $G_i = \{e_{f1}, e_{f2}, ..., e_{fm}\}$ and $C = \{e_{c1}, e_{c2}, ..., e_{ck}\}$, where $0 \le m, k \le n$, $G_i \in V_e$. Information charts mainly show features in the following types. **Aggregation** reduce the data

| Attr. | Channels | Component | Enc. | N |
|-------|----------|-----------|------|---|
| **Visual** | color | Red | [0, 1] | 1 |
| | | Green | [0, 1] | 1 |
| | | Blue | [0, 1] | 1 |
| | position | width | [0, 1] | 1 |
| | | height | [0, 1] | 1 |
| | | up | [0, 1] | 1 |
| | | down | [0, 1] | 1 |
| | | left | [0, 1] | 1 |
| | | right | [0, 1] | 1 |
| | opacity | opacity | [0, 1] | 1 |
| | type | is rect | 1-Hot | 3 |
| | | is circle | 1-Hot | |
| | | is text | 1-Hot | |
| **Data** | quantitative | value | [0, 1] | 1 |
| | ordinal | dimension | 1-Hot | MAX |
| | categorical | dimension | 1-Hot | MAX |

Table 1: Tensor encoding of visual and data attributes. **Enc.** is the encoding method, which can be normalization ([0, 1]) or one-hot encoding. **MAX** is the maximum possible choice number of the one-hot vector. **N** is the number of encoded channels.

item number by some statistic values such as maximum, minimum, average, or sum. **Trend** shows how quantitative value changes by the ordinal attributes. **Comparison** presents the difference, which often grabs the users' attention. **Distribution** like the cluster and outliers are the patterns user often observed. Table 2 list the common features and the corresponding description template and example.

- **Description:** A description is a natural language presentation of a chart feature, which is generated from the chart type and relevant elements list through our summarized templates. The details of translating the chart feature into a description will be described in section 3.3.

- **Caption:** A caption is composed of several chart feature descriptions.

### 3.2 Chart Feature Detection

This part is to extract the **chart feature** from the chart. He et al. [8] introduces the residual network (Res-Nets) in the task of 2D-image recognition, which can well handle the deeper networks which well handled the problem of gradient vanishing. The residual networks have direct skip connections across several layers. Unlike the original Res-Nets with two-dimension layers that deal with 2D-image tasks, our model tackles the 1D-tensors. Residual CNN keeps the sizes of the tensor, which can handle charts with different number of elements. A desired underlying mapping that encourages the model to approximate is $H(x)$. Now with the direct connection sent the input $x$ directly to the output, the stacked components only need to approximate $F(x)$, where $H(x) = F(x) + x$. $F(x)$ learns the residual value, which largely reduces the learning complexity.

Each component in the block has a 1D-convolutional layer, a batch normalization layer, and a leaky ReLu layer. Batch Normalization layer normalizes the output of the convolutional layer to make the signal of tensors more outstanding.

Taking an element list as input, the model can analyze the relationship and generate some global features after training. As Figure 2 shows, the input tensor, and the output tensor are 1-D tensors with an element dimension. In the input tensor, each line is an element. The input shows the visual information and the data information, while the output presents which group it belongs to. The information of the visual and data information of each element is encoded in the input tensor. Table 1 presents how to encode the attributes into tensors. Basically, the quantitative attributes are normalized to a range of [0, 1], the categorical attributes are encoded using the one-hot method with a *MAX* channel number. The output of the model is a list of several features, and each feature classifies the elements into several

