



WYTIWYR: A User Intent-Aware Framework with Multi-modal Inputs for Visualization Retrieval

Shishi Xiao¹, Yihan Hou¹, Cheng Jin² , and Wei Zeng^{1,2†} ¹The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China²The Hong Kong University of Science and Technology, Hong Kong SAR, China

Abstract

Retrieving charts from a large corpus is a fundamental task that can benefit numerous applications such as visualization recommendations. The retrieved results are expected to conform to both explicit visual attributes (e.g., chart type, colormap) and implicit user intents (e.g., design style, context information) that vary upon application scenarios. However, existing example-based chart retrieval methods are built upon non-decoupled and low-level visual features that are hard to interpret, while definition-based ones are constrained to pre-defined attributes that are hard to extend. In this work, we propose a new framework, namely WYTIWYR (What-You-Think-Is-What-You-Retrieve), that integrates user intents into the chart retrieval process. The framework consists of two stages: first, the Annotation stage disentangles the visual attributes within the query chart; and second, the Retrieval stage embeds the user's intent with customized text prompt as well as bitmap query chart, to recall targeted retrieval result. We develop a prototype WYTIWYR system leveraging a contrastive language-image pre-training (CLIP) model to achieve zero-shot classification as well as multi-modal input encoding, and test the prototype on a large corpus with charts crawled from the Internet. Quantitative experiments, case studies, and qualitative interviews are conducted. The results demonstrate the usability and effectiveness of our proposed framework.

CCS Concepts

• **Human-centered computing** → Visualization; • **Information systems** → Query intent; • **Computing methodologies** → Artificial intelligence;

1. Introduction

Data visualization can empower users' understanding of data and facilitate communication. Therefore, enormous charts are flourishing on the Internet. Designing an appropriate chart is a time-consuming and labor-intensive process that needs to consider various visual attributes (e.g., chart type, color) [Shn96], as well as the design style (e.g., context information, 2D/3D effect) [MTW*12]. Therefore, designing based on existing examples, rather than starting with sketches, is a more preferred approach [BFW21, PSP21]. Chart retrieval, as an approach to return the ranked examples with respect to the corresponding query, has gained much interest in both industry and academia [LWW*22, HA19, SHL*16].

However, existing works mainly focus on improving the similarity between query charts and retrieval results while neglecting implicit user intent. As illustrated in Figure 1, existing chart retrieval frameworks might be divided into two categories: example-based (Figure 1 (left-top)) and definition-based (Figure 1 (left-bottom)) approaches. Example-based methods (e.g., [SSK06,

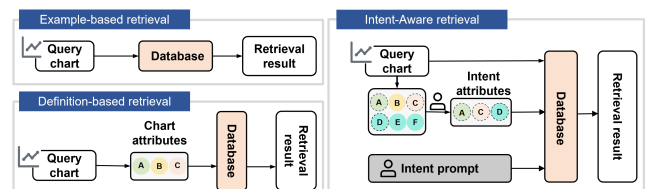


Figure 1: Comparison of pipelines of existing and our proposed chart retrieval frameworks: example-based retrieval (left-top), definition-based retrieval (left-bottom), and our proposed intent-aware retrieval (right).

DW14, BBK*18]) characterize the relevance criteria by the visual similarity of charts, whilst definition-based methods (e.g., [CCA15, HA19]) measure similarity based on attributes extracted from the input charts. However, these methods may suffer from both input and target shifts problems. First, many existing works take only specific formats (e.g., scalable vector graphics (SVG) [HA19, LWW*22]) of visualization as inputs, which are often synthesized with monotonous data distributions and limited chart types that incurs overfitting problem of models embedded in the framework. In contrast, most online charts are generally in bitmap formats,

† Wei Zeng is the corresponding author. Email: weizeng@ust.hk

and they are highly diverse in terms of chart types and visual styles [BDM*18]. In addition, online charts contain noisy and redundant information, putting higher demands on generalizability and robustness of the retrieval framework. Second, example-based methods would indiscriminately recall pixel-wise visual similarity in the input chart, while definition-based methods consider only a small number of pre-defined and fixed chart attributes. Such rigid similarity measurements would return unwanted target charts. Furthermore, it may tend to limited the retrieval to a specific chart attribute, a specific combination of chart attributes, or even chart attributes that are not necessarily enclosed by the query chart.

To mitigate such shifts, we propose a novel chart retrieval framework named WYTIWYR. It explicitly extracts visual attributes for the query chart, and offers a set of target charts through a customized retrieval procedure that considers flexible combinations of the visual attributes and users' intents. As Figure 1 (right) illustrates, our method tears down the query chart flexibly with customizable attributes and actively injects user intent as auxiliary inputs to seek better chart retrieval outputs.

Our intent-aware framework consists of two essential stages, the first stage is *Annotation*, which highlights explicitly disentangling the visual attributes for query charts. The disentangled attributes would enable the flexibly for users to combine the attributes representing their intent. Because these attributes differ amongst chart visualizations, we performed preliminary research with 18 visualization categories to establish the branching of each attribute and what users would anticipate when retrieving charts. Based on the study, we utilize deep neural networks to train several independent attribute classifiers tailored to four primary visual attributes, namely {*Type, Trend, Color, Layout*}. To enable the resilience to user intent, we leverage the state-of-the-art contrastive language-image pretraining (CLIP) [RKH*21] model famous for zero-shot classification. The model allows users for their own creation of the attribute classifier to identify the extended attributes. These attributes are excluded from our preliminary set of classifiers but are still within the query process.

The second stage is *Retrieval*, which highlights the role of human-in-the-loop through intent attributes as prompts. The intent attributes would condition intent-aware filter, narrow the search scope from a huge collection of charts, and finally generate multiple candidate charts. The multi-modal encoder fuses context information from both the query chart and user intent prompt into a joint representation. After that, similarity modeling is conducted between such joint representation and candidate charts. Ultimately, we rank the similarity scores in decreasing order and output the top-*K* results as the target charts. In summary, our main contributions are summarized as follows:

- **Intent-aware Retrieval.** We propose a novel framework integrating user intent into chart retrieval process. The *Annotation* stage disentangles attributes and enables a flexible combination of attributes. The *Retrieval* stage digests both query chart and user intent as multi-modal inputs to get target retrieval results.
- **Prototype Construction.** We implement a prototype system combining CLIP model with visual interface to support intent-aware retrieval. Dataset, code, pretrained model are released at <https://github.com/SerendipitysX/WYTIWYR>.

- **Extensive Evaluation.** We conduct extensive quantitative experiments, case studies as well as expert interviews to validate the effectiveness of our approach.

2. Related Works

Chart Attributes. Visualization is the process of mapping data to images. The data can be encoded in different ways, yielding various types of chart attributes that can be categorized in several levels. Low-level visual stimuli of charts include color, position, and shape [RTOT06], while high-level taxonomy includes consideration of the object of study, data, design model, and user model [TM04]. The importance of choosing appropriate chart attributes has been highlighted. Recent studies (e.g., [MMA18, KRS*21, LCF*15, SLC*23]) stress the effects of chart attributes on users' comprehension and cognitive loads. Specifically, Li et al. [LCF*15] point out that two charts of the same data but with different layouts can cause perceptual imbalance, even though they are informatively equivalent. Despite the importance, the design space of chart attributes is too large for designers to choose from, resulting in various types of charts in designers' own style [SLC*23]. Most of these works follow a coarse-to-fine strategy to first identify chart types and then extract visual marks and channels [SKC*11, JKS*17, PMH17, YZF*22].

Existing methods often mix visual attributes as global style for similarity estimation [MTW*18, SHL*16, JKS*17], in which not all attributes are of interest to users [LWW*22]. To fill the gap, the *Annotation* stage of the proposed WYTIWYR framework allows users to disentangle attributes in a given chart.

