Visual Informatics 4 (2020) 24–40

Contents lists available at ScienceDirect

Visual Informatics

journal homepage: www.elsevier.com/locate/visinf

A survey on automatic infographics and visualization recommendations

Sujia Zhu, Guodao Sun *, Qi Jiang, Meng Zha, Ronghua Liang

College of Computer Science and Technology, Zhejiang University of Technology, China

ARTICLE INFO

Article history:
Available online 5 August 2020

Keywords:
Automatic visualization
Data-driven
Knowledge-based
Machine learning
Visual embellishments

ABSTRACT

Automatic infographics generators employ machine learning algorithms/user-defined rules and visual embellishments into the creation of infographics. It is an emerging topic in the field of information visualization that has requirements in many sectors, such as dashboard design, data analysis, and visualization recommendation. The growing popularity of visual analytics in recent years brings increased attention to automatic infographics. This creates the need for a broad survey that reviews and assesses the significant advances in this field. Automatic tools aim to lower the barrier for visually analyzing data by automatically generating visualizations for analysts to search and make a choice, instead of manually specifying. This survey reviews and classifies automatic tools and papers of visualization recommendations into a set of application categories including network-graph visualizations, annotation visualizations, and storytelling visualization. More importantly, this report presents several challenges and promising directions for future work in the field of automatic infographics and visualization recommendations.

1. Introduction

As the demand for rapid analysis of visualization grows, there is an increasing requirement to design visualization tools, which allow users to efficiently generate visualizations. Shown in Fig. 1, authoring tools are increasingly towards automatic. The evolution of them can be divided into four steps. Firstly, programmatic step for visualizing data includes imperative languages and libraries. Designers need to have an ability of programming to get infographics, such as *D3* (Bostock et al., 2011), *Vega-Lite* (Satyanarayan et al., 2016), *ECharts* (Li et al., 2018), and *VisComposer* (Honghui et al., 2018). These tools are designed for users who are familiar with coding and visualizations. Secondly, visual building step for easy visualizations includes template editing (*iCharts* (García et al., 2007) and *RAWGraphs* (Mauri et al., 2017), shelf configuration (Polestar (Pioch and Everett, 2006), *Voyager* (Wongsuphasawat et al., 2015), and *Voyager 2* (Wongsuphasawat et al., 2017)), visual building (*iVisDesigner* (Ren et al., 2014), *Data Illustrator* (Liu et al., 2018), *VisComposer* (Mei et al., 2018a), and *Charticulator* (Ren et al., 2018a)). These tools are designed for users who are familiar with coding but not familiar with visualizations. Users need to pre-conceive blueprints, then interact with the system, such as clicking or dragging with visual items, to get more expressive and aesthetic visualizations.

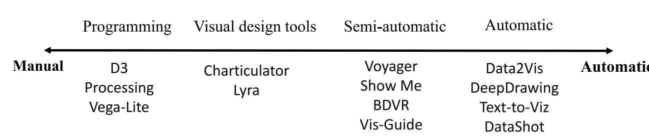

Fig. 1. Tools/systems for generating visualizations. Infographics are created with a spectrum of tools automatically or manually.

Thirdly, the semi-automatic step is involved with few interactions to efficiently obtain visualizations, such as *SAGE* (Roth et al., 1995) and *BDVR* (Gotz and Wen, 2009). Lastly, automatic step is designed for no-human involved tools to efficiently get visualization recommendations, such as *Text-to-Viz* (Cui et al., 2019) and *Click2Annotate* (Chen et al., 2010a). These tools are gradually emerging for users who are not familiar with visualization or coding.

Effective creation of infographics enables designers and analysts to reduce tedious work, design successful strategies and make informed decisions. Combining data contents with visual embellishments, infographics can effectively deliver more messages in an engaging and memorable manner than tedious raw data. However, it is difficult for users who are familiar with visualizations to design expressed and aesthetic visualizations. They need to understand visual encoding, data characteristics, and common types of visualization. Even experts need to have an idea about design target and take efforts to investigate data

* Corresponding author.
E-mail address: guodao@zjut.edu.cn (G. Sun).

https://doi.org/10.1016/j.visinf.2020.07.002

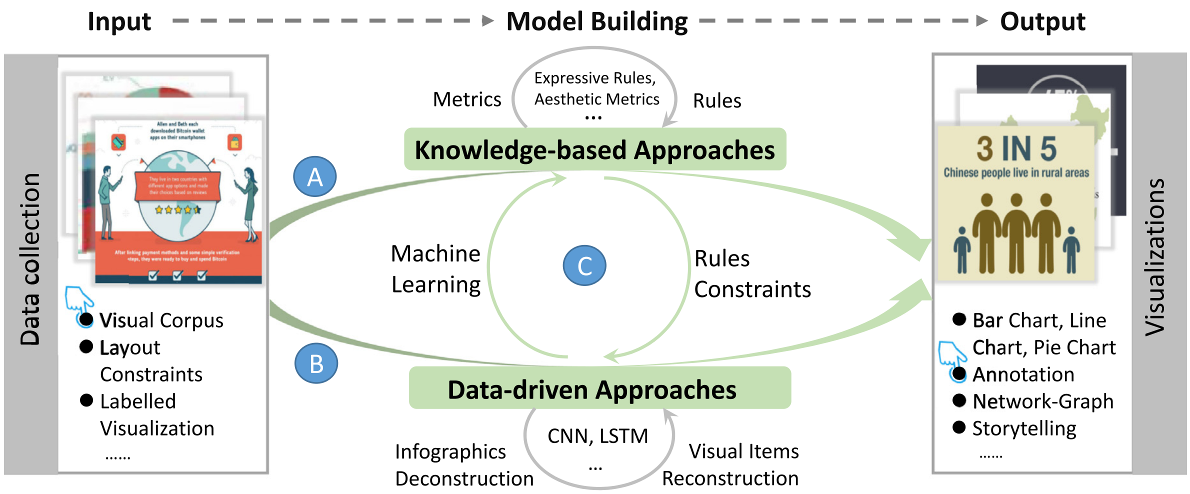

Fig. 2. Three paths of automatic infographics and visualizations. Existing works are divided into three categories: (A) knowledge-based approaches; (B) data-driven approaches; and (C) hybrid approaches.

analysis with kinds of attempts for designing an ideal visualization. Furthermore, designers need to consider not only perceptual effectiveness, but also visual styles when creating an infographic, which is an inefficient process.

To address these challenges, automated infographics (Chen et al., 2019; Cui et al., 2019; Wang et al., 2019b) have been developed in recent years through a proper combination of machine learning/user-defined rules and visual elements. It does not require users to have expertise in visualization and can provide users with professional infographics.

The tool for automatically generating visualization requires that no or few human are involved. Mackinlay (1986) presented the first widely accepted presentation tool that automatically designs efficient visualizations (such as bar charts, scatter plots, and connected graphs) of relational data. This is the foundational and inspiring study on the automated creation of infographics. For automatically generating visualizations, they described two challenges: visual encoding for information and design standards for visualizations. Later, Wang et al. (2019b) provided more specific challenges of automatic generation tools. The first challenge is to extract features from datasets and organize characteristics into a meaningful ground truth. The extracted features should be reliable and interesting, and the identified ground truth should be meaningful and easy-understand. The second challenge is making a choice of proper visualizations that can illustrate the meaningful stories.

Our motivations in conducting this survey are two folds. First, we aim to review the latest developments of researches for semi-automatically or automatically infographics generating tools and provide a concise and comprehensive review of the field. However, researches and tools for generating visualizations in the last few years mainly focus on some narrow topics such as manually infographics (Ren et al., 2018a; Mei et al., 2018a). Several surveys (Grammel et al., 2013; Mei et al., 2018b) for generating visualizations published in the last few years mainly focus on some narrow topics of interactive tools. A comprehensive survey that reviews the research of semi-automatically or automatically generating visualizations is still absent. Second, this survey aims to organize, classify, and compare recent researches to provide a critical assessment of the research and understand current research trends. We organize and classify existing researches on automatically generating infographics and their applications.

The contributions of this paper are as follows. First, we discuss existing works on automated visualization design and recommendation generators that combine advanced technologies of machine learning and guidelines obtained from visual knowledge researches. This paper presents a comprehensive survey of the developments of automatic infographics. Second, it provides a novel classification of the results. We categorize these systems into knowledge-based, data-driven, and hybrid automatic design tools for visualization. In addition, we described applications of these tools. Third, it identifies new research challenges and future work, which can help related researchers to enhance the understanding of this field.

The structure of the paper is as follows. In Section 2, we categorize the approaches into three parts: knowledge-based, data-driven, and hybrid visualization design tools. Theories are discussed and demonstrated in an intuitive manner. In Section 3, 4, and 5, we comprehensively and detailly describe paper classifications and applications using knowledge-based, data-driven or hybrid approaches respectively. Besides, we analyze the research challenges and trends in the specific applications. Section 6 review the researches of evaluating infographics and infographics generating tools, which are informed for invalidating automatic infographics generating tools. In addition, some considerations are summarized for future researches in this field. Finally, Section 7 concludes the paper and outlines future challenges in this research domain.

2. Automatic infographics, models, and framework

Data visualization technology has developed rapidly, and a large number of visualization methods and technologies have appeared. However, choosing a suitable visualization expression requires not only exploring data but also understanding the characteristics of expression and structure of infographics. This makes it difficult for non-domain users to design appropriate visualizations.

Automatic visualization design focuses on generating visualizations with no or few human-involved operations. This can promote the design of effective visual encodings, enhance rapid visual exploration, and facilitate efficient visual analysis. Mackinlay (1986) proposed a foundational idea of codifying Bertin's semiology of graphics as algebraic operators to automatically create graphical visualizations for "node-link" data. Casner (1991) extended this work by comparing and quantifying visualization alternatives via a series of metrics on generated visualization depending on tasks. Goldstein et al. (1994) added alternative types of visualizations, quantify alternatives, and then rank the results of effective and expressive recommendations. However,

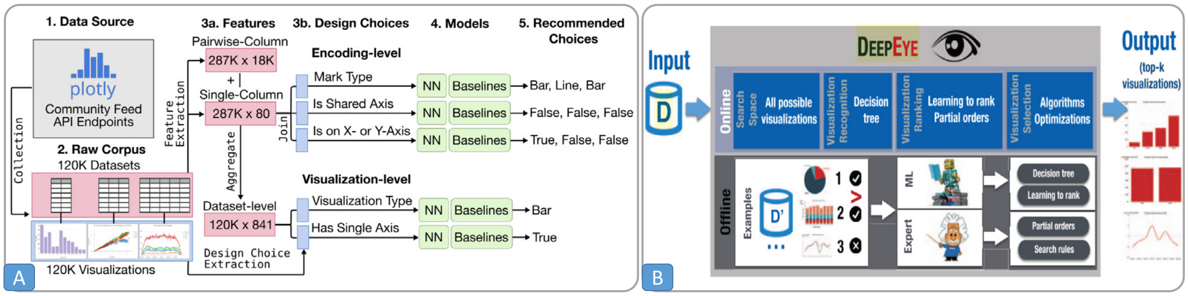

Fig. 3. Two kind of types of automatic generator. (A) Workflow of data-driven automatic generator in *VizML* (Hu et al., 2019) (B) Workflow of hybrid automatic generator for visualizations in *DataTone* (Luo et al., 2018).

their work did not consider or discuss in the aspects of user knowledge for their automatic visualizations. Chen et al. (2019) then proposed an idea of deconstructing and reconstructing infographics by employing user-defined rules and data-driven constraints.

As shown in Fig. 2, after reviewing the previous work, we categorized the models into three kinds for creating visualization tools. Fig. 2(A) is a data-driven visualization model (Hu et al., 2019; Cui et al., 2019), Fig. 2(B) is a user-defined rule to establish a visualization, Fig. 2(C) is hybrid model (Luo et al., 2018; Gao et al., 2015) that incorporates data-driven and knowledge-based models.

For the first data-driven model, a natural way to capture human perception is by learning visualization examples. Then, some researchers proposed machine learning-based approaches (Hu et al., 2019; Cui et al., 2019). They employed machine learning techniques and visual embellishments into the generation of visualizations. Firstly, visual elements are deconstructed and labeled from infographics. Then, a machine learning model is trained to predict visualizations, and visual elements are reconstructed to create recommendations. Hu et al. (2019) introduced a framework to depict the process of building the data-driven model for automatic creation of information visualizations. Fig. 3(A) illustrates the entire process of building the data-driven model. The process starts by collecting infographics and then extracting visual components from infographics. Thus, a raw visual corpus with associated descriptions is obtained. After that, models are trained to predict potential visualizations, and layout constraints are defined for generating visualization alternatives. Finally, layout algorithms are proposed for quantifying and ranking visualization recommendations for users to select.

The second rules-based model is mainly involved with user-defined rules based on certain obtained experiences of the design of the visualization. Using machine learning models as black-boxes has shortcomings such as the lack of interpretability. Thus, some researchers proposed user-defined rules to make constraints for improving the results of visualization recommendations.

