Towards Natural Language Interfaces for Data Visualization: A Survey

Leixian Shen, Enya Shen, Yuyu Luo, Xiaocong Yang, Xuming Hu, Xiongshuai Zhang, Zhiwei Tai, and Jianmin Wang

Abstract—Utilizing Visualization-oriented Natural Language Interfaces (V-NLI) as a complementary input modality to direct manipulation for visual analytics can provide an engaging user experience. It enables users to focus on their tasks rather than having to worry about how to operate visualization tools on the interface. In the past two decades, leveraging advanced natural language processing technologies, numerous V-NLI systems have been developed in academic research and commercial software, especially in recent years. In this article, we conduct a comprehensive review of the existing V-NLIs. In order to classify each article, we develop categorical dimensions based on a classic information visualization pipeline with the extension of a V-NLI layer. The following seven stages are used: query interpretation, data transformation, visual mapping, view transformation, human interaction, dialogue management, and presentation. Finally, we also shed light on several promising directions for future work in the V-NLI community.

Index Terms—Data visualization, natural language interfaces, survey

✦

1 INTRODUCTION

THE use of interactive visualization is becoming increasingly popular in data analytics [17]. As a common part of analytics suites, Windows, Icons, Menus, and Pointer (WIMP) interfaces have been widely employed to facilitate interactive visual analysis in current practices. However, this interaction paradigm presents a steep learning curve in visualization tools since it requires users to translate their analysis intents into tool-specific operations [126], as shown in the upper part of Fig. 1.

Over the years, the rapid development of Natural Language Processing (NLP) technology has provided great opportunities to explore a natural-language-based interaction paradigm for data visualization [18], [271]. With the help of advanced NLP toolkits [1], [3], [21], [81], [157], a surge of Visualization-oriented Natural Language Interfaces (V-NLI) emerged recently as a complementary input modality to traditional WIMP interaction. V-NLIs accept the user's natural language queries (e.g., "visualize the distribution of budget as a histogram") as input and output appropriate visualizations (e.g., correspondingly, a histogram with the *production budget*, a data attribute of cars dataset). The emergence of V-NLI can greatly enhance the usability of visualization tools in terms of: (a) Convenience and novice-

friendliness. Natural language is a skill that is mastered by the public. By using natural language to interact with computers, V-NLI closes the tool-specific manipulations to users, as shown in Fig. 1, facilitating the analysis flow for novices. (b) Intuitiveness and effectiveness. It is a consensus that visual analysis is most effective when users can focus on their data rather than manipulations on the interface of analysis tools [83]. With the help of V-NLI, users can express their analytic tasks in their terms. (c) Humanistic care. A sizeable amount of information we access nowadays is supported by visual means. V-NLI can be an innovative means for non-visual access, which promotes the inclusion of blind and low vision people.

However, designing and implementing V-NLIs is challenging due to the ambiguous and underspecified nature of human language, the complexity of maintaining context in a conversational flow, and the lack of discoverability (communicating to users what the system can do). In the past two decades, to address the challenges, numerous systems have been developed in academic research and commercial software, especially in recent years. The timeline of V-NLI is shown in Fig. 2. Back in 2001, Cox *et al.* [40] presented an initial prototype of NLI for visualization, which can only accept well-structured queries. Articulate [239] introduced a two-step process to create visualizations from NL queries almost a decade later. It first extracts the user's analytic task and data attributes and then automatically determines the appropriate visualizations based on the information. Although the infancy-stage investigations were a promising start, as natural language was not yet a prevalent interaction modality, the V-NLI systems were restricted to simple prototypes. However, since Apple integrated Siri [219] into the iPhone, NLIs began to attract more attention. Around 2013, the advent of word embeddings [163] promoted the advances of neural networks for NLP, rekindling commercial interest in V-NLI. IBM firstly published their NL-based cognitive service, Watson Analytics [4],
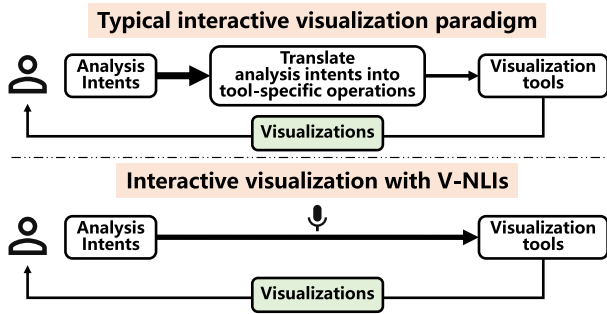
Fig. 1. The traditional interaction paradigm requires users to translate their analysis intents into tool-specific operations [126]. With the help of V-NLI, users can express their analysis intents in their terms.