| | Type | Default Template | Case |
|---|---|---|---|
| Basical | Absolute | The {value name} in {atribute} is {value} | *The temperature in 14:00 is 27 degrees.* |
| Complex | Trend | The {value name} goes {trend} to {end value} by {end ordinal} | *The temperature goes up to 27 degrees by 14:00, then goes down to 12 degrees by midnight.* |
| | Maximum | The {Group 1 Name} has {extreme relationship} in {share attribute} | *The U.S. has the highest GDP value.* |
| | Minimum | The {Group 1 Name} has {extreme relationship} in {share attribute} | *The apple has the lowest price.* |
| | Cluster | There are {cluster number} clusters around {position}. | *There is a cluster around (3, 5).* |
| | Outlier | There are {outlier number} outliers in total, which are {outlier positions}. | *There are two outliers in total, which are (1.88, 1.23) and (1.63, 1.1).* |
| | Compare | The {value name} of {Group 1 Name} is {degree} {relationship} {Group 2 Name} in {share attribute} | *The price of the apple is higher than orange in 2019.* |
| Second-Order | Trend-Global | The {value name} goes {trend} from {begin value} to {end value} | *The population of the world goes up from 2010 to 2018.* |
| | Trend-local | The {value name} of {Group 1 Name} have {local trend} in {focal ordinal} | *The population of the world has a sudden decrease in the 1940s.* |
| | Trend-Sum | The sum {value name} of {Group 1 Name} goes {trend} from {begin ordinal} to {end ordinal} | *The sum sales of games, DVD, and video goes up from 2000 to 2003.* |
| | Trend-Sum-local | The sum {value name} of {Group 1 Name} have {local trend} in {focal ordinal} | *The sum population of Europe and Africa has a sudden decrease in 1944.* |
| | Compare-Trend | The {value name} in {Group 1 Name} goes {focal trend} from {begin focal value} to {end focal value}, while {Group 2 Name} goes {contextual trend} | *The population in Asia goes up from 2010 to 2020, while it decreases in Europe.* |
| | Compare-Sum | The {value name} of {Group 1 Name} is {degree} {relationship} {Group 2 Name} in {share attribute} | *The sum GDP of France and Germany in 2019 is much higher than that in 1950.* |

Table 2: Default summarized data feature types and corresponding description type. They are categorized into zeroth-order, first-order, second-order features. In the output descriptions, the predicate changes between singular and plural form as the subject changes.

groups. The output tensor stores each feature's grouped information, i.e., which group the element belongs to. For example, the right of the Figure 2 shows Element 4 in Feature 1 belongs to Group 1, while in Feature 2 belongs to Group 3.
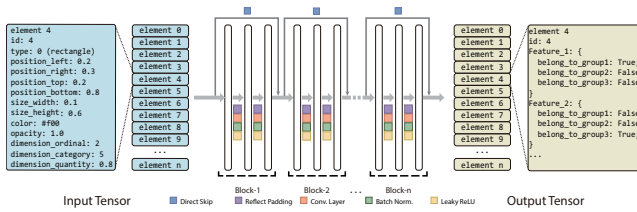


Figure 2: The generator of the deep learning part. it takes the data element with visual and data information as input, then outputs the features. Here the data elements and the features are aligned into the tensor format, both of them are the 1-dimension (elements dimension) tensors with a channel dimension. For both of the data elements and the fact, each line represents an element.

## 3.3 Description Generation

The description content is determined by the feature, which is defined by element groups and chart feature type. While the summary template is analogous to the skeleton of the description, the chart feature decides the sprite of the description.

Based on the feature types above and description type, Table 2 shows the common feature and descriptions. Based on the feature types above, we craft some templates for description generation, following the main idea from Wilbik et al. [17]. The type of feature decides a feature template. We calculate components required by the template based on the output element groups.

For example, the *Group*1 is <State A, 2017, 4%> and the *Group*2 is <State B, 2017, 3%>, and the type is **compare**. With the template

{**Group1 name**} is {**degree**} {**relationship**} {**Group2 name**} in {**share attribute**},

We derive the fact in natural language from the feature. Specifically, the *Group*1 *name* here is *State A*, and the *Group*2 *name* refers to *State B*. The degree and the relationship can be calculated using the quantitative value of these two elements group. Then the description will be generated as follows: *"The value of State A is higher than State B in 2017"*. The degree and the relationship are calculated when given the value of the grouped elements. Then, based on these examples, replacing the objects and relations in the description, we obtain a caption for this fact.

## 4 TRAINING

There is a vast space for designing charts. Accordingly, the machine learning model needs a large amount of data to cover the possible space. To make the visualization type cover the common visualization, we choose to apply our model in Ordinal-Category-Quantity data and Category-Quantity-Quantity data. Which can support the following types of charts: (1) group bar chart; (2) stacked bar chart; (3) line chart; (4) scatter plot. They can cover all typical attribute types and have the potential to involve most feature types.

### 4.1 Hardware and Schedule

Our model is deployed on a symmetric multiprocessing node, equipped with 2 NVIDIA 1080Ti graphics cards, with 11GB memory each. Data is stored on a local disk with 1TB capacity. Currently, our model is in graphics cards with 12 Res-Net blocks, the channel size is 384, and the size of the training dataset is 3000. It takes 213 seconds for each epoch on average, and totally 11.8 hours for 200 epochs.