Chart Retrieval. The core for effective retrieval is to delineate the similarity between the query input and retrieval candidates in the database. Based on the object of similarity measurement in the retrieval process, existing chart retrieval strategies can be mainly categorized into two classes: definition- and example-based approaches. Definition-based approaches (e.g., [CCA15, SHL*16, HA19, ZDCC21, ZFF22, LWW*22]) characterize the criteria of similarity by explicit chart attributes, making it preferable for well-configured chart formats, including SVG-style and Vega-Lite grammars. For instance, Hoque and Agrawala [HA19] developed a search engine for D3 visualizations by using a JSON-like configuration that dictates query visual encoding. Moreover, some studies (e.g., [HMSA08, SGP*18]) further use user interaction records and visualization states represented in a provenance graph to match the previous exploration states. Recent example-based methods regard visualization charts as images and leverage deep learning models [MTW*18, ZFC*23, SHL*16, JKS*17, YHZ23] to automatically extract visual features via an end-to-end manner. A set of implicit low-level features are leveraged to estimate the global similarity. However, definition-based approaches only consider partially pre-determined visual attributes, while example-based approaches tend to match the query with all indiscriminate attributes.

Our proposed WYTIWYR framework combines the advantages of example-based approaches in terms of their capability of capturing implicit attributes and also definition-based approaches that leverage explicit attributes with high interpretability. Moreover, the *Retrieval* stage enables a flexible combination of these attributes with respect to user intent.

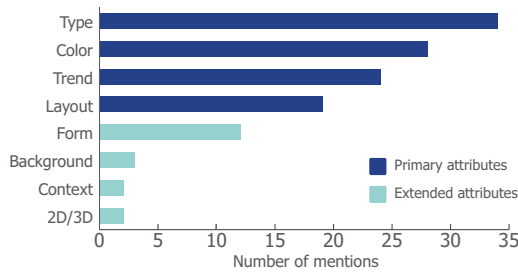


Figure 2: Results of our preliminary study. Type, color, trend, and layout are voted by most participants and identified as primary attributes, whilst the others are categorized as extended attributes.

Vision-Language Pretraining. Large-scale pretrained models have shown promising performance in both natural language processing and vision tasks these years, such as BERT [DCLT18], RoBERTa [LOG*19], and GPT series [BMR*20, RNS*18, RWC*19]. Following the paradigm of pretrain and finetune, many downstream tasks [GKSS*19, XBK*15, LGR*20] transfer the knowledge from the pretrained model without training a new model from scratch by utilizing it. Prompt, a text sequence in the form of natural language, links the pretrained model and downstream tasks as a bridge. Strobelt et al. [SWS*22] develops a visual interface to provide a promising prompt evaluation. Combined with a high-capacity text encoder and visual encoder, contrastive language-image pretraining (CLIP) [RKH*21] learns heterogeneous multi-modality representations from 400 million image-text pairs by resorting to semantic supervision from the embedding space of CLIP. Several studies [PWS*21, GPM*21, GLKC22] extend its applicability in a zero-shot manner, which means the model can predict samples whose class is not observed during training.

In this work, we take advantage of vision-language pretraining models, to encode user intents as a prompt to collaborate with the decisive process of retrieval. The CLIP-driven model is leveraged to align user intent with corresponding visual attributes in both *Annotation* and *Retrieval* stages. In this way, we can offer highly customizable attributes for chart annotation and multi-modal representation extraction for chart retrieval.

3. WYTIWYR Framework

To build the WYTIWYR framework, we first formulate chart attributes based on a preliminary study (Sec. 3.1), then present an overview of the WYTIWYR framework (Sec. 3.2).

3.1. Attribute Formulation

There are two categories of chart attributes for chart retrieval: 1) *primary attributes* that are generally considered for chart retrieval, and 2) *extended attributes* that meet the specific needs of different users. To better understand user intents, we conducted a comprehensive study to categorize chart attributes.

Study design. Our preliminary study was conducted online and involved 40 participants whose ages ranged from 20 to 52 (mean = 24.3). Before the study, we first introduced the background and purpose of the experiment to the participants. Then we tested their knowledge on the chart through a set of verification questions. Only

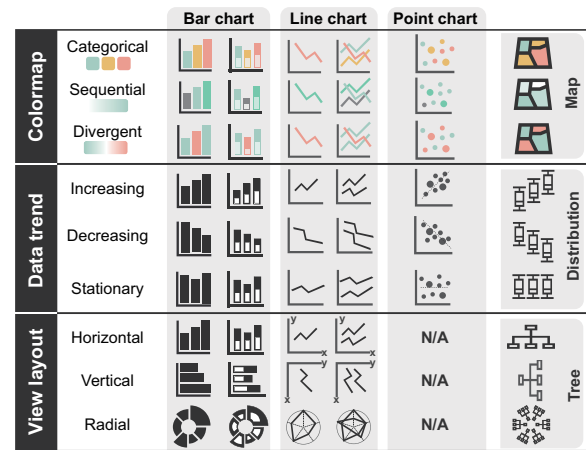


Figure 3: Illustration of attribute options of primary attributes for several chart types. Some options may not be available for certain chart types, such as Layout for the point chart.

the participants who could identify typical charts, such as bar and pie charts, were allowed to participate in the study. Fortunately, all participants were familiar with charts since their popularity on social media. Next, the participants were shown multiple real-world charts and asked to indicate what attributes were of interest. To eliminate any potential subjective bias during the chart selection process for testing, a random sample of five images (three in the Beagle dataset [BDM*18], two in other real-world examples) was taken for 18 type of chart from the collected data. The testing charts were carefully chosen according to chart type taxonomy as in the study by Borkin et al. [BVB*13]. The detail of sampled testing chart can be seen at supplementary material. Their answers were recorded in text and summarized in different perspectives after the experiments. The perspectives were cross-validated by two of our co-authors to ensure correctness.

Result. As shown in Figure 2, we identify a total of eight perspectives of attributes are of interest from the feedback, which are also consistent with previous studies [BVB*13, LWW*22]. Among them, *Type*, *Color*, *Trend*, and *Layout* are the four attributes mostly frequently mentioned by the participants and can be applied to most types of charts. Although the *Form* attribute was also frequently mentioned by participants, it is primarily associated with line charts and less frequently mentioned in the other chart types. Therefore, to ensure generalization and expressivity, we decided not to combine *Form* with the other four attributes. We dub *primary attributes* as $\mathcal{T}_p := \{Type, Color, Trend, Layout\}$. In addition, the participants also brought up some other chart attributes, including *form*, *background*, *context*, and *2D/3D*, which we dubbed as *extended attributes* \mathcal{T}_e that includes the other attributes. For these chart attributes, the variation between different users is large, making it difficult to be comprehensively enumerated.

Moreover, the study also revealed a set of design options for each chart attribute. Specifically, we choose 10 primary categories and 18 subcategories of chart types from the visualization taxonomy [BVB*13]. Figure 3 illustrates the options in the attribute type of $\{color, trend, layout\}$ for several chart types including bar, line, and point charts. Notice that some chart types may not have all

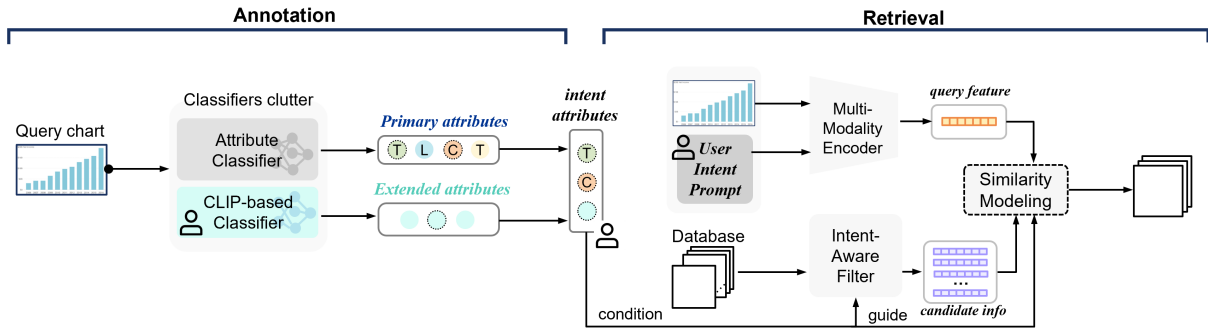


Figure 4: Overall pipeline of attributes-aware retrieval. The pipeline consists of two stages: Annotation for extracting the attributes and Retrieval for modeling the similarity between query charts and charts in the database.

options mentioned above. For example, *layout* for point charts are not available, and maps only have *color* options.