in 2014. Microsoft Power BI's Q&A [5] and Tableau's Ask data [2] were announced in 2018 and 2019, respectively, offering various features like autocompletion and context management. DataTone [63] first introduced ambiguity widgets, which are user interface widgets (e.g., dropdown and slider) to help users modify system-generated responses. As opposed to one-off commands, Eviza [205] enabled users to have interactive conversations with their data. After technological accumulation, the past five years have seen the outbreak of V-NLI (see the number of yearly published papers in Fig. 2). With the development of hardware devices, synergistic multimodal visualization interfaces gained notable interest. Orko [231] was the first system to combine touch and speech input on tablet devices, and Data@Hand [115] focused on smartphones. Sneak pique [206] explored how autocompletion can improve system discoverability while helping users formulate analytical questions. The pretrained language models obtained new state-of-the-art results on various NLP tasks from 2018, which provided great opportunities to improve the intelligence of V-NLI [49], [179]. ADVISor [143] and ncNet [150] were follow-up deep learning-based solutions. Quda [61] and NLV [228] contributed datasets of NL queries for visual data analytics, and nvBench produced the first V-NLI benchmark [151]. Beyond data exploration, FlowSense [273] augmented a data flow-based visualization system with V-NLI. The NL4DV [174] toolkit can be easily integrated into existing visualization systems to provide V-NLI service.

Literature on V-NLI research is proliferating, covering aspects such as Visualization (VIS), Human-Computer Interaction (HCI), Natural Language Processing (NLP), and Data Mining and Management (DMM). As a result, there is an increasing need to better organize the research landscape, categorize current work, identify knowledge gaps, and assist people who are new to this growing area to understand the challenges and subtleties in the community. For this purpose, there have been several prior efforts to summarize the advances in this area. For example, Srinivasan and Stasko (Short paper in EuroVis 2017 [230]) conducted a simple examination of five existing V-NLI

systems by comparing and contrasting them based on the tasks they allow users to perform. They (Journal paper in CGA 2020 [232]) further highlighted critical challenges for evaluating V-NLIs and discussed benefits and considerations of three task framing strategies. Although the two surveys can provide valuable guidance for follow-up research, with the outbreak of V-NLI in recent years, considerable new works are to be covered and details to be discussed. To the best of our knowledge, this paper is the first step towards comprehensively reviewing V-NLI approaches in a systematic manner.

This paper is structured as follows: First, we explain the scope and methodology of the survey in Section 2. What follows is the classification overview of existing works in Section 3. Then, the comprehensive survey is presented in Sections 4, 5, 6, 7, 8, 9, and 10, corresponding to seven stages extracted in the information visualization pipeline [28]. Finally, we shed light on several promising directions for future work in Section 11.

2 SURVEY LANDSCAPE

2.1 Scope of the Survey

In order to narrow the scope of the survey within a controllable range, we focused on visualization-oriented natural language interfaces, which accept natural language queries as input and output appropriate visualizations automatically. Users can input natural language in various ways, including typing with a keyboard (e.g., Datatone [63] in Fig. 7), speech input via a microphone (e.g., InChorus [226] in Fig. 9), selecting text from articles (e.g., Metoyer *et al.* [162] in Fig. 13), and inputting existing textual descriptions (e.g., Vis-Annotator [122] in Fig. 12).