### 4.2 Results

From the precision of predicted results using 250 test data, We use element-level precision and recall to verify the learning result.

$$precision = \frac{|\{predicted\ elements\} \cap \{ground\ truth\ elements\}|}{|\{predicted\ elements\}|},$$

$$recall = \frac{|\{predicted\ elements\} \cap \{ground\ truth\ elements\}|}{|\{ground\ truth\ elements\}|}.$$

The mean precision of our model predicting focal elements is 86.8%, while the mean recall is 97.0%. For contextual elements, the mean precision is 85.7%, and the mean recall is 98.1%. It is confirmed that the trained model can predict most of the focal and contextual elements.
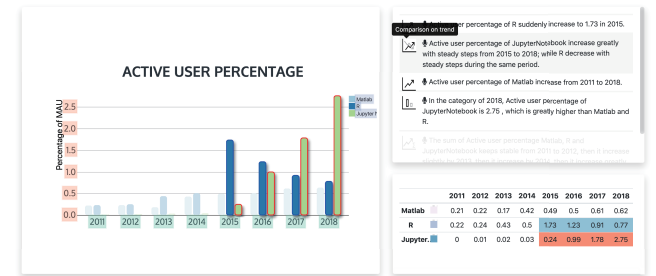


Figure 3: Our proposed interactive system supports generating text captions for target charts. Left: Visualization in SVG format. Top Right: The caption composed of several sentences are automatically synthesized by our method. Bottom Right: Underlying data in the tabular form is corresponding to the chart on the left. When users upload a chart in SVG format, after the visual information parsed, the encoding are extracted, and the data informatioin is recovered, Relevant caption would be generated automatically.
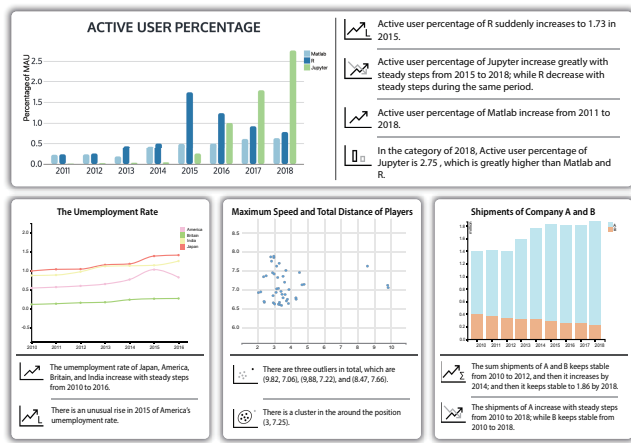
Figure 4: Some example cases of generated caption of the chart. The chart type includes the bar chart, the scatterplot, and the line chart. The generated sentences includes comparison, trend, distribution, and high-order features.

| Chart Type | Sig. | Cover. | Difference (Count) |
|---|---|---|---|
| Group Bar | 95% | 87% | extremes(2) |
| Stacked Bar | 97% | 92% | extremes(2) |
| Line | 93% | 80% | fluctuation(3), relative pos(1) |

Table 3: The result of the significance (Sig.), coverage (Cover.) indices and details of the difference. The rate is considered as the ratio of the sentences sharing the same sense between machine captions and manual captions. Generally, our approach captures what people see in a chart. However, the extremes, fluctuation, and relative position among categories are sometimes neglected.
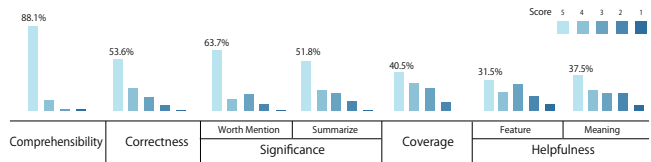


Figure 5: The score distribution of criteria in different aspects. We use a Likert scale: 5 - "strongly agree", 4 - "agree", 3 - "neutral", 2 - "disagree" and "strongly disagree". Most participants are satisfied with the generated captions.

## 5 EVALUATION

After the training procedure, we deploy the pre-trained model into our demo system as shown in Figure 3. Figure 4 shows the captions generated from some charts, which verify the effectiveness of the model in a sense. As there is no ready-to-use public tool for chart summarization or baseline dataset to the best of our knowledge, we evaluate our approach through carefully-designed user interviews based on the interactive system.

### 5.1 Survey

We recruit 15 volunteers for interviews, including a data scientist, an info-graphics designer, a UI designer, an editor, an automation technician, two IT engineers, five college students and three professors from different disciplines like computer science, Chinese language, etc.