3.2. Overall Pipeline

The workflow of our proposed WYTIWYR framework is depicted in Figure 4. For the *Annotation* stage, we employ several classifiers based on robust neural network architecture to disentangle $\{\mathcal{T}_p, \mathcal{T}_e\} \in \mathcal{T}$ embedded in the query chart \mathcal{Q} . After that, users can select the disentangled attributes \mathcal{I}_A as their intents based on their needs. Also, \mathcal{I}_A would condition the components of the filter in the retrieval stage. For the *Retrieval* stage, we take a dual-path similarity modeling between the *query feature* and the *candidate info* with the guidance of intent attributes to yield the return charts. For the path of the *query feature*, bitmap query chart \mathcal{Q} with user intent text prompt \mathcal{I}_P is fed into a multi-modal encoder to obtain the joint representation. For the *candidate info* path, the feature is produced by charts in the database filtered by the intent-aware filter built by \mathcal{I}_A . Throughout the overall pipeline, user intent can be injected and guide the chart retrieval process by the following three ways:

- **Classifier Customization.** Users can tailor the usage of classifiers depending on their needs in order to determine the presence of an attribute in the query chart.
- **Disentangled Attributes Selection.** The attributes adopted in the *Retrieval* stage can be selected and combined by the users.
- **User Prompt Tuning.** Users can add specific implicit intent as the text prompt to guide the retrieval process, which is independent of the query chart.

4. WYTIWYR Prototype System

4.1. Stage1: Annotation

As shown in Figure 5, the task of the *Annotation* stage is to disentangle and generate the visual attributes \mathcal{T} adhered to the given query chart \mathcal{Q} . For the primary attributes \mathcal{T}_p , we adopt four classifiers for separate extraction in a supervised learning manner. For the extended attributes \mathcal{T}_e , as they are absent from our training, we conduct the extraction process in the fashion of zero-shot learning.

4.1.1. Annotation for Primary Attributes

Redundant Information Removal. Several practical scenario-based raw data are shown in Figure 6 (a), including fancy deco-

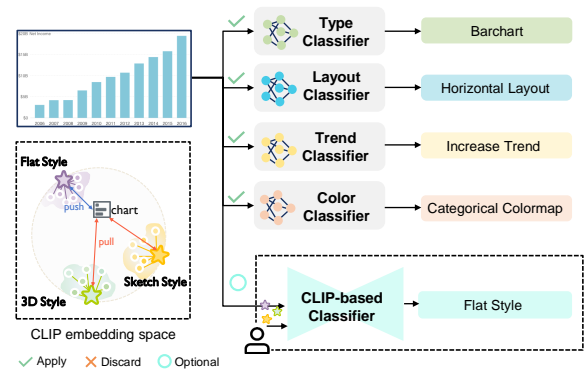


Figure 5: The Annotation stage of our pipeline. To disentangle various attributes in the query chart, four classifiers are built for primary attributes. In addition, one CLIP-based classifier is employed to optionally annotate the extended attribute via user customization.

ration, text information, background, and legends. This work focuses on the intrinsic attributes of \mathcal{Q} instead of this redundant information. Hence we remove the redundant information using ISNet [QDH*22], a segmentation network with pretrained weights.

Nevertheless, there exists context beyond the segmentation capability of ISNet, such as the emojis in the left-most example in Figure 6 (a) & (b). The redundant emojis in red and yellow colors would degrade the performance of colormap classification. Instead of being categorized as *categorical colormap*, the chart would be mistakenly recognized to *diverging colormap*. To mitigate such issue, we ignore the colors that share less than 10% pixels of non-transparent regions of the image after segmentation.

Unified Classifiers. For annotation of the primary attributes, we employ ResNet50 [HZRS16] architecture as the backbone for all four attribute classifiers. Thanks to the residual connection among the convolution layers, the network is both lightweight and performance-guaranteed. During the training process, the network will learn the complex features under the supervision of given attributes. In the reference process, the trained classifier would identify the corresponding attribute in the query chart, as in Figure 5.

The only difference in the four classifiers is the number of output channels in the last fully connection layer. For instance, the number of *Type* classifiers is set to 18 since there are 18 types of charts

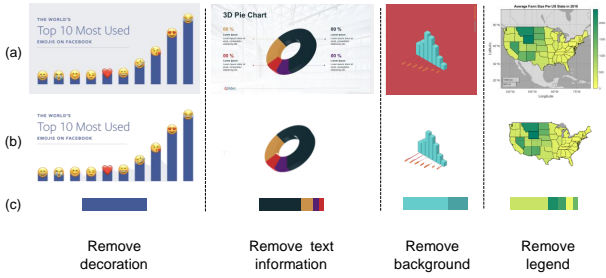


Figure 6: Redundant information removal and main color extraction. (a) Raw images. (b) The chart after removing redundant information. (c) The extracted color after filtering redundant colors.

under consideration. Similarly, based on the specific task, we set this number to 3 for the *Layout*, *Trend*, and *Color* classifiers with the respect to their attribute options; see Figure 3 for detail. Note that Q may not have the coverage of all attributes included in the classifiers. For instance, \mathcal{T}_p of a heatmap chart is $\{Type, Color\}$ while $\{Trend, Layout\}$ attributes are absent.

Loss Function. There are two challenges to overcome when formulating the loss function: 1) imbalanced samples in each class of the dataset (see Table 1), and 2) the existence of noisy samples that are hard to classify. In order to improve the accuracy for \mathcal{T}_p annotation, we introduce Focal Loss [LGG*17], which modifies the standard cross entropy loss to overcome the above-mentioned challenges. The Focal Loss is defined as:

$$\mathcal{L}_{FL}(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t), \quad (1)$$

where $p_t \in [0, 1]$ represents the estimated probability of class t , α_t represents the scaling factor, and γ represents the modulating factor. Among them, α_t is set by inverse class frequency, thus learning parameters tend to contribute to classes with fewer samples, and γ assists in up-weight the loss assigned to poor-classified examples, avoiding the possibility that the training process is dominated by the amount of well-classified samples.

4.1.2. Annotation for Extended Attributes

Since the primary attributes only consider the general needs in chart retrieval, we offer an optional classifier manipulated by the users to distinguish \mathcal{T}_e in this query chart, as shown in Figure 5. The input of \mathcal{T}_e classifier requires the user to provide several labels of the attribute apart from Q (e.g., style). We denote these labels as $\{t_1, t_2, \dots, t_m\} \in T$. As these labels are out of our dataset, a task of zero-shot classification is naturally formed. In the following, we will briefly introduce the mechanism of the CLIP model at a general level with a toy example as in Figure 5, where the user sets the text labels as [*3D style*, *Flat style*, *Sketch style*].

The CLIP model aligns T embeddings with Q embeddings in a multi-modality embedding space in a contrastive manner. Specifically, in the embedding space of the example in Figure 5, the distance of Q and *3D style* would be less than the other two irrelevant labels. As such, the query results tend to be *3D style* bar charts. The CLIP model balances well between the performance and computational cost in the prototype system. Nevertheless, the

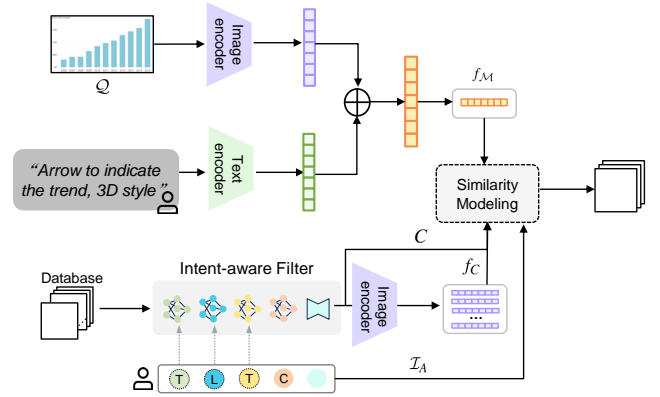


Figure 7: The Retrieval stage consists of two branches. The intent-aware filter branch (bottom) filters out charts based on intent attributes selected by the user, generating multiple chart candidates. The multi-modal encoder branch (top) produces the query feature based on the multi-modal inputs. Similarity modeling is conducted between results of these two branches.

model in our framework can be feasibly replaced with other advanced language-image models, e.g., [LLXH22, LSG*21, ZZF*22].