In addition, several fields are closely related to V-NLI. As shown in Table 2, the *Related Stage* column lists overlap stages with V-NLI in Fig. 3 that apply similar technologies. In order to make the survey more comprehensive, when introducing V-NLI in the following sections, we will involve additional crucial discussions on the aforementioned related fields, with explanations of their importance and relationship with V-NLI. For example, Visualization Recommendation (VisRec) [187], [289] acts as the back-end engine of V-NLI to recommend visualizations. Natural Language Interface for Database (NLI4DB) [6] and Conversational Interface for Visualization (CIV) [20] share similar principles with V-NLI. Natural Language Generation for Visualization (NLG4Vis) [144], [177] and Visual Question Answering (VQA) [110], [158] complement visual data analysis with natural language as output. Annotation [190], [213] and Narrative Storytelling [241] present fascinating visualizations by combining textual and visual elements.

2.2 Survey Methodology

To comprehensively survey V-NLI, we performed an exhaustive review of relevant journals and conferences
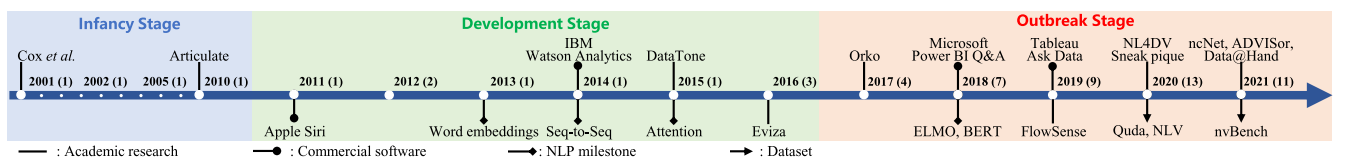

Fig. 2. Timeline of V-NLI. We briefly divide the timeline into the infancy, development, and outbreak stages. The number of yearly published papers is attached. The timeline consists of four parts: academic research, commercial software, NLP milestone, and dataset.
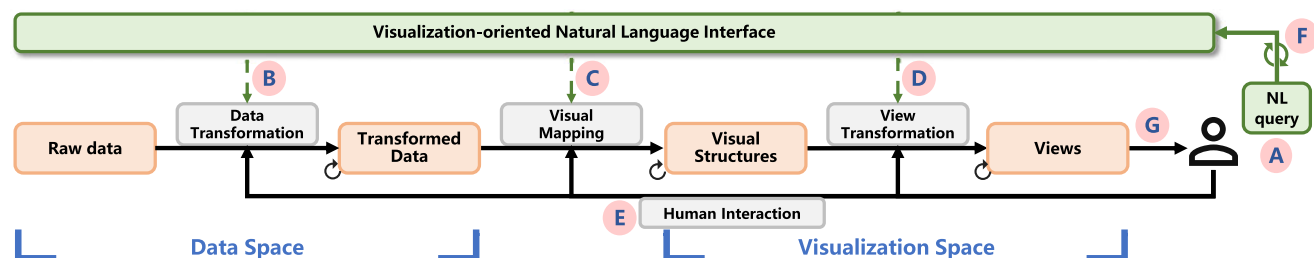
Fig. 3. Extension of classic information visualization pipeline proposed by Card *et al.* [28] with V-NLI. It depicts how V-NLI works to construct visualizations, which consists of the following seven stages: (A) Query Interpretation, (B) Data Transformation, (C) Visual Mapping, (D) View Transformation, (E) Human Interaction, (F) Dialogue Management, and (G) Presentation.