In the second part of the survey, participants are required to make simple captions freely as well as to write descriptive sentences under certain constraints like emphasizing trends, comparisons, extremes, and so on. The simple caption represents the fact regarded most important in the chart, while the other requirement helps us to find out the similarities and differences between the machine-generated captions and manual captions. The summarized result is shown in Table 3. From a statistical perspective, the result of our approach highly accords with most people in the study. However, compared with these manually made captions, some kinds of data facts are neglected by our model. For instance, for the line chart used in the study, three of the fifteen participants describe the wild fluctuation spontaneously while the machine missed. Moreover, in the case of bar charts, our system scores the descriptive sentence about extremes low, yet two people choose the extreme as the prime fact. This is relevant to our crowd-sourcing data collecting strategies, which results in the machine's preference for the majority.

### 5.2 Questionnaire

To quantitatively evaluate the captions produced by our approach, we send out online questionnaires and receive 50 valid responses. A majority of the responses come from college students in different disciplines, but there are also software engineers, chemistry engineers, product managers, college professors, librarians, etc.

#### 5.2.1 Design

We mainly assess the effectiveness of our approach from the following five progressive aspects. Moreover, statements in the questionnaire are noted in *italic*. Participants are required to give a five-order-magnitude rating of their opinions for each statement, where 5 stands for "strongly agree", 4 for "agree", 3 for "neutral", 2 for "disagree", and 1 for "strongly disagree".

1. **Comprehensibility:** The degree to which the user understands the caption. *I understand what the caption means.*
2. **Correctness:** The correctness of the content described by the caption. *All the generated facts in the caption are correct.*
3. **Significance:** The degree to which the caption deserves to be described. *All the generated facts in the caption is worth mentioning.*
4. **Coverage:** The degree to which the important facts are described. *The caption mentions all the facts that I think is important in the chart.*
5. **Helpfulness:** To what extent the caption can help the users. *The caption provide some information that helps me better understand the chart.*

Based on the criteria mentioned above, our current implemented learning model focuses on the Ordinal-Category-Quantity (OCQ) as well as the Quantity-Quantity(QQ) data. There are ten charts in all, including group bar charts, stacked bar charts, line charts, and scatter plots. At the end of the questionnaire, we leave a comment box for any reviews.

#### 5.2.2 Feedback

After collecting the questionnaires and doing analysis, we gain more insights into both the advantages and drawbacks of our method. The summary is shown in Figure 5. From the bar charts of Figure 5 with generally positive attitudes, we may affirm our approach.

**Comprehensibility:** Everyone finds captions easy to understand and give ratings of five points to approximately 90 percent of the charts. Therefore, the average rating of this criterion reaches surprising 4.8. A freshman of liberal arts indicated that it was acceptable to see such expressions from a machine, and she thought it was good enough.

**Correctness:** This criterion also gets full recognition and scores high. In most cases, participants find the description consisting of

194

the chart feature. An engineer from an IT enterprise said that the descriptions about trends were precise, and the judgments of outliers were consistent with his opinion. However, he supposed the caption would be more accurate if there were more details.

**Significance:** About 60 percent of captions are given strong applaud for their significance. A junior of data science stated that he found most captions discern the crucial features.

**Coverage:** Participants largely agree that the overall sentences well capture the major features of the charts. A graduate of computer science said that it seemed that the computer could read his mind, and what he thought of were all included in the captions.

**Helpfulness:** Attitudes towards this criterion of the participants vary. On the one hand, some agree that captions help them read the chart with guidance. While on the other hand, some doubt that the captions help little since they are easily readable facts which highly accord with human perceptions.

**Others:** Most participants are satisfied with the captions. In particular, one participant shows great interest in the inner mechanism and want to see the effect of even more complicated charts. However, there exist some complaints about the tedious stereotype. A senior of journalism said that some sentences picked up overmuch exact numbers, whereas human never spoke in that way, and that it would be better if the machine could make a summary. Also, a junior student in history major explained that he was tired of the monotonous sentences with words like "increase" and "unusual drops".

These opinions reveal the problem of lacking corpus in our current implementation, more or less. Notwithstanding, improvements could be made by enlarging the alternative word database and predefined rules for the feature-description mapping. For instance, adjectives like "greatly", "sharply", and "largely" could be used to describe the relatively significant change of trend. Moreover, switching sentences with the same sense would improve the diversity of the expression.

Also, we received suggestions to join closely related sentences together because separated sentences may bring bias to the analyst. Another participant also mentions that a well-organized paragraph, i.e., not individual sentences, would make the combination of the chart and caption easier for the audience to read.