4.2. Stage2: Retrieval

As the *Annotation* stage completes the disentanglement of the attributes \mathcal{T} in the query chart, users in the *Retrieval* stage can select the intent attributes \mathcal{I}_A based on their intents. Depicted in Figure 7, there are two branches with different modules for our *Retrieval* stage, namely multi-modal encoder and intent-aware filter. The intent-aware filter rules out charts whose attributes are beyond the range of \mathcal{I}_A . The multi-modal encoder integrates Q and \mathcal{I}_p as a joint representation. Then, similarity modeling is conducted between results of these two branches, yielding the final retrieval results based on the modeling score.

4.2.1. Multi-Modal Encoder

This branch would form a CLIP-generated *query feature* with inputs Q , \mathcal{I}_p . As introduced in Sec. 4.1.2, the CLIP embedding space has heterogeneous aligned text and visual features from the multi-modal input. Hence, Q and \mathcal{I}_p are encoded into features by their respective CLIP encoders, and fused as a joint multi-modal feature f_M that can be denoted as:

$$f_M = \frac{F_\theta(Q) + G_\theta(\mathcal{I}_p)}{2}, \quad (2)$$

where F_θ and G_θ are denoted as the image encoder and text encoder of the CLIP model, respectively.

4.2.2. Intent-Aware Filter

The branch of Intent-Aware Filter rules out irrelevant charts in the database and forms the *candidate feature*. Previous works (e.g., [HA19, MTW*18]) on chart filtering primarily rely on fixed settings of the model, which are coarse and neglect the user's intents. For this shortcoming, we propose an intent-aware filter constructed by \mathcal{I}_A , a set of disentangled attributes selected by the user.

For example, with Q illustrated in Figure 7, a user may select the $[Type, Layout, Trend]$ as the target attributes for the retrieval, then the filter retains n charts containing these target attributes in the chart database. Then, the retained charts would serve as candidates $C := \{c_1, c_2, \dots, c_n\}$ that are encoded as *candidate feature* $f_C := \{f_{c_1}, f_{c_2}, \dots, f_{c_n}\}$ by image encoder in the CLIP model.

4.2.3. Similarity Modeling

As the core of retrieval, similarity modeling is comprehensively performed with triplets $\{Q, \mathcal{I}_A, f_M\}$. Among them, Q provides the global perception composed of implicit features. The explicit intent attributes \mathcal{I}_A and implicit query feature f_M that include user intent prompt and query chart, work together to recall user-intended examples throughout the retrieval process. The overall similarity score S can be denoted as:

$$S = S_Q \cdot e^{\nu S_{\mathcal{I}_A} + \mu S_M}, \quad (3)$$

where S_Q demotes the global perception at pixel level, and $S_{\mathcal{I}_A}$ and S_M denotes the similarity score of user-selected attributes and the multi-modality feature, respectively. The scaling factors ν and μ are empirically set as 1 and 5, respectively. All S_Q , $S_{\mathcal{I}_A}$ and S_M are normalized with range of $[0, 1]$. In the following, we will introduce these similarity scores in detail.

Global Perception Score. Although salient attributes with respect to user intent are disentangled and selected, a multitude of implicit features are intertwined within the chart image, which forms the concept of global perception. We adopt the classic cosine similarity and compute it at the pixel level between Q and the feature of every candidate $f_{C_i}, i = 1, 2, \dots, n$ as follows:

$$S_Q = \frac{F_\theta(Q) \cdot f_{C_i}}{\|F_\theta(Q)\| \|f_{C_i}\|}. \quad (4)$$

Intent Attributes Score. Extended attribute \mathcal{T}_e contains a user's expected attributes of retrieval results charts apart from four primary attributes. Among the attributes, *Type* and *Layout* define core properties of a chart, which is easy to distinguish from Q . Therefore, we neglect them in the similarity crafting. The other attributes serve as the miscellaneous variant, and the change of them can be minuscule. This motivates us to build another score $S_{\mathcal{I}_A}$ to further enhance the retrieval process. Therefore, we mainly consider the following three attributes: *Trend* \mathcal{N} , *Color* \mathcal{C} and *extended attribute* \mathcal{T}_e . $S_{\mathcal{I}_A}$ is formulated as:

$$S_{\mathcal{I}_A} = S_{\mathcal{N}} + S_{\mathcal{C}} + S_{\mathcal{T}_e}. \quad (5)$$

For $S_{\mathcal{N}}$, we build an extractor to extract the trend feature from the query chart Q and the every candidate chart $C_i, i = 1, 2, \dots, n$. The extractor shares the parameters with a trained trend classifier in the *Annotation* stage except the last fully connected layer. Then, $S_{\mathcal{N}}$ can be estimated by cosine similarity between the extracted features.

For $S_{\mathcal{C}}$, we follow the scheme in Figure 6 to transform Q into a proportional color palette after the steps of background removal and color extraction. Then the proportional color palette is transformed into a 128-bin color histogram in all RGB channels and separately stored in three vectors. We then concatenate these vectors to form

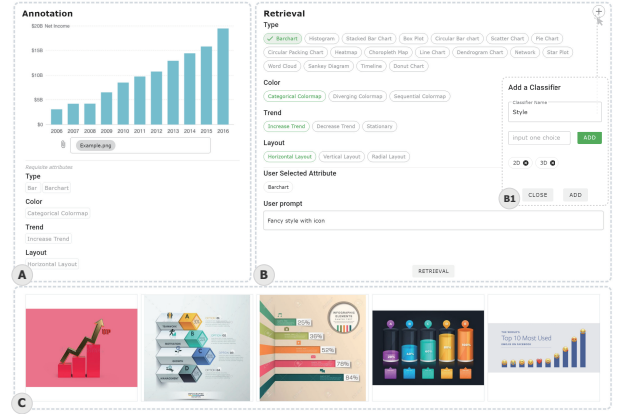


Figure 8: Interface of our prototype system. A) the Annotation view decouples the properties of the query chart; B) the Retrieval view allows users to select the primary and extended attributes, and C) the Result view shows the top 5 retrieved charts.

the ultimate color vector V . Denote $V_Q, V_{C_i}, i = 1, 2, \dots, n$ as the color vectors of $Q, V_{C_i}, i = 1, 2, \dots, n$, respectively, we then estimate the cosine similarity between them.

For $S_{\mathcal{T}_e}$, since it is hard to quantify the similarity of intents, we leverage the power of the CLIP model again to give an accurate response. We feed the CLIP model with candidate chart $C_i, i = 1, 2, \dots, n$ and text labels in user intent $t_i, i = 1, 2, \dots, m$. For each chart, we would obtain m outputs $\{y_1, y_2, \dots, y_m\}$. Prior to the computation, the user would select one text label indexed s as the attribute that best represented their intent. Then, we denote $S_{\mathcal{T}_e}$ for a candidate chart as:

$$S_{\mathcal{T}_e} = \frac{e^{y_s}}{\sum_m e^{y_m}}. \quad (6)$$

Feature Matching Score. Similarly, we match the closeness between the multi-modal feature and the candidate feature by the following equation:

$$S_M = \frac{f_M \cdot f_{C_i}}{\|f_M\| \|f_{C_i}\|}. \quad (7)$$

4.3. System Interface

We design an interface for the prototype system that allows users to retrieve a chart according to their intents.

Annotation View. In this view, users can upload a query chart and get disentangled primary attributes, including *Type*, *Color*, *Trend*, and *Layout*. As shown in Figure 8 (A), a bar chart is uploaded and the automatically annotated attributes, “*Bar chart, Categorical Colormap, Increasing Trend, Horizontal Layout*” are presented.

Retrieval View. In this view (Figure 8 (B)), users can view and choose both primary and extended attributes. The first tag in green corresponds to the annotated attributes from the query chart, while other options are also shown, allowing users to select the desired attributes. If users feel that the default primary attributes are insufficient or have other attributes of interest, they can add a new classifier by clicking on the chart in the upper right corner, which

reflects the philosophy of the proposed intent-aware design. In the input box below, users can enter their intentions as intent prompts for more customized queries. As shown in Figure 8 (B1), the user selects “Bar chart” for the type attribute and enters “Fancy style with icon” in the Retrieval view.