throughout the past twenty years (2000-2021), which broadly covers VIS, HCI, NLP, and DMM. The relevant venues are shown in Table 1. We began our survey by searching keywords ("natural language" AND "visualization") in Google Scholar, resulting in 1436 papers (VIS: 455, HCI: 489, NLP: 289, and DMM: 203). We also searched for representative related works that appeared earlier or in the cited references of related papers. During the review process, we first examined the titles of papers from these publications to identify candidate papers. Next, abstracts of the candidate papers were browsed to determine whether they were related to V-NLI. The full text was reviewed to make a final decision if we could not obtain precise information from the title and abstract. Finally, we collected 57 papers about V-NLI that accepts natural language as input and outputs visualizations. Details of the systems are listed in Table 3, including the applied NLP technologies, chart types supported, visualization recommendation algorithm adopted, and various characteristics in V-NLI, which will be discussed in Sections 4, 5, 6, 7, 8, 9, and 10. On this basis, we proceeded to a comprehensive analysis of the 57 papers to systematically understand the main research trends. We also collected 283 papers of related works described in Section 2.1. Table 2 lists representative papers of each topic for reference, which will be selectively discussed with V-NLI characteristics in the subsequent sections. More information can be found at https://v-nlis.github.io/V-NLIs-survey/.

3 CLASSIFICATION OVERVIEW

The information visualization pipeline presented by Card *et al.* [28] (see Fig. 3) describes how the raw data transits into visualizations and interacts with the user. We extend this pipeline with an additional V-NLI layer (Green colored in Fig. 3). On this basis, we move forward to develop

categorical dimensions by focusing on how V-NLI facilitates visualization generation. Inspired by McNabb *et al.* [160], to facilitate the categorization process, the following stages in the pipeline are used:

- *Query Interpretation (A):* Since we add the V-NLI layer in the pipeline, query interpretation is a foundational stage. Semantic and syntax analysis is generally performed first to discover hierarchical structures of the NL queries so that the system can parse relevant information in terms of data attributes and analytic tasks. Due to the vague nature of natural language, dealing with the underspecified utterances is another essential task in this stage. Details can be found in Section 4.
- *Data Transformation (B):* In the original pipeline, this stage mainly plays a role in transforming raw data into data tables along with various operations (e.g., aggregation and pivot). Since the majority of raw data to analyze is already in a tabular format, we rename it transformed data. Data transformation is responsible for generating alternative data subsets or derivations for visualization. All operations on the data plane belong to this stage. Details can be found in Section 5.
- *Visual Mapping (C):* This stage focuses on mapping the information extracted from NL queries to visual structures. The three elements of visual mapping for information visualization are spatial substrate, graphical elements, and graphical properties [28]. Spatial substrate is the space to create visualizations, and it is essential to consider the layout configuration. Graphical elements are the visual elements appearing in the spatial substrate (e.g., points, lines, surfaces, and volumes). Graphical properties can be implemented on the graphical elements to make them more noticeable (e.g., size, orientation, color, textures, and shapes). Details can be found in Section 6.
- *View Transformation (D):* View transformation (rendering) transforms visual structures into views by specifying graphical properties that turn these structures into pixels. Common forms include navigation, animation, and visual distortion (e.g., fisheye lens). However, in the context of our survey, this stage is rarely involved in V-NLI. Details can be found in Section 7.
- *Human Interaction (E):* Human interactions with the visualization interface feed back into the pipeline. A user can connect with a visualization manually by modifying or transforming a view state, or by reviewing the use, effectiveness, and knowledge on the

TABLE 1
Relevant Venues

Field	Venues
Visualization (VIS)	IEEE VIS (InfoVis, VAST, SciVis), EuroVis, PacificVis, TVCG, CGF, CGA
Human-Computer Interaction (HCI)	CHI, UIST, IUI
Natural Language Processing (NLP)	ACL, EMNLP, NAACL, COLING
Data Mining and Management (DMM)	KDD, SIGMOD, ICDE, VLDB