Meanwhile, for the third hybrid model, data-driven models together with knowledge-based models can often lead to better analysis results. Therefore, the combination of data-driven models and knowledge-based models are the ideal method for building models of automatically generating visualizations. Luo et al. (2018) also introduced a pipeline to describe the steps of building the hybrid model. Fig. 3(B) illustrates the process of building the hybrid model, which consists of two parts: offline part and online part. For the offline part, two machine learning models are trained. One is a decision tree to determine whether provided data and a reference visualization is true or false, and the other is a learning-to-rank model that ranks visualizations. For the online part, all possible visualizations are generated to qualify and rank visualization recommendations.

In summary, the establishment of these models mainly involves the following two challenges. The first challenge infers to infographics collection, visual items identification, and visual items labeling. The second one concerns the study on user-defined rules and algorithms of automatically generating visualization. Fig. 4 describes the summary of some automatic visualization generators.

3. Data-driven automatic visualizations

This section mainly introduces applications of data-driven automatic visualizations, which use data-driven constraints or machine learning techniques to predict user-intend visualizations.

3.1. Statistical automatic visualizations

Most common infographics in our daily life are statistical charts, such as pie charts, bar charts, line charts, etc. Many researchers have proposed corresponding methods and tools to automatically generate statistical charts. Therefore, this section mainly introduces layout methods, automatic tools, and existing challenges from two aspects of data-driven rules and machine learning models.

The former is involved with two aspects. One is user-defined constraints, which is based on the characteristics of data, to automatically generate all possible visualizations. The other is quality metrics to quantify and rank all generated visualizations. Saket et al. (2018) implemented a research to verify the effectiveness of most common type of visualizations (see Fig. 5) through general data analysis tasks using two kinds of data (cars dataset and movies dataset). When factors (time, accuracy and preference) are taken into consideration, the result are revealed and convergent to five guidelines: using bar charts for finding groups, using line charts for finding correlations, using scatterplots for finding anomalies, avoiding using line charts to precisely find the value of a specific data point, and avoiding using tables and pie charts for relational information. Based on the above results, They created a decision tree as a predictive model and developed *Kopol1* as a recommender to create visualization alternatives when given tasks and the types of dimensions.

Kim and Heer (2018) accomplished an research to estimate subject performance across kinds of tasks and characteristics of datasets. During their experiment, users are asked to conduct tasks for making a combination of features and making visualization alternatives a rank. Based on the result of their research, they developed a model for a dozen of scatterplots encodings, the types of target, and the cardinality and entropy of some data dimensions.

The latter employs machine learning models to predict all possible visualizations. Key et al. (2012) proposed *Vizdeck* for the disorganized relational dataset.

Research	Data Source	Type	ML	Model	Ranking	Design Space	Input	Output
VizML, Hu et al.(2019)	Plotly	Static	√	NN	√	5 types	Visualization Pairs	BLPS
DeepEye, Luo et al.(2018)	Heterogeneous	Static	√	Statistics	√	4 types	Pairwise Comparisons	BLPS
Data2Vis, Dibia et al.(2019)	Voyager	Static	√	DNN	√	-	Number	BLPS
Text-to-vis, Cui et al.(2019)	Heterogeneous	Static	√	CNN+CRF	√	Describe, Visual	Proportion-Related Text	Combination
Timeline, Chen et al.(2019)	Web	Static	√	Mask R-CNN	X	Scale, Layout	Bitmap Image	Combination
DeepDrawing, Wang et al.(2019)	Tool	Static	√	LSTM	X	Presentation, Structure	Node-Link Data	Graph
DataShot, Wang et al.(2019)	Heterogeneous	Static	√	Decision Tree	√	Content,Present,Sheet	Data facts	PDF
DataSite, Zhe et al.(2018)	Films,Records	Interactive	X	Statistics	X	-	Number, Category, Temporal	HLS, heatmap
Voder, Srinivasan et al.(2018)	Heterogeneous	Interactive	X	Natural Language Generator	X	-	Number, Category,	HBS,Donut, Strip
Show Me, Mackinlay et al.(2007)	Heterogeneous	Interactive	X	-	√	3 tasks	Text, Temporal, Number, Boolean	Small multiples
VisGuide, Cao et al.(2020)	Heterogeneous	Interactive	√	Linear Regression	X	Chart Selection	-	BL
ManyEyes, Wattenberg et al.(2007)	Heterogeneous	Interactive	X	User-defined	X	10+types	-	BLPSH, Treemap
DataClips, Amini et al.(2016)	Heterogeneous	Interactive	X	User-defined	X	5 types	-	Data videos
VizDeck, Key et al(2012)	Heterogeneous	Interactive	√	Linear Model	√	9+types	-	Dashboards
Foresight, Demiralp et al.(2017)	-	Interactive	X	Statistics	√	3+types	Number	-
Lai et al. (2020)	Heterogeneous	Interactive	√	Mask-RNN	√	6 types	Charts	Annotation
Voyager/Voyager2, Wongsuphasawat et al.(2015)	Films,Records	Interactive	X	Statistics	√	-	-	-
Vis-Annotator, Lai et al.(2020)	Web	Static	√	Mask R-CNN	√	Location, Text	Charts, Text	Annotation
Contextifier, Hullman et al.(2013)	Articles	Interactive	X	Feature-generator-based	√	4 types	Articles	Annotation
TSIs, Bryan et al.(2017)	Heterogeneous	Interactive	X	Feature-generator-based	√	3 tasks	-	Annotation
NewsViews, Gao et al.(2014)	News York	Interactive	X	Thematic-maps-generator Locations identification, Annotations-generator-based	√	6 modules	Articles	Annotation
AutoCaption, Liu et al.(2020)	Web	Static	√	Deep-Learning	X	3 modules	SVG-based Chart	Caption
Draco, Moritz et al.(2019)	Heterogeneous	Static	√	RankSVM	√	-	A query (dataset, partial specifications, and tasks)	-
Zoya et al.(2017)	Web	Static	√	NN, ResNet-50	X	Textual, Visual	Infographics	Visual hashtags

Fig. 4. Summary of certain automatic visualization generators. Bar(B), Line(L), Pie(P), Scatter plot(S), and Histogram(H) are common visualizations that most automatic tools can generate. "Ranking" means "Ranking Support". Systems/tools support kinds of algorithms to rank visual or textual presentations.

It is the first automatic system to recommend visualizations by training a model, which learns the correlations among data records and properties of visualization to predict user-preferred charts. Zhe et al. (2018) then took a further step and proposed *DataSite* by concentrating on continuous computation from a library of automatic algorithms. Srinivasan et al. (2018) introduced *Voder*, which use a series of statistical functions to combine natural language generation techniques into their system. Dibia and Demiralp (2019) presented *Data2Vis*, which use Recurrent Neural Network to automatically translate JSON-format datasets into visualization specifications. This approach combined with long short-term memory to created an end-to-end generation model, which is trained with about 4300 visualization examples created in *Vega-Lite* (Satyanarayan et al., 2016). Fig. 6 shows visualizations examples, which are generated by *Data2vis* when given race dataset.

However, it is a critical challenge for extracting and interpreting visual elements from the diagrams. To address this issue, Battle et al. (2018) proposed an approach for automatically extracting visualizations from the Internet and describing visualizations with annotation. From the collected visualizations, most common types of visualizations are bar charts, line charts, scatter charts, and geographic maps. Thus, they extract SVG-based graphs on the web and automatically classify and describe them by type. Based on this approach, Hu et al. (2019) introduced *VizML*. The model are trained via 1000,000 dataset-visualization pairs collected from public visualization communities, which were orders of magnitude larger than that of *Data2Vis* and *DeepEye*. *VizML* provide a importance measurement of interpretation for the characteristics and integrated it into visualization interfaces.

Above techniques can automatically generate common infographics, there are some challenges:

- Scalable methods for collecting and labeling training data. It is a lack of public and comprehensive visual primitive library. This challenge involves not only the identification and extraction of items from a large number of infographics, but also the labels and description of visual items, which may express different meanings in fields.
- Lacking interpretability and explainability of machine learning models. The interpretability combined with the user-defined rules can further improve the quality of visualization recommendations.

3.2. Automatic annotations

Annotation acts an increasingly significant role in helping people to understand the chart and expressing information from designers. It has been widely used in a variety of applications such as efficient analysis of assisted story-telling (Bylinskii et al., 2017), error detection (Grammarly, 2012), and classification (Júnior et al., 2017). Annotations visually point to salient data characteristics or visual elements. This section reviews automatic data-driven annotations while knowledge-based annotations can be seen in Section 4.2.

A lot of work uses various methods to achieve the purpose of automatic annotation. Kandogan (2012) presented a workflow, which mainly includes: visual feature detection, feature ranking, and generating annotations. Lai et al. (2020) then proposed a comprehensive workflow for automatic annotation, which is inspired by the process of manually annotating. The workflow consists of four parts: object detection for identifying visual entities and visible texts in the chart, NLP for parsing the description to generate queries for the described entities, annotation for fulfilling queries to anchor each sentence to locations of corresponding image, and describing charts. The process of automatically annotating involved with automatically identifying and describing/highlighting key visual elements (Kandogan, 2012).

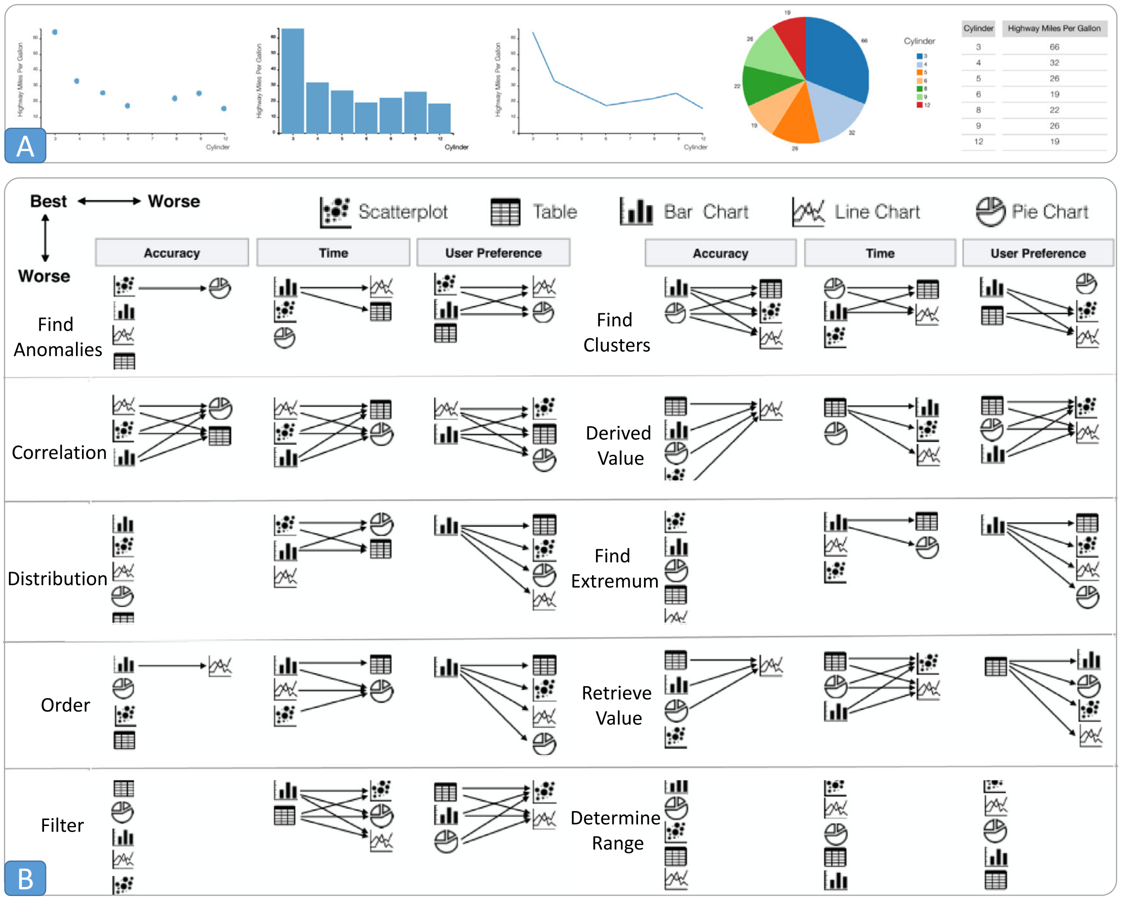

Fig. 5. (A) General types of visualizations used in Saket et al. (2018), such as scatterplot, bar chart, line chart, pie chart, and table. (B) Correlations between visualization types across tasks and performance metrics.