## 6 CONCLUSION & FUTURE WORKS

In this work, deep neural networks are employed to construct a learning scheme to achieve the task of caption generation for charts, in which the usage of 1-D residual neural networks (Res-Nets) well shows the ability to extract the feature from the charts. Our model performs well in the feature extraction task, which presents the complicated relationship of data elements. The pre-trained model is deployed to a demo system, where the caption generation based on an uploaded vector-based chart is allowed. The presented results and positive response from the user study demonstrate the effectiveness of our proposed approach.

In the future, we would like to move forward to improve the language quality, the descriptive ability, and the chart type coverage of the captions generated through our approach. This includes a more intelligent caption pipeline, a more natural way to generate language, and broader coverage of the dataset together. We hope to seek research opportunities in the recent progress of natural language processing. Additionally, we regard it as an important direction to explore how the caption and the corresponding chart can be combined, supporting a better experience in exploratory data analysis.

## REFERENCES

[1] R. A. Al-Zaidy and C. L. Giles. Automatic extraction of data from bar charts. In *Proceedings of the ACM International Conference on Knowledge Capture*, 2015.

[2] C. Bryan, K.-L. Ma, and J. Woodring. Temporal summary images: An approach to narrative visualization via interactive annotation generation and placement. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):511–520, 2016.

[3] S. Carberry, S. Elzer, and S. Demir. Information graphics: an untapped resource for digital libraries. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 581–588, 2006.

[4] S. Demir, D. Oliver, E. Schwartz, S. Elzer, S. Carberry, K. F. Mccoy, and D. Chester. Interactive sight: textual access to simple bar charts. *New Review of Hypermedia and Multimedia*, 16(3):245–279, 2010.

[5] Ç. Demiralp, P. J. Haas, S. Parthasarathy, and T. Pedapati. Foresight: Recommending visual insights. Number 12, pages 1937–1940. VLDB Endowment, 2017.

[6] S. Elzer, N. Green, S. Carberry, and J. Hoffman. A model of perceptual task effort for bar charts and its role in recognizing intention. *User Modeling and User-Adapted Interaction*, 16(1):1–30, 2006.

[7] C. Goncu and K. Marriott. Tactile chart generation tool. In *Proceedings of the ACM International Conference on Computers and Accessibility*, pages 255–256, 2008.

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[9] J. Hullman, N. Diakopoulos, and E. Adar. Contextifier: automatic generation of annotated stock visualizations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2707–2716, 2013.

[10] V. O. Mittal, G. Carenini, J. D. Moore, and S. Roth. Describing complex charts in natural language: A caption generation system. *Computational Linguistics*, 24(3):431–467, 1998.

[11] J. Poco and J. Heer. Reverse-engineering visualizations: Recovering visual encodings from chart images. In *Computer Graphics Forum*, volume 36, pages 353–363. Wiley Online Library, 2017.

[12] E. Reiter and R. Dale. Building applied natural language generation systems. *Natural Language Engineering*, 3(1):57–87, 1997.

[13] M. Savva, N. Kong, A. Chhajta, L. Fei-Fei, M. Agrawala, and J. Heer. Revision: Automated classification, analysis and redesign of chart images. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, pages 393–402, 2011.

[14] A. Srinivasan, S. M. Drucker, A. Endert, and J. T. Stasko. Augmenting visualizations with interactive data facts to facilitate interpretation and communication. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):672–681, 2019.

[15] B. Tang, S. Han, M. L. Yiu, R. Ding, and D. Zhang. Extracting top-k insights from multi-dimensional data. In *Proceedings of the ACM International Conference on Management of Data*, pages 1509–1524, 2017.

[16] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164, 2015.

[17] A. Wilbik, J. M. Keller, and G. L. Alexander. Linguistic summarization of sensor data for eldercare. In *Proceedings of the IEEE Conference on International Conference on Systems, Man and Cybernetics*, pages 2595–2599, 2011.

[18] P. Wu, S. Carberry, S. Elzer, and D. Chester. Recognizing the intended message of line graphs. In *Diagrammatic Representation and Inference*, pages 220–234. Springer, 2010.

[19] 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 *Proceedings of International Conference on Machine Learning*, pages 2048–2057, 2015.

[20] L. A. Zadeh. A prototype-centered approach to adding deduction capability to search engines-the concept of protoform. In *Proceedings of the IEEE Intelligent Systems*, volume 1, pages 2–3, 2002.