TABLE 2
Representative Papers of Related Works to V-NLI

Topic	Related Stage	Survey	Representative Paper
Visualization Recommendation (VisRec)	B/C/D/E/G	[187], [289]	[32], [37], [42], [46], [51], [70], [84], [86], [106], [109], [114], [130], [138], [142], [149], [152], [154], [168], [170], [185], [186], [203], [234], [246], [247], [253], [259], [260], [261], [263], [276], [277], [288]
Natural Language Interface for DataBase (NLI4DB)	A/B/E/F/G	[6], [290]	[13], [14], [15], [19], [22], [55], [72], [73], [74], [80], [87], [92], [95], [118], [135], [136], [137], [141], [173], [175], [188], [193], [202], [210], [220], [221], [252], [268], [270], [274], [285], [287]
Conversational Interface for Visualization (CIV)	A/B/C/D/E/F/G		[20], [58], [77], [129], [134], [171], [172], [201], [216]
Natural Language Generation for Visualization (NLG4Vis)	B/E/G		[39], [42], [44], [45], [59], [111], [144], [166], [167], [177], [184], [223]
Visual Question Answering (VQA)	A/B/E/F/G	[158]	[27], [30], [97], [98], [110], [143], [155], [222], [258], [269]
Annotation and Narrative Storytelling	A/B/C/D/E/F/G	[200], [241]	[11], [26], [32], [33], [34], [35], [36], [41], [62], [64], [65], [82], [89], [101], [112], [114], [119], [122], [123], [146], [162], [165], [183], [190], [213], [240], [254], [256], [257], [265], [275]

Related Stage column lists the overlap with V-NLI stages in Fig. 3 that apply similar technologies.

visualization. Dimara *et al.* [52] defined *interaction for visualization* as: *"The interplay between a person and a data interface involving a data-related intent, at least one action from the person and an interface reaction that is perceived as such."* Describing interaction requires all the mandatory components: interplay, user, and interface. Details can be found in Section 8.

- *Dialogue Management* (F): Interpreting an utterance contextually is essential for visualization system intelligence, which is particularly evident in V-NLI. This stage involves each step in the pipeline and concentrates on facilitating a conversation with the system based on the current visualization state and previous utterances. Details can be found in Section 9.
- *Presentation (G):* We name this stage as *Presentation*. The classic pipeline focuses on how to generate visualizations but ignores presenting them to the user. With natural language integrated into the pipeline, the vast majority of V-NLI systems accept natural language as input and directly display generated visualizations. Furthermore, complementing visualizations with natural language can provide additional surprises to the user. Details can be found in Section 10.

The pipeline is meant to model work activities. A single utterance can cover multiple stages (e.g., data transformation, visual mapping, and human interaction), and users can iterate over any of these stages. An utterance can also be interpreted contextually. The following seven sections will discuss the seven stages accordingly.

4 QUERY INTERPRETATION

Query interpretation is the foundation of all subsequent stages. This section will discuss how to perform semantic and syntax analysis of the input natural language queries, infer analytic tasks of the user and data attributes to be analyzed, and make defaults for underspecified utterances.

4.1 Semantic and Syntax Analysis

Semantic and syntax analysis can be powerful to discover hierarchical structures and understand meanings in human language. The semantic parser can conduct a series of NLP sub-tasks on the query string to extract valuable details that can be used to detect relevant phrases. The processing steps include tokenization, identifying parts-of-speech (POS) tags,