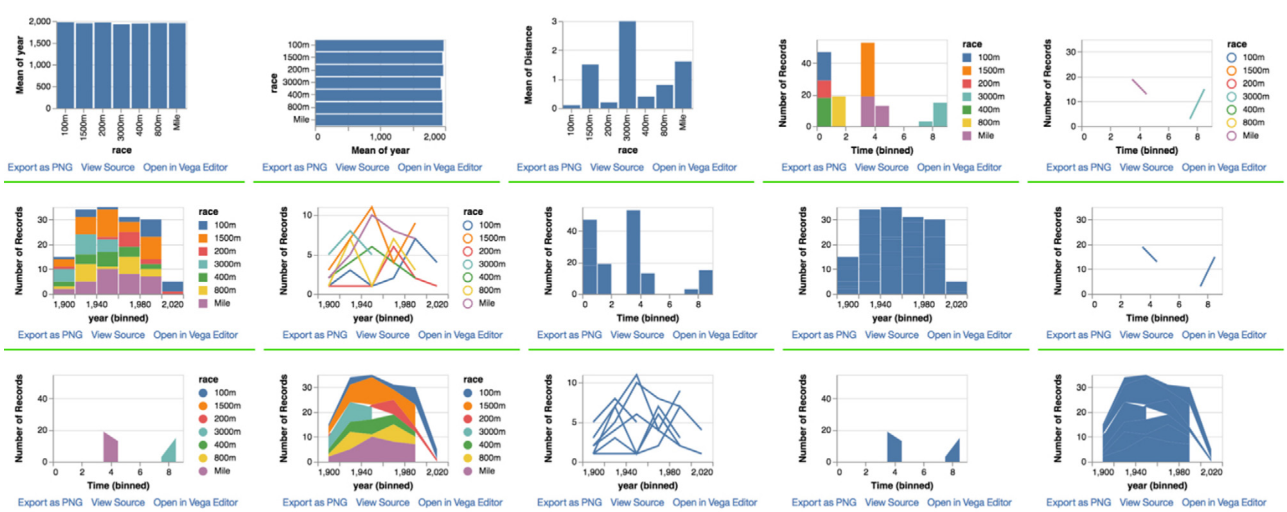

Fig. 6. Visualization alternatives generated by *Data2vis*.

A lot of work uses various techniques to achieve the purpose of automatic annotation. Some researchers use common statistic methods to automatically identify and extract significant information from data. Kandogan, 2012 computed clusters, outliers and trend in point-based charts. Then special points are automatically highlighted and labeled. Bryan et al. (2016) presented temporal summary images, which provide automatic annotations to recommend data points of interest for three type of data attributes: numerical vectors, storylines, and alluvial diagrams. Fig. 7(A) illustrates the approach of finding points of interest. Then, they defined the importance calculation of each annotations' dimension and points of interest. Based on this function, the importance of all possible annotations are quantified, filtered, and ranked. Finally, they applied an existing kernel density-based algorithm to place and display annotations. Fig. 7(B) and (C) show the examples of Temporal Summary Images (*TSIs*)

Kong and Agrawala (2012) used different machine learning models to extract graphical marks aiming at various charts. To

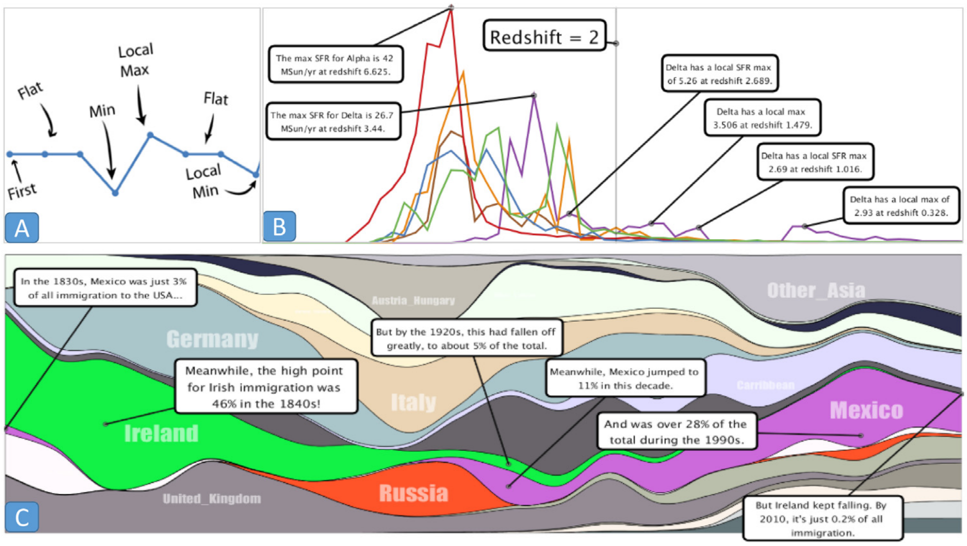

Fig. 7. (A) An approach of finding points of interest from time-series data, the characteristics of a vector are extracted by computing special changes, such as sharply increased point and decrease point. (B) An example of *TSIs* after employing numerical vectors method. (C) An example of *TSIs* after employing alluvial diagram method.

create most graphic overlays, users only need to know the visual marks and axis attributes of the encoded data, and do not need to access the underlying data values. Therefore, they extracted the features and marks in the chart through a combination of automatic extraction and semi-automatic extraction, then they generated some notes (reference structures, highlights, and summary statistics). However, the system only considers two kind of visual elements(marks and axis). This may generate the opposite result when a chart is combined with many visual elements. Júnior et al. (2017) defined the design space (annotation form and annotation target type) to design and implement ChartAccent. Bylinskii et al. (2017) proposed a computational system, which can automatically produce various textual and visual hashtags when an infographic is inputted. They scraped above 63,000 static infographics from the visual website, and each of them is manually categorized, labeled, and described. They further merged redundant tags to obtain a dataset with hundreds of tags. The method mainly consists of two parts: predicting the text mark of the information map and locating the most representative visible area. Text information is automatically extracted from the information map and convert it into a 300-dimensional word2vec representation. This mean word2vec representation was fed into two single-hidden-layer neural networks for predicting the category and tags of each infographic.

More recently, researchers use machine learning to create annotations for visualizations. These approaches reduced the user's cognitive burden. Liu et al. (2020) proposed *AutoCaption*, which employs the GAN scheme embedded with Res-Net generator to recognize the visual significance of SVG-based visualizations. Meanwhile, they used a common statistic-based method to identify the correlations among visual elements. Lai et al. (2020) used a Mask R-CNN model to identify and extract visual elements in the target visualizations, along with their visual properties. Corpus consists of three kind of types: bar charts, pie charts, and scatterplots. They are all filtered after randomly collected from the web. For each type, Two-thirds of the infographics are assigned to train, with the rest are utilized to be validated. Fig. 8 shows the examples of this work.

Above techniques satisfy the automatic annotation, some challenges are remained as following:

- A short of public corpus. Infographics are searched and collected online and then labeled manually, which is a tedious and time-consuming process.
- Identifying the wrong features of infographics. Machine learning technology will inevitably suffer from recognition deviation, which will result in errors in automatically annotated objects.
- Recognizing semantic incorrectly. Existing semantic recognition technology will inevitably have errors, which will lead to the incorrect expression of the meaning of automatic annotation.

3.3. Graph and network visualizations

Graphs can be visually represented by node-link diagrams, matrix chart, or hybrid visualization of node-link diagrams and matrix chart (Sun et al., 2013). Graphs are widely used to illustrate the correlations among data instances. Thus, researchers focused to study the approaches of generating a graph to visualize data more intuitively. This helps users to understand and explore kinds of network data such as complex social media data.

Graph drawing methods are often based on various layout principles: from spring-embedder-based layout algorithms (Frick et al., 1994) and dimension-reduction-based techniques (Gansner et al., 2004) to energy-based approaches (Jacomy et al., 2014). Users need to acquire various parameters to achieve the desired graph drawing. Nowadays, Some researchers devoted to employing machine learning to the creation of network visualization (Moldovan et al., 2018). Some literature had been published for reviewing graph drawing with the machine learning models (dos Santos Vieira et al., 2015). Interested readers can refer to this complete survey for more details about past researches. This subsection reviews only recent researches.

There are a lot of approaches based on nonlinear dimensionality reduction and machine learning models, which can automatically encode complex graph construction into simple low-dimensional embeddings. Such as matrix factorization-based algorithms (Ou et al., 2016) and graph neural networks approaches (You et al., 2018).

Hamilton et al., 2017 then developed a framework to unify explain the advances and further strengthen some significant work. Traditional machine learning methods depend on user-defined rules to identify and extract features of graphs to encode the structural information. Kipf and Welling (2016) proposed a latent variable model for graph-structured data with a decoder and loss

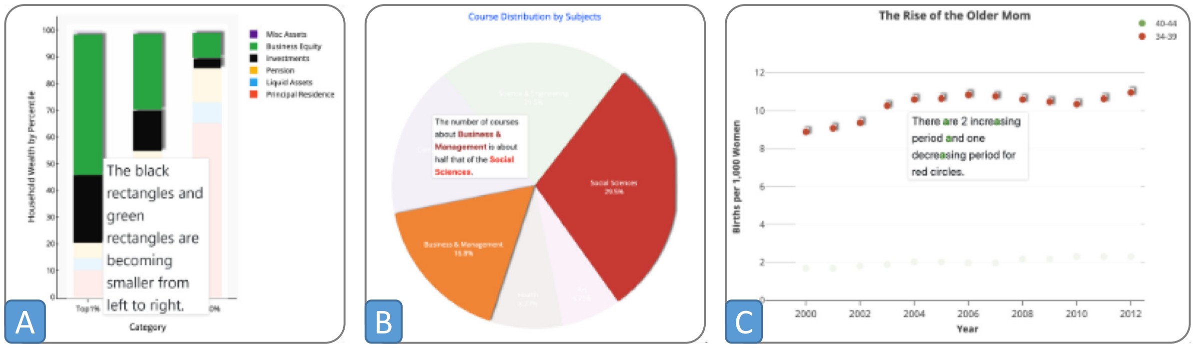

Fig. 8. Annotated cases in Lai et al. (2020). The charts in (A) and (C) visually illustrate examples with property-based descriptions. The chart in (B) visually presents a case with text-based descriptions.

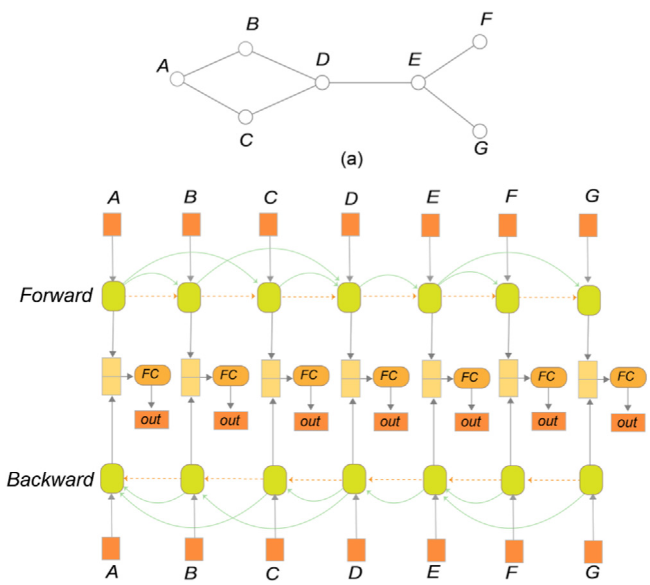

Fig. 9. An illustration of graph-LSTM-based model architecture in *DeepDrawing* (Wang et al., 2019a).

function similar to the graph factorization method. They validated the usefulness with an effective result on a link prediction task in citation networks.

Wang et al. (2019a) proposed a graph-LSTM-based approach, *DeepDrawing*, which can automatically analyze the structure of data graph and generate a graph drawing when the graph/network structure is input. The structure of model is shown in Fig. 9. Commonly used LSTM architecture is usually a linear chain, and one of their main limitations is that they can only model sequence data. However, for graph drawing, the dataset of input is usually a structure of graph/network, rather than a linear chain. The layout location of one point in the graph depends on all other points directly or indirectly connected to it. Using the general LSTM model, this dependency information can still be weakened or lost, especially for LSTM units that are far apart from each other. This model adds direct connection among LSTM units to model the topology of the input network.

Then, a linear chain among adjacent units of a hierarchy LSTM is utilized to propagate the overall state of the previous graph nodes to the following nodes along the chain. The original graph is converted into a series of nodes as input data of a model, each LSTM unit takes a node's feature vector as input to generate the output state of each node. The green arrows among the units visually encode the solid edges in the graph structure, and the dotted yellow arrows encode the "false" edges, which connect adjacent nodes in the sequence of BFS ordered nodes. Considering that the graph is usually not serial data, each of nodes should be considered when trying to draw a graph visualization, since subsequent nodes in the sequence will also affect the position of the previous node in the actual graph drawing process. To better simulate this interaction, they introduced reverse propagation by simply reversing the direction of the link in forwarding propagation. Then, the output results are combined of LSTM units in forwarding and backward propagation and input into the feature vector, which is further input into the fully connected layer to generate the final two-dimensional coordinates of each node.

Employing a machine learning model to learn the layout rules and aesthetic standards in drawing, graphs are automatically generated to help users get out of tedious operations on parameters adjustment. However, there still remain some challenges:

- The diversity of users' requirements results in a complicated and difficult establishment of certain machine learning models. Thus, it is a difficult task to use the existed machine learning techniques to train.
- The lack of interpretation. There is no information about what graph layout aspects are learned by machine learning model.