Result View. This view (Figure 8 (C)) shows top five query results, arranged in decreasing order from left to right according to their similarity scores.

5. Evaluation

5.1. Quantitative Experiment

Table 1: Our integrated dataset consists of 18 types of charts from Beagle [BDM*18] and manual collection.

Chart Type	#Count	Chart Type	#Count
Bar chart	7269	Heatmap	352
Stacked Bar Chart	1159	Line Graph	11605
Circular Bar Chart	608	Star Plot	491
Donut Chart	2459	Choropleth Map	640
Pie Chart	2587	Scatter Plot	3000
Sankey Diagram	266	Word Cloud	406
Timeline	324	Dendrogram	298
Box Plot	571	Network	395
Histogram	695	Circular packing chart	139

Dataset. Previous works concentrate on retrieval of limited chart types, e.g., single [MTW*18] or a few type [CCA15, HBL*19] chart retrieval. Moreover, some of the literature coarsely considers the categorization of charts, by broadly grouping several chart categories into one [LWW*22]. Many datasets are synthesized with simple composition and monotonous variation, making it difficult to adapt to real-world scenarios that are much more complex. In this work, we utilize the Beagle dataset [BDM*18] with charts in bitmap format, which offers visualization collections designed by real-world users through multiple tools, including D3, Plotly, Chartblocks, Fusion Charts, and Graphiq. To make our framework more robust, we further add more difficult instances by manually collecting 4k images from Pinterest, which supplements a large number of stylized and irregular charts. We filter out charts whose type is beyond our scope (e.g., scientific visualizations) and manually label them with \mathcal{T}_p . Finally, our dataset consists of 33,260 images in total. The detailed distribution is shown in Table 1.

Annotation Accuracy. We evaluate the performance of \mathcal{T}_p annotation using ResNet50 architectures with different losses. As Table 2 lists, the best result is achieved with the designated Focal Loss.

Retrieval Results. To evaluate the retrieval performance comprehensively, we examine both precisions and recall via the F1-score. Denoting the classification result with true positive, true negative, false positive, and false negative conditions as TP, TN, FP, FN , F1-score can be formulated as:

$$F1\text{-score} = \frac{2TP}{2TP + FP + FN}. \quad (8)$$

The retrieval performance with \mathcal{T}_p is individually computed using F1-score in the top- K fashion, where $K \in \{3, 5, 10\}$. We devise

Table 2: Accuracy of Annotation results.

Method	Type	Trend	Layout	Color
ResNet50+MSE Loss	0.9518	0.8290	0.9537	0.7963
ResNet50+Focal Loss	0.9601	0.8424	0.9653	0.8019

Table 3: F1-scores of Retrieval results.

Top-K	Method	F1-Score			
		Type	Trend	Layout	Color
3	HOG	0.6199	0.6140	0.5449	0.6800
	CNN	0.9154	0.8043	0.7095	0.7717
	Ours	0.9549	0.8360	0.9280	0.8260
5	HOG	0.5067	0.4853	0.3857	0.6045
	CNN	0.8872	0.7550	0.6324	0.7076
	Ours	0.9364	0.7944	0.9118	0.7730
10	HOG	0.4124	0.4261	0.2741	0.5388
	CNN	0.8607	0.7085	0.5527	0.6520
	Ours	0.9091	0.7546	0.8812	0.7223

two settings of with and without user intent to fully evaluate our approach. For the setting of the absence of user intent, similarity estimation depends on the global perception S_{Global} .

To demonstrate the effectiveness of our approach, we take two other approaches as the counterpart: a conventional method (Histogram of Oriented Gradients [DT05], denoted as HOG from hereon), and a deep-learning-based method (ResNet50, denoted as CNN from hereon). The quantitative results are displayed in Table 3. The results reveal that our method significantly outperforms the others. Furthermore, as K increases, our method keeps the performance superiority while other methods degrade dramatically. For the evaluation of user intent retrieval, as there are no ground-truth labels to compute quantitative results, we visually compare results generated by different methods for several typical retrieval results. As shown in Figure 9, CNN outputs visually more similar results than HOG. Our method also produces similar results without user intent. Nevertheless, our method allows users to add their intent by selecting the disentangled primary attributes \mathcal{T}_p or adding extra intent, which contributes to retrieving results of interest as shown in the last two rows.

5.2. Case Study

During the process of visualization design, many inspirations may arise, but bringing those inspirations to fruition can be time-consuming. Retrieval is a fast way to validate those inspirations. Inspired by [BLBL22], two usage scenarios of chart retrieval are illustrated as a proof-of-concept of the WYTIWYR framework: 1) extending the design space by explicit goals and 2) fuzzy retrieval based on implicit user intent. For each scenario, we compare the results generated by our intent-aware chart retrieval technique with those generated by intent-free chart retrieval technique.

5.2.1. Design Space Extension by Explicit Goals

With the prototype system, users can customize retrieval inputs based on disentangled attributes of the query chart and the user



Figure 9: Visual comparison of chart retrieval by different methods. First row: Retrieval results by the conventional HOG method. Second row: Retrieval results by the deep-learning-based CNN method. Third to fourth rows: Retrieval results of our WYTIWYR framework.

intent prompt. Moreover, beyond the four primary attribute classifiers, users can further add customized classifiers to extract extended attributes. In this scenario, we divide the disentangled attribute operation into three categories: origin attribute change, new attribute addition, and existing attribute deletion. In the end, we show an interesting case of attribute transfer that combines these operations. Figure 10 shows four cases of our study. In general, retrieval without user intent provides similar results as the input but gives minimum design insights. Instead, with the guidance of user intent, our framework retrieves more diverse results that are well-accepted. The following lists more details for each case.

Original Attribute Change. The disentangled attributes are independent of each other, thus, users can replace one of the attributes while keeping others unchangeable. In Figure 10 (a), the color of the choropleth map is changed to user prompt “pink”.

New Attribute Addition. Users can add new attributes, together with the disentangled attributes, to better describe their specific needs. In Figure 10 (b), users add “with a curve to indicate the distribution”, and the results are changed to combinations of histograms and line chart.

Existing Attribute Deletion. As the attributes can be disentangled as users’ needs, they can remove some attributes of no interest. In Figure 10 (c), the user dislikes dark background, then can discard the attribute by adding a new classifier with labels of [“dark background”, “white background”].

Attribute Transfer. As in Figure 10 (d), given a circular packing chart as a query chart, the user gets the attribute annotation as {Type: “Circular Packing Chart”, Color: “Sequential Colormap”}. The user would like to get heatmaps with the same form as the query chart but with more contrast colors. S/he can get the desired results by changing Type and Color attributes to {Type: “Heatmap”, Color: “Diverging Colormap”} in the retrieval stage.

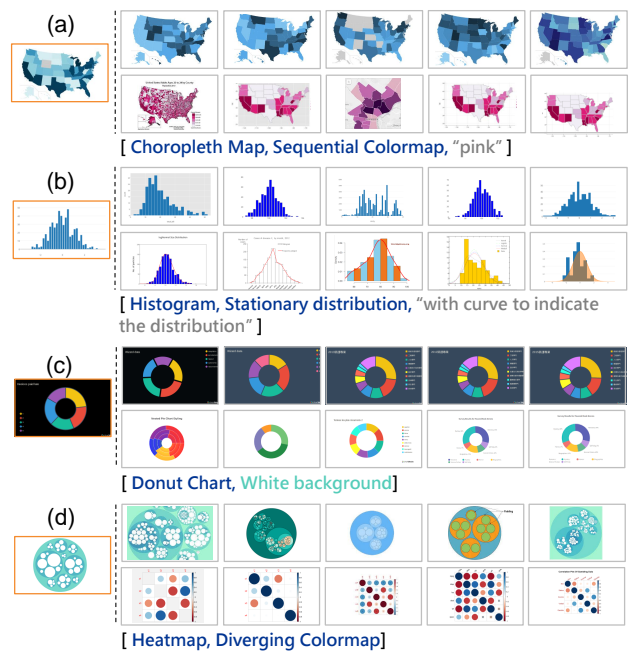


Figure 10: Four cases for extending the design space by explicit attributes. In each case, the first line presents results without user intent, while the second line presents results with user intent.