recognizing name entities, removing stop words, performing stemming, creating a dependency tree, generating N-grams, etc. For example, Flowsense [273] is a natural language interface designed for a dataflow visualization system [272]. It applies a semantic parser with special utterances (table column names, node labels, node types, and dataset names) tagging and placeholders. Fig. 4 displays a parse tree for the derivation of the user's query. The five major components of a query pattern and its related parts are highlighted by a unique color, and the expanded sub-diagram is illustrated at the bottom. Flowsense is powered by the Stanford SEMPRE framework [282] and CoreNLP toolkit [157]. Advanced NLP toolkits [1], [3], [7], [21], [81], [157], [182] allow developers to quickly integrate NLP services into their systems. As semantic and syntax analysis is a fundamental task for V-NLI, almost all systems support semantic parsing by directly using existing NLP toolkits, such as CoreNLP [157], NLTK [21], OpenNLP [1], SpaCy [81], Stanza [182], Flair [7], and GoogleNLP [3], as listed in Table 3 (column *NLP Toolkit or Technology*). We also sort out commonly used characteristics of existing NLP toolkits in Table 4 for reference. In addition, recently, there have been some systems that do not rely on these tools but leverage Language Models (LMs) to directly "interpret" queries [143], [150]. They first generate a rich representation of the input by translating them into high-dimensional vectors and then adopt neural networks to enable smart visualization inference.

Discussion. Most V-NLI systems rely on existing NLP toolkits for semantic and syntax analysis. So far, these tools are a good choice for query parsing and can be easily integrated into the system. However, they are trained on NLP datasets, thus lacking adequate consideration of visualization elements (e.g., mark, visual channel, and encoding property). A promising solution may be to develop a new NLP toolkit specifically for visualization.

4.2 Task Inference

4.2.1 Task Modeling

A growing body of literature recognizes that the user's analytic task is vital for automatic visualization creation [24], [108], [113], [191], [194]. As for task modeling, there have been many prior efforts to provide definitions of the analytic tasks. For instance, Amar *et al.* [9] proposed ten low-level analytic tasks that capture the user's activities while