3.4. Automatic storytelling visualizations

Narrative visualization is a significant branch of visualization. An increasing number of people attempt to visually describe patterns such as news reporting (Satyanarayan and Heer, 2014). Users not only need to explore data to find points of interest but also require to have the abilities about relevant visualization knowledge. Through various visualization attempts, proper visualization is obtained. Therefore, it is a time-consuming process for designers, especially for non-domain users to design an intended narrative visualization. Thus, some experts proposed approaches for automatically generating storytelling visualization (Chen et al., 2019; Gao et al., 2014). Visually telling a story usually demands to choose an order in which to illustrate visualizations, such as time-series and spaces.

Telling stories on geographic map is one common approach. Information can be expressively and effectively conveyed on map. Gao et al. (2014) presented *NewsView*, an automated news visualization interface that generates interactive, annotated maps without requiring professional designers. *NewsView* can provide customized geographic visualizations for news, which is driven by 1.5M location-related articles from the New York Times, table database, and table crawled databases. They follow three criteria, which are general to other types of data. The first is a variable-to-article similarity, which is devised to calculate pointwise mutual

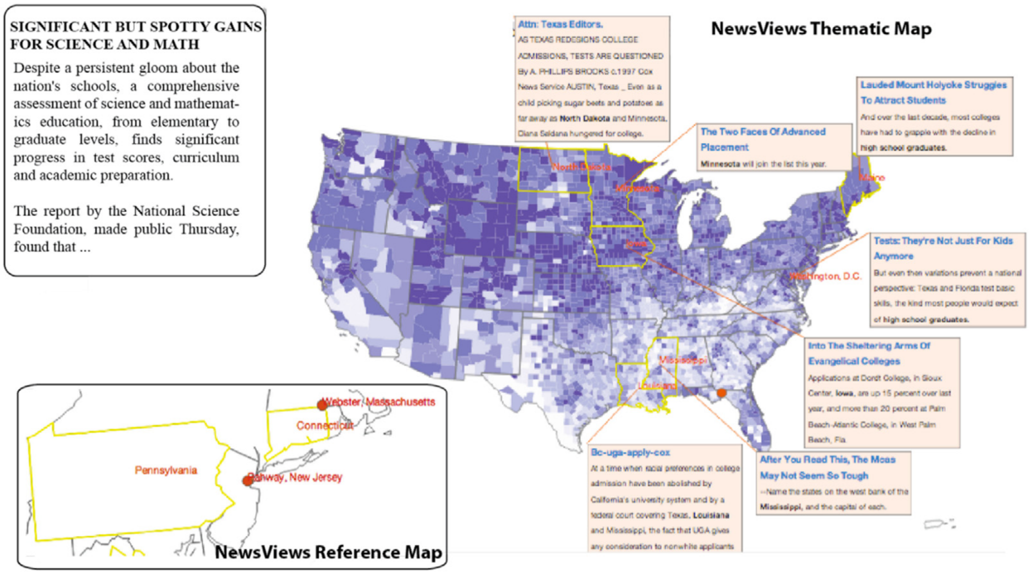

Fig. 10. An application of *NewsView* (Gao et al., 2014). It automatically depicted a custom annotated thematic data given an article.

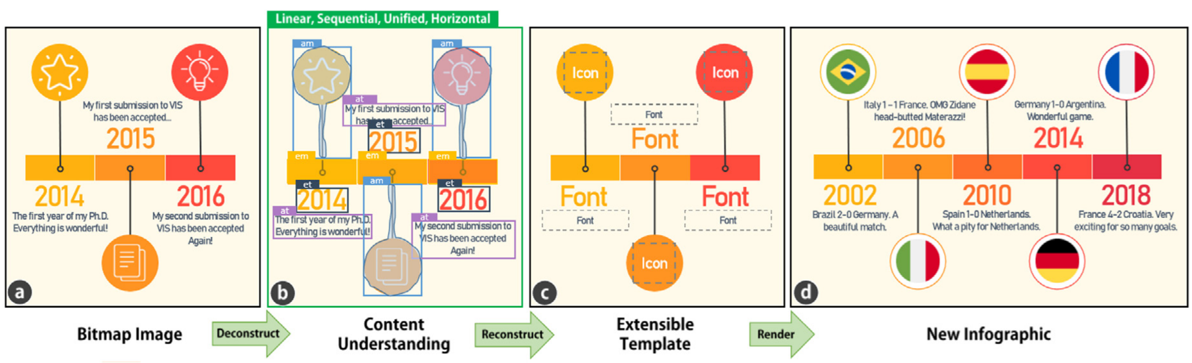

Fig. 11. An automatic approach (Chen et al., 2019) to extract an architecture of timeline template from a timeline infographic and extend the timeline visualizations with an updated dataset. (a) The input of an infographic about timeline, (b) Extraction of visual elements and text from the timeline infographic; (c) Identification and extraction of the architecture from the timeline infographic; (d) Generation of a new timeline infographic.

information for identifying the best variable for a thematic map. The second is annotation relevancy as captured by cosine similarity. The third is visual interestingness as captured by Moran's I. This ranking technique makes a choice of the visualizations to maximize the relevance of annotations and article. Fig. 10 shows an application of *NewsView*, which illustrates an article about the results of a nationwide evaluation of educational performance.

The timeline is another popular narrative method. Information can be expressively and intuitively expressed with a time sequence. Shown in Fig. 11, Chen et al. (2019) proposed an automatic approach of extracting an extensible timeline template from a bitmap image and generating a new timeline infographic with renewed data. The approach can be divided into two parts: deconstructing bitmap timeline infographics and reconstructing extensible template. To deconstruct bitmap timeline infographics, they extended ResNeXt (Xie et al., 2017) with Feature Pyramid Network (Lin et al., 2017). This parts is developed from the global and local perspectives respectively. Globally, they employed ResNeXt to extract features of image within 2048 channels, and used two fully connected layers as class headers to predict type and orientation of a image. Locally, they defined six type of elements and performed sliding windows (Lampert et al., 2008) to detect the detailed elements. To reconstruct extensible template, They eliminated repeated and fixed the failed identifications with non-maximum merging and redundancy recovery. They reused and refreshed the collected text and visual items with DL GrabCut and text recognition techniques. They incorporated about 4300 timeline images and collected real-world about 400 timeline pictures online. After training the model, they used cases to validate the effectiveness of this approach by automatically extracting the structure and content of timeline charts and create the timeline chart visualization.

Chen et al. (2020) extended the Vega grammar and proposed *PapARVis Designer*, which generates static and virtual charts to augment visualizations. By comparing the similarities and dissimilarities of dataflow between virtual design and augmented static charts, this tool automatically verifies the dataflow of visual design and provides hints for debugging invalid visual encodings.

After reviewing works of automatically creating storytelling visualizations, we find the following limitation and future directions:

- Limitation: It is difficult to select a proper machine learning model to train. Furthermore, it is the challenge of collecting comprehensive infographic datasets as the train data.
- Future work: Extending automated storytelling visualization design, such as geographic infographics generation. Interpretation for automated visualization of the model.

3.5. Summary

In this section, we review and discuss the work of automatically creating visualizations based on data-driven approaches. Most of these works use machine learning models to automatically generate infographics. Some studies need to pre-define design constraints based on the characteristics of data, rank visualizations based on quality calculation, and recommend visualization authors to make a choice of visualizations.

These approaches or tools can automatically generate and recommend users high-quality visualizations. Users do not need to understand the relevant knowledge of information visualization to effectively obtain visual design. Nevertheless, there are still quite a few research challenges that must be addressed. One challenge is involved with the establishment of a high-quality visual coding corpus. Although the existing tools (Zhang et al., 2020; Poco et al., 2017) have the ability to extract visual elements from the diagram, the wrong description of visual elements remains to be addressed. An additional challenge is interpretability issues of approaches based on machine learning. The interpretability of machine learning can be integrated with user-defined knowledge to further enhance the expression and completeness of automatically generated visualization, which are the most important conditions for visualization.

4. Knowledge-based visualizations

This section mainly introduces knowledge-based automated visualization design techniques, which use a series of user-defined constraints and visualization design constraints to guide aesthetic and expressive visualization recommendations (Bordegoni et al., 1997).

4.1. Traditional visualization recommendations

This section mainly introduces automatic tools on generating common visualization recommendations such as line chart and bar chart based on perceptual measurements.

Some experts proposed semi-automatic tools (such as *SAGE* (Roth and Mattis, 1990, 1991) and *Sagetools* (Roth and Mattis, 1991) or framework (Goldstein et al., 1994) to decrease the tedious operations. Microsoft Power BI's "Quick Insights" template, Tableau's "Explain Data" feature, and Google Sheets can also provide automatic designs of visualization.

Mackinlay et al. (2007) proposed *Show Me*, which has been incorporated automatic presentation into Tableau. *Show Me* can automatically generate small multiple views and user experience for automatic presentation functionality. Fig. 12 shows the alternative visualizations. *MashupAdvisor* (Elmeleegy et al., 2008) and behavior-driven visualization recommendation(*BDVR*) (Gotz and Wen, 2009) can automatically provide users recommendations while users requested an issue query. *BDVR* defined four types: scanning, flipping, swapping, and drilling-down, based on users' behaviors. The algorithm detects patterns and provides users the intended visualizations. Wills and Wilkinson (2010) presented *AutoVis*, which use statistical criteria over data dimensions and records in recommending and ranking visualizations.

More recently, researchers have proposed automatic design engines for more expressive and effective visualizations design (Luo et al., 2018; Wang et al., 2019b). *Voyager* (Wongsuphasawat et al., 2015) and *Voyager 2* (Wongsuphasawat et al., 2017) are the mixed-initiative systems. *Voyager* proposed series of considerations to guide the charts design of recommendations. *Voyager 2* then synthesized interactive and automated visualization specifications to assist analysts in overall exploration and focused analysis. Ke et al. (2013) proposed *SEEDB*, which has the ability to automatically identify and recommend to the analyst visualizations after users input a query. The query is conjunction with a variety of existing database systems, and the system supports analysts three types of mechanisms to query issues: SQL query for domain users, casual query that can be transformed into formal query with a query builder tool, and pre-defined query for analysts unfamiliar with SQL. Wongsuphasawat et al. (2016) then proposed *CompassQL*, which can be able to generate a set of visualization recommendations through a set of enumeration rules combined with approaches towards choosing, ranking, and classifying visualizations.

Wang et al. (2019b) presented *DataShot*, which generates fact sheets automatically from tabular data. Considering that a high-quality fact sheet needs to be thought-provoking, insightful, informative, easy-understand, and aesthetically pleasing. They identify a common qualitative analysis (such as infographic structures, presentation layout, fact types, and visualization styles) for fact sheet designs during the study. They proposed a fact sheet auto-generation workflow, which consists of extracting data fact, compositing data fact, and synthesizing presentation, based on a formative survey. For data extraction, it involved the computation of importance score (significance score, impact score, and context score) of each data fact. For data fact composition, the topic of a fact sheet is extracted and ranked, and a density-based top-n algorithm is defined to measure the dissimilarity between two data facts. For synthesizing presentation, it includes a fact-visual mapping model building, fact description generating, fact sheet layout, and fact sheet styling. Fig. 13 shows the generated examples by *DataShot*.

4.2. Automatic annotations

Section 3.2 has already described several applications of annotation. This section mainly introduces some applications of automated annotation based on user-defined rules,

Some experts focus on the application of automated annotations for error warning, such as the universally used *Microsoft Office Word* for automatically checking errors in the text. The designers create the analyze tree based on the grammar rules of various language types. Shown in Fig. 14(A), *Microsoft Office Word* uses red wavy lines to indicate spelling mistakes, uses green wavy lines to make a notation of grammatical errors, etc. This work greatly improves work efficiency. Inspired by the red wavy underline that indicates spelling mistakes, Hopkins et al. (2020) then introduced *VisuaLint* using a series of heuristics for five kinds of errors: dual-axis charts with various scales, perceptually invalid color encodings, missing legends, truncated axes, and inexpressive size encodings. Fig. 14(B) highlights three examples of error annotations.

Some work is designed for helping users to understand contents such as *Graphoto* (Park et al., 2018). Mittal et al. (1998) extend the automatic graphical presentation systems to create descriptive captions in natural language, to facilitate users to understand the information expressed in the graphic. *Click2Annotate* Chen et al. (2010a) and *Touch2Annotate* (Chen et al., 2010b) are semi-automatically annotations generator. *Click2Annotate* (see Fig. 15) is designed for an insight management solution, which consists of several parts: insight browsing and retrieval, insight network, insight sharing/exporting in collaborative visualization, and visualization recommendation and notification. Uses can obtain high-quality annotations with a "typing free" annotation approach for multi-touch system *Touch2Annotate*.