5.2.2. Fuzzy Retrieval by User Intent

Designing an expressive visualization incurs a steep learning rate for some novices, who may not have a specific design prior and prefer to perform trial and error on the search engine to seek a desirable design. Below, we list three scenarios to illustrate how our method supports such retrieval. Similarly, we will compare the results of the intent-free and intent-aware retrieval.

Text Information Seeking. Keyword-based searching is popular,

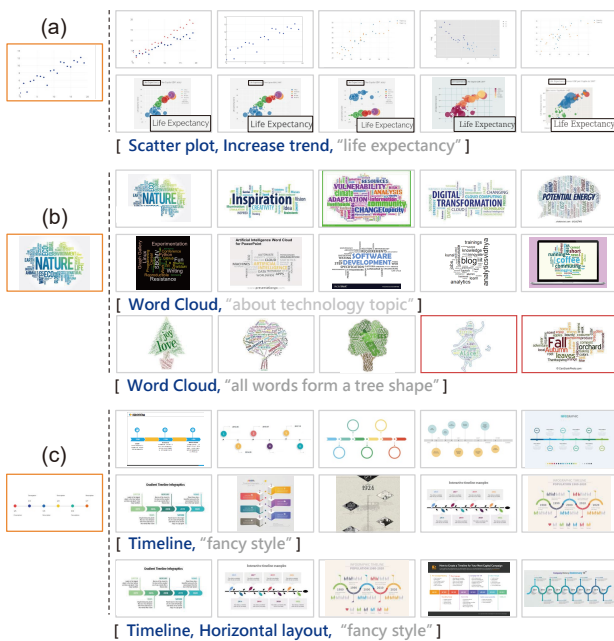


Figure 11: Three cases showing the comparison for intent-free and intent-aware retrieval in the fuzzy retrieving scenario. Negative examples are highlighted with a red border.

but existing works approach this goal by storing text information in advance [HA19, CCA15] or relying on OCR [SKC* 11], hindering generalizability and robustness of the works. Instead, our method allows to recognize text information based on knowledge transferred by the pretrained CLIP model. As shown in Figure 11 (a), to generate a chart for visualizing “life expectancy”, the user keeps the origin chart type *Scatter plot* and data trend *Increase trend*, and adds the user intent as a text prompt “life expectancy”. The intent-aware retrieval results imitate the famous design in Gampinder.

Relevant Topic Finding. Our method also has the capacity to find chart examples related to a specific topic. Previous work used word2vec [Chu17] to model the distances between words [HA19], limiting the search scope within texts. We enhance the generalizability of matching text prompt with both text information and visual attributes. In Figure 11 (b), given a word cloud with the theme of nature, the user can retrieve some examples with text prompt as “about technology topic”. Our system returns examples with content about technology. Moreover, with the text prompt of “all words form a shape of a tree”, our system aligns the semantic information of text with visual attributes, returning examples with tree shapes instead of only containing the word “tree”.

Abstract Description Searching. In the process of retrieval, users not having a clear search target tend to use ambiguous and abstract descriptions. In Figure 11 (c), the user inputs a basic timeline diagram with “fancy style”, and gets several well-designed examples. To constrain the scope of *vertical layout*, the user adds such attribute and performs targeted retrieval.

5.3. Qualitative Evaluation

To verify the usability of our proposed framework, we conducted semi-structured interviews on our prototype system.

5.3.1. Study Design

Our study recruited seven participants (three women, four men) with ages 22 to 26. To demonstrate that our framework is useful for different levels of users, we enroll four experts, *i.e.*, visualization designers (E1 - E4), and three novices with only experience in using visualizations in reports or courses (N1 - N3).

Before the experiment, we collected basic information from the participants through a questionnaire and introduced them to the use scenarios of chart retrieval. Then we instructed them to familiarize themselves with the system workflow. We also showed some examples of using prompts to help participants understand the prompt better and make a better choice of prompts. After understanding the background, participants were allowed to freely explore and use our system. They could upload a query chart and try to get a satisfactory design in multiple iterations. If there was no satisfactory design, they could also try combinations of attributes and prompts to see if the retrieved chart has inspiring results. Throughout the process, participants were encouraged to think aloud and give feedback whenever they wanted. After they felt it was sufficient, we conducted interviews about the usability of our system. Three questions were included: 1) Whether our approach is *effective* in helping them find the chart that meets their intents; 2) Whether our approach is *efficient* enough to avoid long waits; 3) Whether they are *satisfied* with our system. The experiment lasted about 45 minutes. The participants were not compensated with money but beverages worth about \$5 dollars after the study.

5.3.2. Feedback

Effectiveness. In general, all users agreed that our system effectively supported for retrieving the desired charts based on their intents. Most participants (6 out of 7) thought that our setting of the primary attribute was appropriate and comprehensive. E2 suggested to extract the main hue used in the chart in the annotation stage. Regarding the comprehensiveness and accuracy of the prompts, all participants felt that their search needs were largely met and the results were generally consistent with their intents. N1 spent a long time exploring the map as shown in Figure 10 (a). He found it exciting when typing “India”, an India map was retrieved. Our system also effectively recognized his intents when he searched for maps of various colors.

The participants pointed out some limitations as well. When the prompt given by the user had multiple meanings, sometimes only one or two of the search results were exactly what the user expects. E4 tried to add a trend arrow on the bar chart. However, when he only typed “with arrow”, some bar charts using arrow icons were also retrieved. He said that sometimes it was required to go and try several prompts to get the desired result. E3 encountered a similar situation, and it might be due to our dataset only containing limited charts that matched her specifications. Besides, sometimes participants may want to adjust attributes in the query chart, such as changing the white background to dark (N2), and changing the

colorful design to black and white (E2). However, one or two of the returned results still had the features that they would like to change. “There are times when I want the query to focus more on the prompt I give and less on the original chart,” said N2.

Efficiency and Satisfaction. All participants agreed that our system was responsive and provided a good user experience. They were satisfied with our retrieving framework, and they pointed out some possible improvements. They found it very helpful to decouple the chart into attributes and have a selective query. N2 stated, “It saved me the time of checking a lot of examples and found the desired design directly”. N3 appreciated the function of adding a new classifier since it can be used to explore new types of visualizations. “It is convenient for novices who do not understand the type and content of the chart at first,” he gave the reason. E1, a participant with a design background, said our results were inspiring. She was willing to see more results, even if they did not exactly match the search intents. “This could inspire me to come up with new ideas,” she added.

6. Discussion

To adapt to more complicated real-world data and diverse types of charts, we propose WYTIWYR—an intent-aware framework that offers users a flexible approach to chart retrieval by using both query chart and user intent. Thereby, WYTIWYR can address various issues related to traditional chart retrieval and expedite the design process. In conventional retrieval, not all chart attributes may be relevant to users’ intents, whilst some preferred attributes may even be absent from the chart. Instead, our system disentangles and combines attributes to ensure that the retrieved results encompass all attributes of user’s intents. There are also several limitations in our current methods and avenues for future work.

6.1. Limitation

User Intent or Query Chart. The input of our method consists of a query chart and user intent. The composition of two factors affects the retrieval results. As shown in Figure 10 (b), users have a strong intent of charts that have a curve, and our system would tend to return the results meeting such intents. However, the perception of global distribution may be lost. The priority between user intents as text and global distribution in the input chart is hard to tackle due to the usage scenarios. Furthermore, within the user intent, the weights of selected attributes and text prompts are hard to set in the similarity modeling. A dynamic adjustment of the priority between the user intent and query chart is demanded in this scenario.

Prompt Sensibility. Text prompt is integrated with query chart as joint input for retrieval in our work. However, designing effective prompts is challenging due to the significant impact of even slight wording changes on performance [ZYL22]. In our framework, prompt design is limited by the CLIP pretrained model, which is trained mainly on natural images, not visualization charts. This makes it difficult to accurately interpret expert expressions, such as using “pillow” instead of “bar” to depict a bar chart. The CLIP model also struggles to identify rare or novel objects in both natural images and visualization charts. Fine-tuning the CLIP model or using advanced language-image models (*e.g.*,

[LLXH22, LSG*21, ZZ*22]) are promising. Obtaining enough chart-description pairs for training is also necessary, even the process is time-consuming and resource-intensive.