TABLE 3
Summary of Representative Works in V-NLI

Name	Publication	NLP Toolkit or Technology	Visualization Type	Recommendation Algorithm	Semantic and Syntax Analysis	Task Inference	Data Attributes Inference	Default for Underspecified Utterances	Transformation for Data Insights	Spatial Substrate Mapping	Graphical Elements Mapping	Graphical Properties Mapping	View Transformation	Ambiguity Widget	Autocompletion	Multimodal	WIMP	Dialogue Management	Visual Presentation	Annotation	Narrative Storytelling	NL Description Generation	Visual Question Answering
Cox et al. [40]	IJST'01	Sisl	B/Ta	InfoStill	✗	✓	✓	✗	✓	✓	✓	✓	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗
Kato et al. [105]	COLING'02	Logical form	B/L/P/A	Template	✓	✓	✓	✗	✓	✓	✓	✓	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗
RIA [284]	InfoVis'05	-	I	Optimization	✓	✓	✓	✗	✓	✗	✗	✗	✗	✗	✗	✓	✗	✓	✓	✗	✗	✗	✗
Articulate [239]	LNCS'10	Stanford Parser	S/R/B/L/P/Bo	Decision tree	✓	✓	✓	✗	✓	✓	✓	✓	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗
Contextifier [89]	CHI'13	NLTK	L	-	✓	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✗
NewsViews [64]	CHI'14	OpenCalais	M	-	✓	✓	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✗
Datatone [63]	UIST'15	NLTK/Stanford Parser	S/B/L	Template	✓	✓	✓	✗	✓	✓	✓	✓	✗	✓	✗	✗	✓	✗	✓	✗	✗	✗	✗
Articulate2 [120], [121]	SIGDIAL'16	ClearNLP/Stanford Parser/OpenNLP	B/L/H	Decision tree	✓	✓	✓	✗	✓	✓	✓	✓	✗	✓	✗	✓	✗	✓	✗	✗	✗	✗
Eviza [205]	UIST'16	ANTLR	M/L/B/S	Template	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✗	✗	✗
TimeLineCurator [62]	TVCG'16	TERNIP	Tl	Template	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓
Analyza [50]	IUI'17	Stanford Parser	B/L	Template	✓	✓	✓	✗	✓	✓	✓	✓	✗	✓	✓	✗	✓	✓	✓	✗	✗	✗
TSI [26]	TVCG'17	Template	L/A/H	-	✗	✓	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✗	✗
Ava [134]	CIDR'17	Controlled Natural Language	S/L/B	Template	✓	✓	✓	✗	✓	✓	✓	✓	✗	✗	✗	✗	✓	✗	✓	✗	✗	✗	✓
DeepEye [148]	SIGMOD'18	OpenNLP	P/L/B/S	Template	✓	✗	✓	✗	✓	✓	✓	✓	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗
Evizeon [83]	TVCG'18	CoreNLP	M/L/B/S	Template	✓	✓	✓	✗	✓	✓	✓	✓	✗	✓	✗	✗	✓	✗	✓	✗	✗	✗
Iris [58]	CHI'18	Domain-specific language	S/T	Template	✓	✓	✓	✗	✓	✓	✓	✓	✗	✗	✗	✗	✓	✗	✓	✗	✗	✗	✓
Metoyer et al. [162]	IUI'18	CoreNLP	I	Template	✓	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✗	✓	✗
Orko [231]	TVCG'18	CoreNLP/NLTK	N	-	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✗	✗	✗
ShapeSearch [217], [218]	VLDB'18	Stanford Parser	L	ShapeQuery	✓	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗
Valletto [103]	CHI'18	Spacy	B/S/M	Template	✓	✓	✓	✗	✓	✓	✓	✓	✗	✓	✗	✗	✓	✗	✓	✗	✗	✗
ArkLang [208]	IUI'19	Cocke-Kasami-Younger	B/G/L/M/P/S/Tr	Template	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗
Ask Data [2], [242]	VAST'19	Proprietary	B/G/L/M/P/S/T	Template	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✗	✓	✗	✓	✗	✗	✗
Data2Vis [51]	CGA'19	Seq2Seq	B/A/L/S/T/St	Seq2Seq	✗	✓	✗	✗	✓	✓	✓	✓	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗
Hearst et al. [78]	VIS'19	WordNet	-	-	✓	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Voder [225]	TVCG'19	Stanford Parser	St/Bo/B/S/D	Template	✓	✓	✓	✗	✓	✓	✓	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗	✓	✗
AUDiaL [169]	LNCS'20	CoreNLP	B/L	Knowledge Base	✓	✓	✓	✗	✓	✓	✓	✓	✗	✓	✗	✗	✓	✗	✓	✗	✗	✗
Bacci et al. [10]	LNCS'20	Wit.ai	P/L/B/S	Template	✓	✓	✓	✗	✓	✓	✓	✓	✗	✗	✓	✗	✓	✗	✓	✗	✗	✗
DataBreeze [227]	TVCG'20	CoreNLP/NLTK	S/Uc	Template	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✗	✓	✓	✗	✓	✗	✗	✗
FlowSense [273]	TVCG'20	CoreNLP/SEMPRE	L/S/B/M/N/Ta	Template	✓	✓	✓	✗	✓	✓	✓	✓	✗	✗	✓	✗	✓	✗	✓	✗	✗	✗
InChorus [226]	CHI'20	CoreNLP/NLTK	L/S/B	Template	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✗	✗	✗
NL4DV_Quda [61]	arXiv'20	CoreNLP	St/B/L/A/P/S/Bo/H	Template	✓	✓	✓	✗	✓	✓	✓	✓	✗	✓	✓	✗	✓	✗	✓	✗	✗	✗
Sneak Pique [206]	UIST'20	ANTLR	-	-	✓	✓	✓	✗	✓	✗	✗	✗	✗	✗	✓	✗	✗	✓	✗	✗	✗	✗
Story Analyzer [165]	JCIS'20	CoreNLP	Wc/C/T/Fg/B/M/I	Template	✗	✗	✓	✗	✗	✓	✓	✓	✗	✗	✗	✗	✗	✓	✗	✓	✗	✗
Text-to-Viz [41]	TVCG'20	Stanford Parser	I	Template	✓	✓	✓	✗	✓	✓	✓	✓	✗	✗	✗	✗	✗	✓	✓	✓	✓	✗
Vis-Annotator [122]	CHI'20	Spacy/Mask-RCNN	B/L/P	-	✗	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗
Sentifiers [207]	VIS'20	ANTLR	-	-	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗
NL4DV [174]	TVCG'21	CoreNLP	St/B/L/A/P/S/Bo/H	Template	✓	✓	✓	✗	✓	✓	✓	✓	✗	✓	✓	✗	✓	✗	✓	✗	✗	✗
Data@Hand [115]	CHI'21	Compromise/Chrono	B/L/S/Ra/Bo	Template	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✗	✗	✗
Retrieve-Then-Adapt [183]	TVCG'21	Template	I	Template	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✗
ADVISor [143]	PacificVis'21	End-to-end network	B/L/S	Template	✓	✓	✓	✗	✓	✓	✓	✓	✗	✗	✗	✗	✗	✓	✓	✗	✗	✓
Seq2Vis [151]	SIGMOD'21	Seq-to-Sep model	B/L/S/P/D	Seq2Seq	✗	✗	✗	✗	✓	✓	✓	✓	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗
ncNet [150]	VIS'21	Transformer-based model	B/L/S/P	Template	✗	✗	✗	✗	✓	✓	✓	✓	✗	✗	✗	✗	✗	✓	✓	✗	✗	✓
GeoSneakPique [204]	VIS'21	ANTLR	-	-	✓	✓	✓	✗	✓	✗	✗	✗	✗	✓	✓	✗	✗	✓	✗	✗	✗	✗
Snowy [229]	UIST'21	CoreNLP	-	-	✓	✓	✓	✗	✗	✗	✗	✗	✗	✓	✓	✗	✓	✗	✗	✗	✗	✗
DT2VIS [93]	CGA'21	CoreNLP	B/L/S/P/Bo/R	Template	✓	✓	✓	✗	✓	✓	✓	✓	✗	✗	✓	✗	✓	✗	✓	✗	✗	✗