Based on previous work Kandogan (2012). Hullman et al. (2013a) presented *Contextifier*, which use information source to defined annotations as observational and additive. Layout algorithms for generating informed annotations were informed by

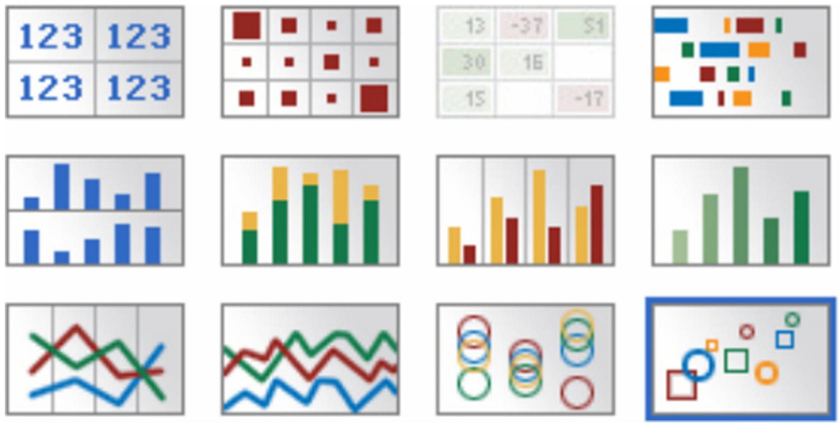

Fig. 12. The Ranking of alternative visualizations of *Show Me* (Mackinlay et al., 2007). Users can iteratively make a choice of visualization recommendations.

Fig. 13. Four examples of *DataShot*. (A) illustration of the events of shark attack happened in swimming activity; (B) an example of the sales status of sports cars; (C) an example of the sales status of manufacturer BMW; (D) illustration of the winning number of gold medals in the Summer Olympics from 1896 to 2012.

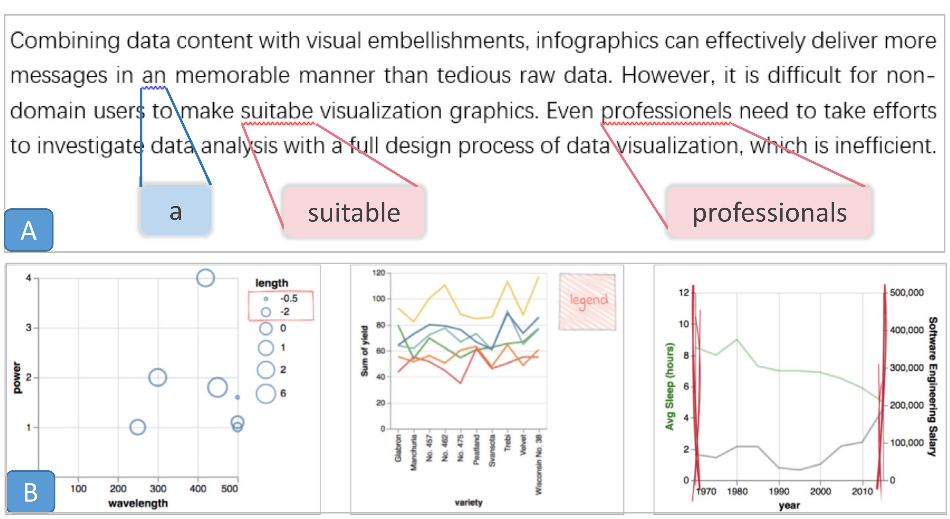

Fig. 14. (A) An example of using *Microsoft Office Word* for error notification. The red wavy lines indicate spelling mistakes while the green wavy lines make a notation of grammatical errors. (B) Three kind of examples in *VisuaLint* represented inexpressive size encodings (left), missing legends (medium), and dual-axis charts with various scales (right) respectively.

a study of professionally designed visualizations when it takes some related design rules into consideration. Considerations are visual salience, contextual connection, and identification of significant events in the company's history. *Contextifier* consists of

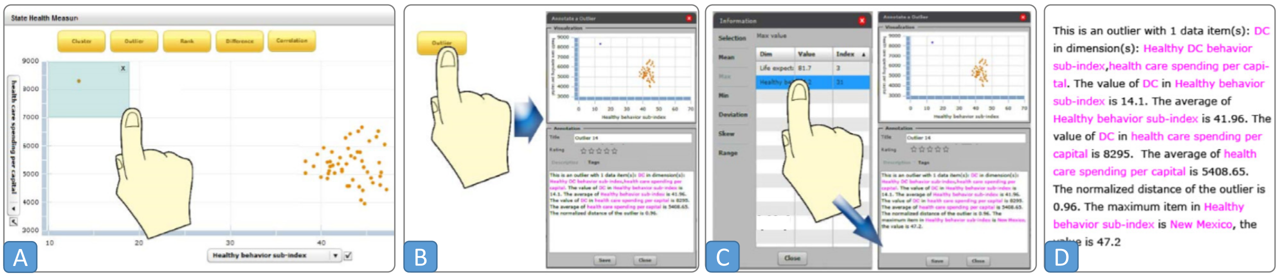

Fig. 15. A scenario where a user annotates an outlier using *Touch2Annotate*, which involves three steps: (A) selection of the outlines by drawing a rectangular boundary to; (B) generation of an annotation; (C) dragging of items into the annotation; (D) final annotation.

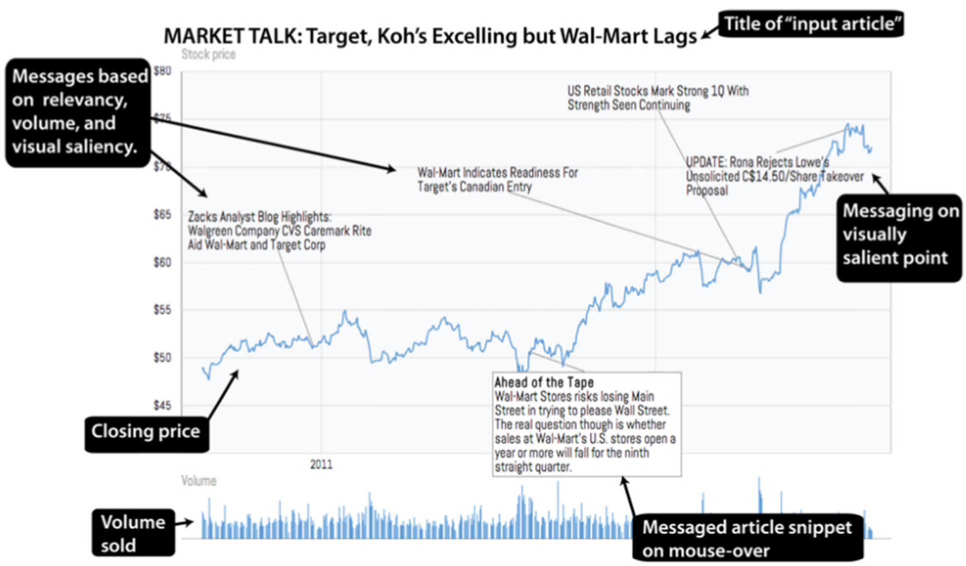

Fig. 16. An annotated visualization are generated by Contextifier (Hullman et al., 2013a).

four main components: a news article corpus, a query generator, an annotation selection engine (consisting of three feature generators and an integrator), and a graph generator. The workflow of the system starts from generating a query and matching it with the full-text index and the obtaining stock series. Then the feature generator calculates the feature from the text or stock sequence, and the feature integrator is used to integrate the obtained feature and use sorting to select notes, and finally use notes and stock sequences to generate line charts. Fig. 16 illustrates an example of annotation results. However, *Contextifier* is only suitable for the visualization of stock sequence data. Thus, the scalability problem of this work needs to be solved. Gao et al. (2014) then presented *NewsViews*, which can generate annotation content from a news corpus like *Contextifier* but can also support observational annotation of outliers (such as minimum value and maximum value). *NewsViews* is designed for working with any kind of articles and appropriate data of various types such as georeferenced data and time series. Kong et al. (2017) defined annotations as visual cues. They extended and studied the basic visual annotations, and categorized them as internal cues(including transparency, brightness, and magnification) and external cues (including colors). Besides, they performed a user study and find that external cues increase visual confusion rather than simplify images, while internal cues can weaken redundant information. Therefore, the effect of internal prompts is usually better than external prompts. Thus, future automated annotations can choose richer and more beautiful visual cues.

4.3. Automatic storytelling visualizations

This section focuses on knowledge-based storytelling visualization using automatic or semi-automatic techniques. It is common that *Google News Timeline* and *Google Finance* present automatically-generated visual storytelling of articles. These tools usually using simple icons, pictures and shot abstract for storytelling.

There are a lot of work for storytelling visualization by adding restrictions to automatically visualize specific types of data (Lu et al., 2020). Hullman et al. (2013b) proposed a graph-driven technique, which has the ability of automatically distinguishing the most effective sequences from a set of visualizations. This approach included defining data attributes for transition labeling and consistency maintaining. They employed an algorithm for identifying useful sequences to minimize local (visualization-to-visualization) changes of transitions. Attractive visualizations then were automatically sorted in the design process to help users make a structure decision in creating narrative visualizations. From a structural perspective, they implemented a qualitative analysis of forty-two visualization examples on explicitly-guided professional narrative. The result finally demonstrated that details narrative sequencing can be systematically approached in visualization systems.

Contextifier, which has been described in Section 4.2, automatically generates annotated visualizations to intuitively tell a story from a given article. Based on these previous work (Hullman et al., 2013a,b; Amini et al., 2015), Amini et al. (2016) presented a data-driven approach for semi-automatic narrative visualizations,

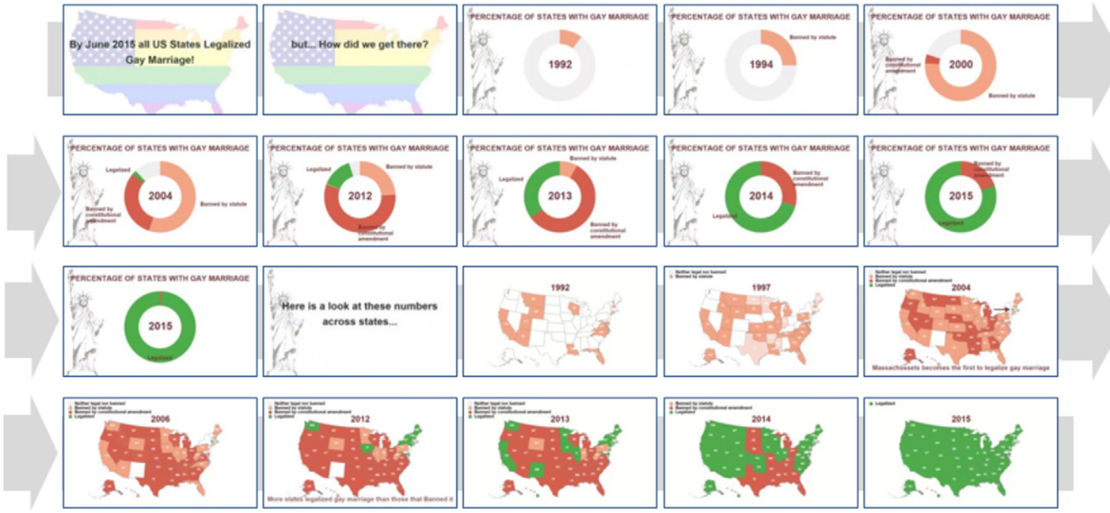

Fig. 17. A video example of same sex marriage via web-based tool *DataClips* (Amini et al., 2016).

called *DataClips*, an authoring tool are designed for lowering the barriers to generating data videos. Non-experts can use *DataClips* to generate story-telling videos such as an illustration in Fig. 17.

Some researchers have made some progress on animated transitions for photos storytelling. Based on the previous work on human recognition (Chellappa et al., 1995) and representative photos selection (Itti et al., 1998), Wen et al. (2012) developed an online comic composition interface, called *Pomics*, which can semi-automatically create continuous comic pages for further refinement after users take photographs of trips or social events as input. A similar idea was adopted to illustrate transitions of gaming behavior in automatic comics-based storytelling system Chu et al. (2014). Then, Bach et al. (2016) develop graph comics for storytelling and present a system, called *DynaVis*, to present and explain temporal changes in networks. They proposed several guidelines for the design of graph comics, the first step is collecting diagrams, related literature, and pictures within comics to extract architectures of typical comics and reconstruct a new architecture. the second step is finding visual encodings that could represent graph objects, their properties, and the possible variations which they may undergo. the third step are principle design for denying what kind of visual marks or attributes should be used. the next step is creating comics, discussing with domain experts, and reading study.

Meanwhile, there are also efforts for exploring in other type data visualization for automatic storytelling (Kim et al., 2017; Ding et al., 2019), and animated narrative visualization for video (Wang et al., 2016), and time-varying data visualization (Li et al., 2010).

4.4. Graph and network visualizations

There have been many years of research on the visualization of automatically generated network-graphs. From simple graph generation (Marks, 1990, 1991a,b; El-Said et al., 1997) to complex social media networks Chen and Neill (2014), researchers have been devoted a lot of time in layout algorithm optimization (Kosak et al., 1994; Henry and Fekete, 2006) and data visualization. This section introduces some classic automatic creation of graphs based on experts' knowledge.