Dataset. Despite collecting a large number of real-world charts, our database remains limited in meeting vast user needs. In Figure 11 (b), only the first three charts are tree-shaped since we only have these charts that meet the user’s intent in our database. The unbalanced dataset may adversely affect retrieval performance, as the attribute classifier training could become biased towards classes with larger volumes. We alleviate this limitation by using focal loss, and will add more less common chart types to balance the dataset.

Chart Attributes. In the preliminary study, we employed several chart types to determine the chart attributes that users tend to concentrate on. We randomly chose samples from both synthesized datasets and real-world collections to diversify the chart attributes. Nevertheless, the attributes may not fully cover user intents. A more comprehensive survey will be conducted to address the issue.

6.2. Future Work

Trade-off Control. To balance between user intent and query chart, we plan to add a slider equipped with dynamic weight control to let the user manage this trade-off by themselves in the near future.

Text Prompt Auto-completion. Previous works [SHKC20, WHS*22] utilize auto-completion as a hint to assist users in the application process, which is also helpful for our system to alleviate the interpretability shift between the user side and the model side. Specifically, we aim to build a mapping table to store the relationship between appropriate prompts and common user text expressions. Then, when a user completes the input, the keywords of it will be extracted by the named entity recognition technique [NS07]. Finally, the corresponding optimal prompt would be automatically completed by the search in the mapping table.

Text-only retrieval. Our system currently supports chart-only retrieval and chart-text retrieval. To offer a more generalized retrieval framework, we plan to add a text-only query, allowing users to obtain their retrieval by only providing text input. The text-only retrieval is beneficial to users who do not have any reference chart. To this end, we aim to prepare several basic charts with primary attributes in advance to serve as temporary query charts, which can be replaced by a more reliable chart from returned results by retrieval. The process can be regarded as an iteration of desirable results, with a clearer search goal and narrowed search scope.

7. Conclusion

In this paper, we propose a user intent-aware chart retrieval framework, which leverages multi-modal input to fuse explicit visual attributes and implicit user intent into the retrieval process. This pipeline consists of two core stages, namely *Annotation*, and *Retrieval*. The *Annotation* stage disentangles visual attributes in the query chart to ensure a flexible combination of user intent attributes used in retrieval. The *Retrieval* stage allows users to integrate the text prompt with the query chart to achieve more customized retrieval. Quantitative experiments prove the superior performance of our method compared with previous methods. Furthermore, we

conduct two case studies containing two common retrieval strategies and interviews to demonstrate the effectiveness and usability. As an initial step to fuse user prompt in chart retrieval, we hope to enhance the prompt capacity to better meets users' growing needs and various usage scenarios. Dataset, code, pretrained model have been released to promote future research in this direction.

Acknowledgment. The authors wish to thank anonymous reviewers for their constructive comments. The work was supported in part by National Natural Science Foundation of China (62172398), and the Red Bird Program at the Hong Kong University of Science and Technology (Guanghzou).

References

- [BBK*18] BEHRISCH M., BLUMENSCHNEIN M., KIM N. W., SHAO L., EL-ASSADY M., FUCHS J., SEEBACHER D., DIEHL A., BRANDES U., PFISTER H., SCHRECK T., WEISKOPF D., KEIM D. A.: Quality metrics for information visualization. *Comput. Graph. Forum* 37, 3 (2018), 625–662. 1
- [BDM*18] BATTLE L., DUAN P., MIRANDA Z., MUKUSHEVA D., CHANG R., STONEBRAKER M.: Beagle: Automated extraction and interpretation of visualizations from the web. In *Proc. ACM CHI* (2018), pp. 594:1–8. 2, 3, 7
- [BFW21] BATTLE L., FENG D., WEBBER K.: Exploring visualization implementation challenges faced by D3 users online. *arXiv preprint arXiv:2108.02299* (2021). 1
- [BLBL22] BAKO H. K., LIU X., BATTLE L., LIU Z.: Understanding how designers find and use data visualization examples. *IEEE Trans. Vis. Comput. Graph.* (2022). 7
- [BMR*20] BROWN T., MANN B., RYDER N., SUBBIAH M., KAPLAN J. D., DHARIWAL P., NEELAKANTAN A., SHYAM P., SASTRY G., ASKELL A., ET AL.: Language models are few-shot learners. In *Proc. NIPS* (2020), pp. 1877–1901. 3
- [BVB*13] BORKIN M. A., VO A. A., BYLINSKII Z., ISOLA P., SUNKAVALLI S., OLIVA A., PFISTER H.: What makes a visualization memorable? *IEEE Trans. Vis. Comput. Graph.* 19, 12 (2013), 2306–2315. 3
- [CCA15] CHEN Z., CAFARELLA M., ADAR E.: DiagramFlyer: A search engine for data-driven diagrams. In *Proc. WWW* (2015), pp. 183–186. 1, 2, 7, 9
- [Chu17] CHURCH K. W.: Word2vec. *Natural Language Engineering* 23, 1 (2017), 155–162. 9
- [DCLT18] DEVLIN J., CHANG M.-W., LEE K., TOUTANOVA K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proc. NAACL* (2018), pp. 4171–4186. 3
- [DT05] DALAL N., TRIGGS B.: Histograms of oriented gradients for human detection. In *Proc. CVPR* (2005), pp. 886–893. 7
- [DW14] DANG T. N., WILKINSON L.: Scagexplorer: Exploring scatterplots by their scagnostics. In *Proc. IEEE PacificVis* (2014), IEEE, pp. 73–80. 1
- [GKSS*19] GOYAL Y., KHOT T., SUMMERS-STAY D., BATRA D., PARIKH D.: Making the V in VQA matter: Elevating the role of image understanding in visual question answering. *Int. J. Comput. Vis.* (2019), 398–414. 3
- [GLKC22] GU X., LIN T.-Y., KUO W., CUI Y.: Open-vocabulary object detection via vision and language knowledge distillation. In *Proc. ICLR* (2022). 3
- [GPM*21] GAL R., PATASHNIK O., MARON H., CHECHIK G., COHEN-OR D.: StyleGAN-NADA: CLIP-guided domain adaptation of image generators. *ACM Trans. Graph.* 41 (2021), 141:1–13. 3
- [HA19] HOQUE E., AGRAWALA M.: Searching the visual style and structure of D3 visualizations. *IEEE Trans. Vis. Comput. Graph.* 26, 1 (2019), 1236–1245. 1, 2, 5, 9
- [HBL*19] HU K., BAKKER M. A., LI S., KRASKA T., HIDALGO C.: VizML: A machine learning approach to visualization recommendation. In *Proc. ACM CHI* (2019), pp. 128: 1–12. 7
- [HMSA08] HEER J., MACKINLAY J., STOLTE C., AGRAWALA M.: Graphical histories for visualization: Supporting analysis, communication, and evaluation. *IEEE Trans. Vis. Comput. Graph.* 14, 6 (2008), 1189–1196. 2
- [HZRS16] HE K., ZHANG X., REN S., SUN J.: Deep residual learning for image recognition. In *Proc. CVPR* (2016), pp. 770–778. 4
- [JKS*17] JUNG D., KIM W., SONG H., HWANG J.-I., LEE B., KIM B., SEO J.: ChartSense: Interactive data extraction from chart images. In *Proc. ACM CHI* (2017), pp. 6706–6717. 2
- [KRS*21] KIM H., ROSSI R., SARMA A., MORITZ D., HULLMAN J.: An automated approach to reasoning about task-oriented insights in responsive visualization. *IEEE Trans. Vis. Comput. Graph.* 28, 1 (2021), 129–139. 2
- [LCF*15] LI Z., CARBERRY S., FANG H., MCCOY K. F., PETERSON K., STAGITIS M.: A novel methodology for retrieving infographics utilizing structure and message content. *DKE 100* (2015), 191–210. 2
- [LGG*17] LIN T.-Y., GOYAL P., GIRSHICK R., HE K., DOLLÁR P.: Focal loss for dense object detection. In *Proc. ICCV* (2017), pp. 2980–2988. 5
- [LGR*20] LU J., GOSWAMI V., ROHRBACH M., PARIKH D., LEE S.: 12-in-1: Multi-task vision and language representation learning. In *Proc. ICML* (2020), pp. 10437–10446. 3
- [LLXH22] LI J., LI D., XIONG C., HOI S.: Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *Proc. ICML* (2022), pp. 12888–12900. 5, 10
- [LOG*19] LIU Y., OTT M., GOYAL N., DU J., JOSHI M., CHEN D., LEVY O., LEWIS M., ZETTMLOYER L., STOYANOV V.: RoBERTa: A robustly optimized BERT pretraining approach. In *Proc. ICLR* (2019). 3
- [LSG*21] LI J., SELVARAJU R., GOTMARE A., JOTY S., XIONG C., HOI S. C. H.: Align before fuse: Vision and language representation learning with momentum distillation. In *Proc. NIPS* (2021), pp. 9694–9705. 5, 10
- [LWW*22] LI H., WANG Y., WU A., WEI H., QU H.: Structure-aware visualization retrieval. In *Proc. ACM CHI* (2022), pp. 409: 1–14. 1, 2, 3, 7
- [MMA18] MAJOOONI A., MASOOD M., AKHAVAN A.: An eye-tracking study on the effect of infographic structures on viewer's comprehension and cognitive load. *Inf. Vis.* 17, 3 (2018), 257–266. 2
- [MTW*12] MOERE A. V., TOMITSCH M., WIMMER C., CHRISTOPH B., GRECHENIG T.: Evaluating the effect of style in information visualization. *IEEE Trans. Vis. Comput. Graph.* 18, 12 (2012), 2739–2748. 1
- [MTW*18] MA Y., TUNG A. K., WANG W., GAO X., PAN Z., CHEN W.: ScatterNet: A deep subjective similarity model for visual analysis of scatterplots. *IEEE Trans. Vis. Comput. Graph.* 26, 3 (2018), 1562–1576. 2, 5, 7
- [NS07] NADEAU D., SEKINE S.: A survey of named entity recognition and classification. *Linguisticae Investigationes* 30, 1 (2007), 3–26. 10
- [PMH17] POCO J., MAYHUA A., HEER J.: Extracting and retargeting color mappings from bitmap images of visualizations. *IEEE Trans. Vis. Comput. Graph.* 24, 1 (2017), 637–646. 2
- [PSP21] PARSONS P., SHUKLA P., PARK C.: Fixation and creativity in data visualization design: Experiences and perspectives of practitioners. In *Proc. IEEE VIS* (2021), pp. 76–80. 1