GraphViz Ellson et al. (2001) and Gephi (Bastian et al., 2009) are the most common tools for drawing network-graph automatically, which incorporates a set of aesthetic criteria and apply layout algorithms for finding aesthetically pleasing network visualizations.

Early work by Kosak et al. (1994) described the rule-based approach for the layout of network diagrams. They generalized the challenges of network-diagram layout takes perceptual organization consideration, which can be categorized into three components: syntactic validity, perceptual organization, and aesthetic optimality. Thus, they discussed two algorithms for these challenges. One is leveraging heuristic layout rules to incrementally compute a layout, and another is finding the possible layout solutions. This layout algorithm plays a crucial role in the development of network-graph visualization. A lot of work is performed based on this work such as a popular book Di Battista et al. (1999).

There are also some researches Duncan et al. (2010) using curved edges to creating aesthetically-appealing, compact visualization. Fig. 18 shows the different types of edges used in (Xu et al., 2012). Fig. 18(B) aimed to automatically generate graph visualizations using Lombardi drawing.

4.5. Summary

This section reviewed and discussed the work to automatically generating visualizations based on knowledge-based approaches. Based on the experience summarized in the visual design process (e.g., efficiency and expressiveness), researchers defined rules and declare layout constraints to recommend visualizations. Most of the work is a semi-automated system, which integrates interaction and automatic visualization specifications. For example, *DataShot* requires users to choose the type of theme. Some researchers have summarized the user's behavior rules, detected the user's intentions, and recommended to the user visualization charts. Existing knowledge-based work is categorized into four classes, namely, statistical visualization generation, annotation generation, network-graph generation, and storytelling visualization generation. For these types of charts, notable advances have been achieved. The existing paper or tools are based on one or several layout rules, so the user-defined layout rules are not comprehensive. Other types of designs are remained to develop such as specialized map designs.

5. Hybrid visualizations

For data-driven systems/tools, designers use data-driven models (such as machine learning models) to predict user-intend visualizations. For knowledge-based systems/tools, designers use

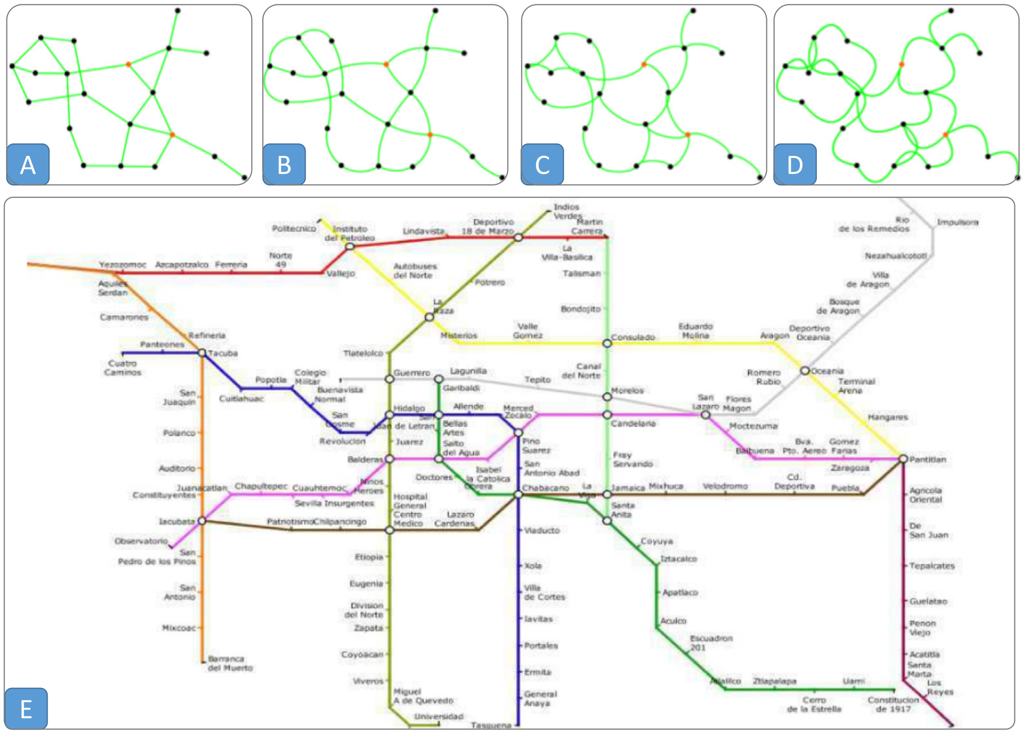

Fig. 18. (A–D) Examples of edge types used in the cured edge study. (A) Straight edge, (B) Lombardi layout, (C) Slightly curved edge, (D) Heavily curved edge. (E) Visualization of automatic metro map layout using multicriteria optimization.

perceptual knowledge about visualization designs to defined constraints for automatically generating visualizations. Hybrid systems/tools incorporate both data-driven models and user-defined constraints, which allow users to participate in the recommendation process and improve the knowledge-base constraints with machine learning. This section mainly introduced hybrid visualization design techniques (Luo et al., 2018).

Luo et al. (2018) presented *DeepEye*, which tackles three key challenges: visualization recognition, visualization quantifying, and visualization ranking. For the first challenge, they leveraged a decision tree to make a determination of good or bad for visualizations. For the second and third challenges, an existing machine learning Chris Burges et al. (2005) is performed, which is widely used in information retrieval, natural language processing, and data mining. Thus, they make use of technique Chris Burges et al. (2005) as black-box for training the model and normalized discounted cumulative gain Valizadegan et al. (2009) for ranking visual recommendations. Visualization examples are generated via *DeepEye* given race dataset (see Fig. 19).

Moritz et al. (2019) proposed *Draco*, which is a constraints-based interface based on Answer Set Programming. The design space of a visualization model in this work is denied by a set of empirical constraints. The space of possible visualization specifications is classified with two types. One is a suite of aggregate rules that species the domains of dimensions (mark, encoding, shape, or text). Another set of integrity constraints that defines how various dimensions can interact with each other. *Draco* employed RankSVM and a learning-to-rank techniques Retrieval (2010) to learn preference model, They used a learning algorithm that allows the model to learn soft constraint weights from ranked pairs of visualizations, and *Draco* used a simple linear model over soft constraint weights to learn how to make a balance among competing design rules.

Aiming at the proportion-related short text, Cui et al. (2019) proposed *Text-to-Vis* to automatically generate infographics, which is combined with visual elements. Firstly, they collected two hundreds distinct infographics and manually deconstructed about 1000 individual infographic units, which is categorized into four main groups: statistics-based, timeline-based, process-based, and location-based. Statistics-based infographics is about a half, of which proportion is the most common type. For reconstructing visual items, They take two parts into consideration: text space and visual space. For text space, they manually labeled visual elements. Then, the model with CNN and CRF is trained on the labeled dataset to identify and extract information. For visual space, they defined and constrained one infographic as several attributes: layout constraints, text description, visual graphic, and proper color. Then, they synthesized the proportion-related input statements to predict users' intention, and leveraged the visual elements in visual corpus to combine all possible visualizations. Finally, each combined result of infographics was then evaluated and ranked by combining three weighted scores, which include semantic score, visual score, and informative score. They applied *Word2Vec* for evaluating semantic score, employed a method of measure non-empty space for getting a visual score, implemented an evaluation on the completeness of message delivery for obtaining an informative score. This approach does not require users a time-consuming and tedious authoring processor design expertise. Infographic recommendations were automatically reconstructed and ranked after inputting a proportion-related data query.

6. Evaluations on infographics and visualizations

There are a lot of evaluations Harrison et al. (2015) on information visualizations and studies including surveys Grammel et al. (2013) and evaluations Satyanarayan et al. (2019); Bateman et al. (2010) and Wun et al. (2016) on interactive visualization creating tools (Bigelow et al., 2016; Liu et al., 2018) and evaluations. However, Most of them focused only on encoding channels and data types, their evaluation criteria remain limited. In the

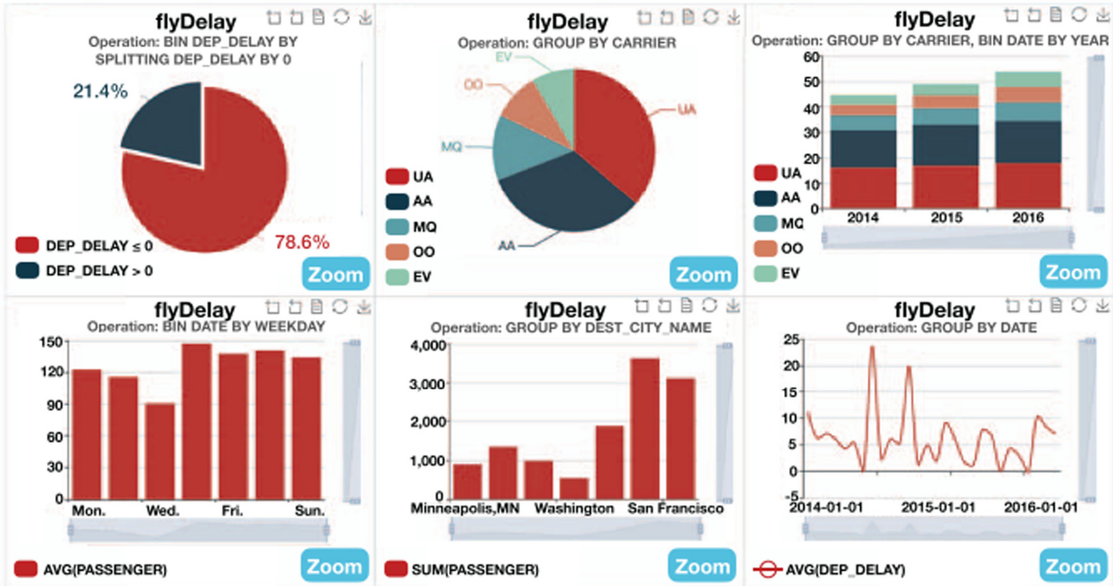

Fig. 19. Charts are automatically generated by *DeepEye*.

existing researches on automatic visualizations, it is a lack of work to fully evaluate the approaches of these tools.

After investigating the literature on two aspects including infographics designs (Xu et al., 2020; Bigelow et al., 2014; Saket et al., 2017) and interactive design tools (Pantazos and Lauesen, 2012; Ren et al., 2018b), we asked two questions: What are the requirements for high-quality information visualization? What evaluation indicators do automatic visualization generating tools need to satisfy? We summarize the some evaluation objectives (Saket et al., 2018; Méndez et al., 2017) and considerations as follows:

- **Diversity-supporting of input datasets**: A measurement is required on diversity-supporting involved with types (such as numeral data, textual data, images, and video), format(such as JSON, CSV, and hybrid format) and content(such as timeline, geographic, and network) of data.
- **The degree of automation during the process**: Qualitative or quantitative measurements are required to reveal how intelligent the techniques are in terms of the procedure of inputting datasets to outputting visualizations.
- **The quality of output visualizations**: Expressing information correctly is the most important condition for automatically generating a visualization. Meanwhile, other criteria are also required, such as comprehensivity (Bateman et al., 2010; Borkin et al., 2013), aesthetic (Harrison et al., 2015), and creativity-supporting (Borgo et al., 2012).
- **Learnability of models**: Automatically generating visualizations needs to reduce the user's learning input as much as possible (Chen et al., 2009).

On the one hand, visualization designs need to be creative, on the other hand, automated design tools are required to minimize interactions involving users. Thus, the work faces an important challenge. It requires to make a balance between creativity and user interaction.

7. Conclusion and future challenges

This state-of-the-art paper reviews existing research in the techniques of automatic visualizations. It introduces a comprehensive overview of many advances in automatically generating visualizations to gain a better understanding of the cutting-edge research in this field. This work is the first step towards reviewing automatic visualizations in a novel and systematic manner. Then, this report classifies existing work of automatic visualizations generating tools into data-driven and knowledge-based approaches. Additionally, through the analysis and comparison across related work, this report identifies the trends and recent developments in automatic visualizations. Furthermore, we divide the literature review into several broad application categories such as automatic storytelling generating, automatic annotation, and automatic network-graphs generating. Next, we discuss and summarize the key challenges and several future researches respectively.

Building comprehensive visual corpus. There is no complete and public corpus in existing work. Building a corpus involves the collection and description of visual elements. Most of the corpus establishment in the existing work is downloaded from the Internet and manually labeled. This work is heavy, tedious, and inefficient. The method of automatically extracting (Tang et al., 2017; Srinivasan et al., 2018) and describing the visual components in the graph may be wrong, especially the professional visual elements related to various field. Due to one visual element may expresses kinds of meanings in different field, this is a great challenge to collect and describe visual elements.

Specific visualizations for special kind of data. Specific tools for automatic visualization recommendations may be more creative and comprehensive. However, visualization involves kinds of data types, and existing automatic infographics are designed for specific types of digital data, such as proportion, timelines, etc. Automatic visualizations should be generalized to support more types of information or other kinds of data, such as geographic location data, large text data, video data, audio data, etc.

Building hybrid models of generating automatic visualizations. Building Models intends to a hybrid model, which incorporate knowledge-based rules and machine learning models for more solid architecture. Although *Draco* trade-off competing design rules using a simple linear model. It is a significant challenge to build one hybrid model, which involves with the interpretability of machine learning and layout reconstruction of visual embellishments. The interpretation of machine learning can strengthen layout algorithm, and layout rules of visual embellishments can be able to prompt machine learning model. In addition, automated visualization model needs to have better mistake tolerance for complex data, and appropriate feedback regulation is also a necessity. Combining interpretability of machine learning model and algorithmic decision-making models is a challenge for automatically generating information visualization.

Supporting more kinds of efficient visualizations and trading-off between creativity and automation. Supporting more types and creativity of visualizations is another challenge. Existing automatic visualizations are mostly typical visualizations such as bar chart, pie chart, line chart, etc. However, more complex, creative, and interactive visualizations need to be supported. Existing automatic visualizations are mostly static visual recommendations, and generating interactive visualizations can be able to help users acquire more information.

Quantifying automatic infographics tools. It is a great challenge to assess the quality of automatic infographic tools, it can be divided into three parts: a measurement of diversity-supporting of input datasets, the degree of automation of a tool during the process of generating visualizations, and the quality of output visualizations. Firstly, measurement is required on diversity-supporting involved with types, format, and content of data. Secondly, a qualitative or quantitative measurement is desired to reveal how the techniques became more and more automatic and intelligent. Thirdly, Although there are many works on evaluations for output visualizations, most of them are evaluated for a certain type of visualizations. Automatically visualization recommendation tools may generate multiple types of visualizations. Thus, it is a great challenge to define a synthesized algorithm for quantifying automatic visualization generating tools.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is partly supported by National Natural Science Foundation of China (No. 61972356, 61902350), Zhejiang Provincial Natural Science Foundation of China (NO.LY19F020026), Zhejiang Provincial Key Research and Development Program of China (No. 2019C01009), and Fundamental Research Funds for the Provincial Universities of Zhejiang (NO. RF-C2019001).

References

Amini, F., Henry Riche, N., Lee, B., Hurter, C., Irani, P., 2015. Understanding data videos: Looking at narrative visualization through the cinematography lens. In: Proceedings of Annual ACM Conference on Human Factors in Computing Systems, pp. 1459–1468.

Amini, F., Riche, N.H., Lee, B., Monroy-Hernandez, A., Irani, P., 2016. Authoring data-driven videos with dataclips. IEEE Trans. Vis. Comput. Graphics 23 (1), 501–510.

Bach, B., Kerracher, N., Hall, K.W., Carpendale, S., Kennedy, J., Henry Riche, N., 2016. Telling stories about dynamic networks with graph comics. In: Proceedings of CHI Conference on Human Factors in Computing Systems, pp. 3670–3682.

Bastian, M., Heymann, S., Jacomy, M., 2009. Gephi: an open source software for exploring and manipulating networks. In: Third International AAAI Conference on Weblogs and Social Media.

Bateman, S., Mandryk, R.L., Gutwin, C., Genest, A., McDine, D., Brooks, C., 2010. Useful junk? The effects of visual embellishment on comprehension and memorability of charts. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 2573–2582.

Battle, L., Duan, P., Miranda, Z., Mukusheva, D., Chang, R., Stonebraker, M., 2018. Beagle: Automated extraction and interpretation of visualizations from the web. In: Proceedings of CHI Conference on Human Factors in Computing Systems, pp. 1–8.

Bigelow, A., Drucker, S., Fisher, D., Meyer, M., 2014. Reflections on how designers design with data. In: Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces, pp. 17–24.

Bigelow, A., Drucker, S., Fisher, D., Meyer, M., 2016. Iterating between tools to create and edit visualizations. IEEE Trans. Vis. Comput. Graphics 23 (1), 481–490.

Bordegoni, M., Faconti, G., Feiner, S., Maybury, M.T., Rist, T., Ruggieri, S., Trahanias, P., Wilson, M., 1997. A standard reference model for intelligent multimedia presentation systems. Comput. Stand. Interfaces 18 (6–7), 477–496.

Borgo, R., Abdul-Rahman, A., Mohamed, F., Grant, P.W., Reppa, I., Floridi, L., Chen, M., 2012. An empirical study on using visual embellishments in visualization. IEEE Trans. Vis. Comput. Graphics 18 (12), 2759–2768.

Borkin, M.A., Vo, A.A., Bylinskii, Z., Isola, P., Sunkavalli, S., Oliva, A., Pfister, H., 2013. What makes a visualization memorable?. IEEE Trans. Vis. Comput. Graphics 19 (12), 2306–2315.

Bostock, M., Ogievetsky, V., Heer, J., 2011. D^3 data-driven documents. IEEE Trans. Vis. Comput. Graphics 17 (12), 2301–2309.

Bryan, C., Ma, K.-L., Woodring, J., 2016. Temporal summary images: An approach to narrative visualization via interactive annotation generation and placement. IEEE Trans. Vis. Comput. Graphics 23 (1), 511–520.

Bylinskii, Z., Alsheikh, S., Madan, S., Recasens, A., Zhong, K., Pfister, H., Durand, F., Oliva, A., 2017. Understanding infographics through textual and visual tag prediction. arXiv:1709.09215.

Casner, S.M., 1991. Task-analytic approach to the automated design of graphic presentations. ACM Trans. Graph. 10 (2), 111–151.

Chellappa, R., Wilson, C.L., Sirohey, S., 1995. Human and machine recognition of faces: A survey. Proc. IEEE 83 (5), 705–741.

Chen, Y., Barlowe, S., Yang, J., 2010a. Click2annotate: Automated insight externalization with rich semantics. In: IEEE Symposium on Visual Analytics Science and Technology, pp. 155–162.

Chen, F., Neill, D.B., 2014. Non-parametric scan statistics for event detection and forecasting in heterogeneous social media graphs. In: Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1166–1175.

Chen, Z., Tong, W.S., Wang, Q.-W., Bach, B., Qu, H., 2020. Augmenting static visualizations with paparvis designer. In: Proceedings of CHI Conference on Human Factors in Computing Systems.

Chen, Z., Wang, Y., Wang, Q., Wang, Y., Qu, H., 2019. Towards automated infographic design: Deep learning-based auto-extraction of extensible timeline. IEEE Trans. Vis. Comput. Graphics 26 (1), 917–926.

Chen, Y., Yang, J., Barlowe, S., Jeong, D.H., 2010b. Touch2Annotate: Generating better annotations with less human effort on multi-touch interfaces. In: CHI'10 Extended Abstracts on Human Factors in Computing Systems, pp. 703–3708.

Chen, Y., Yang, J., Ribarsky, W., 2009. Toward effective insight management in visual analytics systems. In: IEEE Pacific Visualization Symposium, pp. 49–56.

Chris Burges, T.S., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., Hullender, G., 2005. Learning to rank using gradient descent. In: ICML 05.

Chu, W.-T., Yu, C.-H., Wang, H.-H., 2014. Optimized comics-based storytelling for temporal image sequences. IEEE Trans. Multimed. 17 (2), 201–215.

Cui, W., Zhang, X., Wang, Y., Huang, H., Chen, B., Fang, L., Zhang, H., Lou, J.-G., Zhang, D., 2019. Text-to-viz: Automatic generation of infographics from proportion-related natural language statements. IEEE Trans. Vis. Comput. Graphics 26 (1), 906–916.

Di Battista, G., Eades, P., Tamassia, R., Tollis, I.G., 1999. Graph Drawing, vol. 357. Prentice Hall, Upper Saddle River, NJ.

Dibia, V., Demiralp, Ç., 2019. Data2vis: Automatic generation of data visualizations using sequence-to-sequence recurrent neural networks. IEEE Comput. Graph. Appl. 39 (5), 33–46.

Ding, R., Han, S., Xu, Y., Zhang, H., Zhang, D., 2019. Quickinsights: Quick and automatic discovery of insights from multi-dimensional data. In: Proceedings of the 2019 International Conference on Management of Data, pp. 317–332.

dos Santos Vieira, R., do Nascimento, H.A.D., da Silva, W.B., 2015. The application of machine learning to problems in graph drawing a literature review. In: Proc. International Conference on Information, Process, and Knowledge Management, pp. 112–118.

Duncan, C.A., Eppstein, D., Goodrich, M.T., Kobourov, S.G., Nöllenburg, M., 2010. Lombardi drawings of graphs. In: International Symposium on Graph Drawing, pp. 195–207.

El-Said, M., Fischer, G., Gamalel-Din, S., Zaki, M., 1997. ADDI: A tool for automating the design of visual interfaces. Comput. Graph. 21 (1), 79–87.

Ellson, J., Gansner, E., Koutsofios, L., North, S.C., Woodhull, G., 2001. Graphviz—open source graph drawing tools. In: International Symposium on Graph Drawing, pp. 483–484.