- [PWS*21] PATASHNIK O., WU Z., SHECHTMAN E., COHEN-OR D., LISCHINSKI D.: StyleCLIP: Text-driven manipulation of styleGAN imagery. In *Proc. ICCV* (2021), pp. 2085–2094. 3
- [QDH*22] QIN X., DAI H., HU X., FAN D.-P., SHAO L., VAN GOOL L.: Highly accurate dichotomous image segmentation. In *Proc. ECCV* (2022), pp. 38–56. 4
- [RKH*21] RADFORD A., KIM J. W., HALLACY C., RAMESH A., GOH G., AGARWAL S., SASTRY G., ASKELL A., MISHKIN P., CLARK J., ET AL.: Learning transferable visual models from natural language supervision. In *Proc. ICML* (2021), pp. 8748–8763. 2, 3
- [RNS*18] RADFORD A., NARASIMHAN K., SALIMANS T., SUTSKEVER I., ET AL.: *Improving language understanding by generative pre-training*. Tech. rep., 2018. 3
- [RTOT06] RODRIGUES J. F., TRAINA A. J. M., OLIVEIRA M. C. F. D., TRAINA C.: Reviewing data visualization: an analytical taxonomical study. In *Proc. ICCV* (2006), pp. 713–720. 2
- [RWC*19] RADFORD A., WU J., CHILD R., LUAN D., AMODEI D., SUTSKEVER I., ET AL.: Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9. 3
- [SGP*18] STITZ H., GRATZL S., PIRINGER H., ZICHNER T., STREIT M.: Knowledgepearls: Provenance-based visualization retrieval. *IEEE Trans. Vis. Comput. Graph.* 25, 1 (2018), 120–130. 2
- [SHKC20] SETLUR V., HOQUE E., KIM D. H., CHANG A. X.: Sneak pique: Exploring autocompletion as a data discovery scaffold for supporting visual analysis. In *Proc. ACM UIST* (2020), pp. 966–978. 10
- [SHL*16] SIEGEL N., HORVITZ Z., LEVIN R., DIVVALA S., FARHADI A.: Figureseer: Parsing result-figures in research papers. In *Proc. ECCV* (2016). Springer, pp. 664–680. 1, 2
- [Shn96] SHNEIDERMAN B.: The eyes have it: a task by data type taxonomy for information visualizations. In *Proc. IEEE Symp. Vis. Lang.* (1996), pp. 336–343. 1
- [SKC*11] SAVVA M., KONG N., CHHAJTA A., FEI-FEI L., AGRAWALA M., HEER J.: ReVision: Automated classification, analysis and redesign of chart images. In *Proc. ACM UIST* (2011), pp. 393–402. 2, 9
- [SLC*23] SHI Y., LIU P., CHEN S., SUN M., CAO N.: Supporting expressive and faithful pictorial visualization design with visual style transfer. *IEEE Trans. Vis. Comput. Graph.* 29 (2023), 236–246. 2
- [SSK06] SCHNEIDEWIND J., SIPS M., KEIM D. A.: Pixnostics: Towards measuring the value of visualization. In *Proc. IEEE VAST* (2006), pp. 199–206. 1
- [SWS*22] STROBELT H., WEBSON A., SANH V., HOOVER B., BEYER J., PFISTER H., RUSH A. M.: Interactive and visual prompt engineering for ad-hoc task adaptation with large language models. *IEEE Trans. Vis. Comput. Graph.* (2022). 3
- [TM04] TORY M., MÖLLER T.: Rethinking visualization: A high-level taxonomy. In *Proc. IEEE InfoVis* (2004), pp. 151–158. 2
- [WHS*22] WANG Y., HOU Z., SHEN L., WU T., WANG J., HUANG H., ZHANG H., ZHANG D.: Towards natural language-based visualization authoring. *IEEE Trans. Vis. Comput. Graph.* (2022). 10
- [XBK*15] XU K., BA J., KIROS R., CHO K., COURVILLE A., SALAKHUDINOV R., ZEMEL R., BENGIO Y.: Show, attend and tell: Neural image caption generation with visual attention. In *Proc. ICML* (2015), pp. 2048–2057. 3
- [YHZ23] YE Y., HUANG R., ZENG W.: VISAtlas: An image-based exploration and query system for large visualization collections via neural image embedding. *IEEE Trans. Vis. Comput. Graph.* (2023), 1–15. 2
- [YZF*22] YUAN L., ZENG W., FU S., ZENG Z., LI H., FU C. W., QU H.: Deep colormap extraction from visualizations. *IEEE Trans. Vis. Comput. Graph.* 28, 12 (2022), 4048–4060. 2
- [ZDCC21] ZENG W., DONG A., CHEN X., CHENG Z.-L.: VIStory: interactive storyboard for exploring visual information in scientific publications. *J. Vis.* 24, 1 (2021), 69–84. 2
- [ZFC*23] ZHANG T., FENG H., CHEN W., CHEN Z., ZHENG W., LUO X.-N., HUANG W., TUNG A. K.: ChartNavigator: an interactive pattern identification and annotation framework for charts. *IEEE Trans. Knowl. Data Eng.* 35, 2 (2023), 1258–1269. 2
- [ZFF22] ZHAO J., FAN M., FENG M.: ChartSeer: Interactive steering exploratory visual analysis with machine intelligence. *IEEE Trans. Vis. Comput. Graph.* 28, 3 (2022), 1500–1513. 2
- [ZYLL22] ZHOU K., YANG J., LOY C. C., LIU Z.: Learning to prompt for vision-language models. *Int. J. Comput. Vis.* 130, 9 (2022), 2337–2348. 10
- [ZZF*22] ZHANG R., ZHANG W., FANG R., GAO P., LI K., DAI J., QIAO Y., LI H.: Tip-adapter: Training-free adaption of CLIP for few-shot classification. In *Proc. ECCV* (2022), pp. 493–510. 5, 10