Elmeleegy, H., Ivan, A., Akkiraju, R., Goodwin, R., 2008. Mashup advisor: A recommendation tool for mashup development. In: IEEE International Conference on Web Services, pp. 337–344.
Frick, A., Ludwig, A., Mehldau, H., 1994. A fast adaptive layout algorithm for undirected graphs (extended abstract and system demonstration). In: International Symposium on Graph Drawing, pp. 388–403.
Gansner, E.R., Koren, Y., North, S., 2004. Graph drawing by stress majorization. In: International Symposium on Graph Drawing, pp. 39–250.
Gao, T., Dontcheva, M., Adar, E., Liu, Z., Karahalios, K.G., 2015. Datatone: Managing ambiguity in natural language interfaces for data visualization. In: Proceedings of Annual ACM Symposium on User Interface Software and Technology, pp. 489–500.
Gao, T., Hullman, J.R., Adar, E., Hecht, B., Diakopoulos, N., 2014. NewsViews: an automated pipeline for creating custom geovisualizations for news. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 3005–3014.
García, R.L., Guzmán, J.G., Seco, A.D.A., 2007. iCharts: Charts for software process improvement value management. In: Software Process Improvement, European Conference, Eurospi, Potsdam, Germany.
Goldstein, J., Roth, S.F., Kolojejchick, J., Mattis, J., 1994. A framework for knowledge-based interactive data exploration. J. Vis. Lang. Comput. 5 (4), 339–363.
Gotz, D., Wen, Z., 2009. Behavior-driven visualization recommendation. In: Proceedings of International Conference on Intelligent User Interfaces, pp. 315–324.
Grammarly, 2012. Grammarly.
Grammel, L., Bennett, C., Tory, M., Storey, M.-A.D., 2013. A survey of visualization construction user interfaces. In: EuroVis (Short Papers).
Hamilton, W.L., Ying, R., Leskovec, J., 2017. Representation learning on graphs: Methods and applications. arXiv:1709.05584.
Harrison, L., Reinecke, K., Chang, R., 2015. Infographic aesthetics: Designing for the first impression. In: Proceedings of Annual ACM Conference on Human Factors in Computing Systems, pp. 1187–1190.
Henry, N., Fekete, J.-D., 2006. Matrixexplorer: a dual-representation system to explore social networks. IEEE Trans. Vis. Comput. Graphics 12 (5), 677–684.
Honghui, M., Wei, C., Yuxin, M., Huihua, G., Wanqi, H., 2018. VisComposer: A visual programmable composition environment for information visualization. Vis. Inf.
Hopkins, A.K., Correll, M., Satyanarayan, A., CSAIL, M., 2020. VisuaLint: Sketchy in situ annotations of chart construction errors. In: Proceedings of Eurographics Conference on Visualization.
Hu, K., Bakker, M.A., Li, S., Kraska, T., Hidalgo, C., 2019. Vizml: A machine learning approach to visualization recommendation. In: Proceedings of CHI Conference on Human Factors in Computing Systems, pp. 1–12.
Hullman, J., Diakopoulos, N., Adar, E., 2013. Contextifier: automatic generation of annotated stock visualizations. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 2707–2716.
Hullman, J., Drucker, S., Riche, N.H., Lee, B., Fisher, D., Adar, E., 2013b. A deeper understanding of sequence in narrative visualization. IEEE Trans. Vis. Comput. Graphics 19 (12), 2406–2415.
Itti, L., Koch, C., Niebur, E., 1998. A model of saliency-based visual attention for rapid scene analysis. IEEE Trans. Pattern Anal. Mach. Intell. 20 (11), 1254–1259.
Jacomy, M., Venturini, T., Heymann, S., Bastian, M., 2014. Forceatlas2, a continuous graph layout algorithm for handy network visualization designed for the gephi software. PLoS One 9 (6).
Júnior, A.S., Renso, C., Matwin, S., 2017. Analytic: An active learning system for trajectory classification. IEEE Comput. Graph. Appl. 37 (5), 28–39.
Kandogan, E., 2012. Just-in-time annotation of clusters, outliers, and trends in point-based data visualizations. In: IEEE Conference on Visual Analytics Science, Technology (VAST), pp. 73–82.
Ke, M.-T., Fujimoto, S., Imai, T., 2013. Seedb: a simple and morphology-preserving optical clearing agent for neuronal circuit reconstruction. Nature Neurosci. 16 (8), 1154.
Key, A., Howe, B., Perry, D., Aragon, C., 2012. Vizdeck: self-organizing dashboards for visual analytics. In: Proceedings of ACM SIGMOD International Conference on Management of Data, pp. 681–684.
Kim, Y., Heer, J., 2018. Assessing effects of task and data distribution on the effectiveness of visual encodings. Comput. Graph. Forum 37 (3), 157–167.
Kim, Y., Wongsuphasawat, K., Hullman, J., Heer, J., 2017. Graphscape: A model for automated reasoning about visualization similarity and sequencing. In: Proceedings of CHI Conference on Human Factors in Computing Systems, pp. 2628–2638.
Kipf, T.N., Welling, M., 2016. Variational graph auto-encoders. arXiv:1611.07308.
Kong, N., Agrawala, M., 2012. Graphical overlays: Using layered elements to aid chart reading. IEEE Trans. Vis. Comput. Graphics 18 (12), 2631–2638.
Kong, H.-K., Liu, Z., Karahalios, K., 2017. Internal and external visual cue preferences for visualizations in presentations. In: Computer Graphics Forum, vol. 36. (3), pp. 515–525.
Kosak, C., Marks, J., Shieber, S., 1994. Automating the layout of network diagrams with specified visual organization. IEEE Trans. Syst. Man Cybern. 24 (3), 440–454.
Lai, C., Lin, Z., Jiang, R., Han, Y., Liu, C., Yuan, X., 2020. Automatic annotation synchronizing with textual description for visualization. In: Proceedings of CHI Conference on Human Factors in Computing Systems, pp. 1–13.
Lampert, C.H., Blaschko, M.B., Hofmann, T., 2008. Beyond sliding windows: Object localization by efficient subwindow search. In: 2008 IEEE Conference on Computer Vision, Pattern Recognition. IEEE, pp. 1–8.
Li, Y., Lu, A., Ribarsky, W., Wei, C., 2010. Automatic animation for time-varying data visualization. Comput. Graph. Forum 29 (7), 2271–2280.
Li, D., Mei, H., Shen, Y., Su, S., Zhang, W., Wang, J., Ming, Z., Chen, W., 2018. Echarts: A declarative framework for rapid construction of web-based visualization. Vis. Inf. 2, 136–146.
Lin, T., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2017. Feature pyramid networks for object detection. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 936–944.
Liu, Z., Thompson, J., Wilson, A., Dontcheva, M., Delorey, J., Grigg, S., Kerr, B., Stasko, J., 2018. Data Illustrator: Augmenting vector design tools with lazy data binding for expressive visualization authoring. In: Proceedings of CHI Conference on Human Factors in Computing Systems, pp. 1–13.
Liu, C., Xie, L., Han, Y., Wei, D., Yuan, X., 2020. AutoCaption: An approach to generate natural language description from visualization automatically. In: 2020 IEEE Pacific Visualization Symposium, pp. 91–195.
Lu, J., Wang, J., Ye, H., Gu, Y., Chen, W., 2020. Illustrating changes in time-series data with data video. IEEE Comput. Graph. Appl. PP (99), 1–1.
Luo, Y., Qin, X., Tang, N., Li, G., 2018. DeepEye: towards automatic data visualization. In: Proceedings of IEEE International Conference on Data Engineering (ICDE), pp. 101–112.
Mackinlay, J., 1986. Automating the design of graphical presentations of relational information. ACM Trans. Graph. 5 (2), 110–141.
Mackinlay, J., Hanrahan, P., Stolte, C., 2007. Show me: Automatic presentation for visual analysis. IEEE Trans. Vis. Comput. Graphics 13 (6), 1137–1144.
Marks, J., 1990. A syntax and semantics for network diagrams. In: Proceedings of IEEE Workshop on Visual Languages, pp. 104–110.
Marks, J.W., 1991a. Automating the Design of Network Diagrams. Harvard University.
Marks, J., 1991b. A formal specification scheme for network diagrams that facilitates automated design. J. Vis. Lang. Comput. 2 (4), 395–414.
Mauri, M., Elli, T., Caviglia, G., Uboldi, G., Azzi, M., 2017. RAWGraphs: a visualisation platform to create open outputs. In: Proceedings of the 12th Biannual Conference on Italian SIGCHI Chapter, pp. 1–5.
Mei, H., Chen, W., Ma, Y., Guan, H., Hu, W., 2018a. Viscomposer: A visual programmable composition environment for information visualization. Vis. Inf. 2 (1), 71–81.
Mei, H., Ma, Y., Wei, Y., Chen, W., 2018b. The design space of construction tools for information visualization: A survey. J. Vis. Lang. Comput. 44, 120–132.
Méndez, G.G., Hinrichs, U., Nacenta, M.A., 2017. Bottom-up vs. top-down: tradeoffs in efficiency, understanding, freedom and creativity with InfoVis tools. In: Proceedings of CHI Conference on Human Factors in Computing Systems, pp. 841–852.
Mittal, V.O., Carenini, G., Moore, J.D., Roth, S., 1998. Describing complex charts in natural language: A caption generation system. Comput. Linguist. 24 (3), 431–467.
Moldovan, D., Decker, J.M., Wang, F., Johnson, r.A., Lee, B.K., Nado, Z., Sculley, D., Rompf, T., Wiltschko, A.B., 2018. Autograph: Imperative-style coding with graph-based performance. arXiv:1810.08061.
Moritz, D., Wang, C., Nelson, G.L., Lin, H., Smith, A.M., Howe, B., Heer, J., 2019. Formalizing visualization design knowledge as constraints: Actionable and extensible models in draco. IEEE Trans. Vis. Comput. Graphics 25 (1), 438–448.
Ou, M., Cui, P., Pei, J., Zhang, Z., Zhu, W., 2016. Asymmetric transitivity preserving graph embedding. In: Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1105–1114.
Pantazos, K., Lauesen, S., 2012. Constructing visualizations with infovis tools-an evaluation from a user perspective. GRAPP/IVAPP 2012, 731–736.
Park, J.H., Kaufman, A., Mueller, K., 2018. Graphoto: Aesthetically pleasing charts for casual information visualization. IEEE Comput. Graph. Appl. 38 (6), 67–82.
Pioch, N.J., Everett, J.O., 2006. POLESTAR: collaborative knowledge management and sensemaking tools for intelligence analysts. In: Proceedings of ACM International Conference on Information and Knowledge Management, pp. 513–521.
Poco, J., Mayhua, A., Heer, J., 2017. Extracting and retargeting color mappings from bitmap images of visualizations. IEEE Trans. Vis. Comput. Graphics 24 (1), 637–646.
Ren, D., Höllerer, T., Yuan, X., 2014. iVisDesigner: Expressive interactive design of information visualizations. IEEE Trans. Vis. Comput. Graphics 20 (12), 2092–2101.
Ren, D., Lee, B., Brehmer, M., 2018a. Charticulator: Interactive construction of bespoke chart layouts. IEEE Trans. Vis. Comput. Graphics 25 (1), 789–799.

Ren, D., Lee, B., Brehmer, M., Riche, N.H., 2018b. Reflecting on the evaluation of visualization authoring systems: Position paper. In: 2018 IEEE Evaluation and beyond-Methodological Approaches for Visualization, pp. 86–92.

Retrieval, F.T.I.I., 2010. Learning to rank for information retrieval (LR4IR 2007). In: International ACM Sigir Conference on Research and Development in Information Retrieval.

Roth, S.F., Kolojejchick, J., Mattis, J., Chuah, M.C., 1995. Sagetools: An intelligent environment for sketching, browsing, and customizing data-graphics. In: Conference Companion on Human Factors in Computing Systems, pp. 409–410.

Roth, S.F., Mattis, J., 1990. Data characterization for intelligent graphics presentation. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 193–200.

Roth, S.F., Mattis, J., 1991. Automating the presentation of information. In: Proceedings of the IEEE Conference on Artificial Intelligence Applications, pp. 90–97.

Saket, B., Endert, A., Demiralp, Ç., 2018. Task-based effectiveness of basic visualizations. IEEE Trans. Vis. Comput. Graphics 25 (7), 2505–2512.

Saket, B., Srinivasan, A., Ragan, E.D., Endert, A., 2017. Evaluating interactive graphical encodings for data visualization. IEEE Trans. Vis. Comput. Graphics 24 (3), 1316–1330.

Satyanarayan, A., Heer, J., 2014. Authoring narrative visualizations with ellipsis. In: Computer Graphics Forum, vol. 33. pp. 361–370.

Satyanarayan, A., Lee, B., Ren, D., Heer, J., Stasko, J., Thompson, J., Brehmer, M., Liu, Z., 2019. Critical reflections on visualization authoring systems. IEEE Trans. Vis. Comput. Graphics 26 (1), 461–471.

Satyanarayan, A., Moritz, D., Wongsuphasawat, K., Heer, J., 2016. Vega-lite: A grammar of interactive graphics. IEEE Trans. Vis. Comput. Graphics 23 (1), 341–350.

Srinivasan, A., Drucker, S.M., Endert, A., Stasko, J., 2018. Augmenting visualizations with interactive data facts to facilitate interpretation and communication. IEEE Trans. Vis. Comput. Graphics 25 (1), 672–681.

Sun, G.-D., Wu, Y.-C., Liang, R.-H., Liu, S.-X., 2013. A survey of visual analytics techniques and applications: state-of-the-art research and future challenges. J. Comput. Sci. Technol. 28 (5), 852–867.

Tang, B., Han, S., Yiu, M.L., Ding, R., Zhang, D., 2017. Extracting top-k insights from multi-dimensional data. In: Proceedings of ACM International Conference on Management of Data, pp. 1509–1524.

Valizadegan, H., Rong, J., Zhang, R., Mao, J., 2009. Learning to rank by optimizing NDCG measure. In: Advances in Neural Information Processing Systems 22: Conference on Neural Information Processing Systems a Meeting Held December.

Wang, Y., Chen, Z., Li, Q., Ma, X., Luo, Q., Qu, H., 2016. Animated narrative visualization for video clickstream data. In: SIGGRAPH Asia Symposium on Visualization, pp. 1–8.

Wang, Y., Jin, Z., Wang, Q., Cui, W., Ma, T., Qu, H., 2019a. Deepdrawing: A deep learning approach to graph drawing. IEEE Trans. Vis. Comput. Graphics 26 (1), 676–686.

Wang, Y., Sun, Z., Zhang, H., Cui, W., Xu, K., Ma, X., Zhang, D., 2019b. DataShot: Automatic generation of fact sheets from tabular data. IEEE Trans. Vis. Comput. Graphics 26 (1), 895–905.

Wen, M.-H., Thawonmas, R., Chen, K.-T., 2012. Pomics: A computer-aided storytelling system with automatic picture-to-comics composition. In: 2012 Conference on Technologies and Applications of Artificial Intelligence, pp. 314–318.

Wills, G., Wilkinson, L., 2010. Autovis: automatic visualization. Inf. Vis 9 (1), 47–69.

Wongsuphasawat, K., Moritz, D., Anand, A., Mackinlay, J., Howe, B., Heer, J., 2015. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. IEEE Trans. Vis. Comput. Graphics 22 (1), 649–658.

Wongsuphasawat, K., Moritz, D., Anand, A., Mackinlay, J., Howe, B., Heer, J., 2016. Towards a general-purpose query language for visualization recommendation. In: Proceedings of the Workshop on Human-in-the-Loop Data Analytics, pp. 1–6.

Wongsuphasawat, K., Qu, Z., Moritz, D., Chang, R., Ouk, F., Anand, A., Mackinlay, J., Howe, B., Heer, J., 2017. Voyager 2: Augmenting visual analysis with partial view specifications. In: Proceedings of CHI Conference on Human Factors in Computing Systems, pp. 2648–2659.

Wun, T., Payne, J., Huron, S., Carpendale, S., 2016. Comparing bar chart authoring with microsoft excel and tangible tiles. In: Computer Graphics Forum, vol. 35. pp. 111–120.

Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K., 2017. Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1492–1500.

Xu, K., Ottley, A., Walchshofer, C., Streit, M., Chang, R., Wenskovitch, J., 2020. Survey on the analysis of user interactions and visualization provenance.

Xu, K., Rooney, C., Passmore, P., Ham, D.-H., Nguyen, P.H., 2012. A user study on curved edges in graph visualization. IEEE Trans. Vis. Comput. Graphics 18 (12), 2449–2456.

You, J., Ying, R., Ren, X., Hamilton, W.L., Leskovec, J., 2018. Graphrnn: Generating realistic graphs with deep auto-regressive models. arXiv:1802.08773.

Zhang, J.E., Sultanum, N., Bezerianos, A., Chevalier, Fanny, DataQuilt: Extracting visual elements from images to craft pictorial visualizations. In: Proceedings of CHI Conference on Human Factors in Computing Systems, pp. 1–13.

Zhe, C., Badam, S.K., Yalin, A., Elmqvist, N., 2018. Datasite: Proactive visual data exploration with computation of insight-based recommendations. Inf. Vis. (1).