*Abbreviations for **Visualization Type**: Bar(B), Table(Ta), Infographic(I), Scatter(S), Radar(R), Line(L), Pie(P), Boxplot(Bo), Icon(Ic), Map(M), Heatmap(H), Timeline(Tl), Area(A), Network(N), Tree(Tr), Strip(St), Donut(D), Gantt(G), Word clouds(Wc), Force graph(Fg), Range(Ra), Unit column charts(Uc), and Graphics(Gr). The first five columns list basic information, and the left columns are characteristics of V-NLI with categorical dimensions described in Section 3. Details of each column will be discussed in Sections 4, 5, 6, 7, 8, 9, and 10.*

employing visualization tools for data exploration. The ten tasks are later extensively applied in numerous visualization systems [93], [113], [142], [168], [194], [212], [225], [245], [254], as listed in Table 5. Saket et al. [194] evaluated the effectiveness of five commonly used chart types across the ten tasks [9] by a crowdsourced experiment. Furthermore, they derived chart type recommendations for different tasks. Kim et al. [113] measured subject performance across task types derived from the ten tasks [9] and added a compare values task. The analytic tasks are

further grouped into two categories: value tasks that just retrieve or compare individual values and summary tasks that require identification or comparison of aggregate properties. NL4DV [174] includes a Trend task in addition. AutoBrief [107] enhances visualization systems by introducing domain-level tasks, while Matthew et al. [24] contributed a multi-level typology of visualization tasks. Deep into scatter charts, Sarikaya et al. [196] collected model tasks from a variety of sources in data visualization literature to formulate the seed for a scatterplot-specific
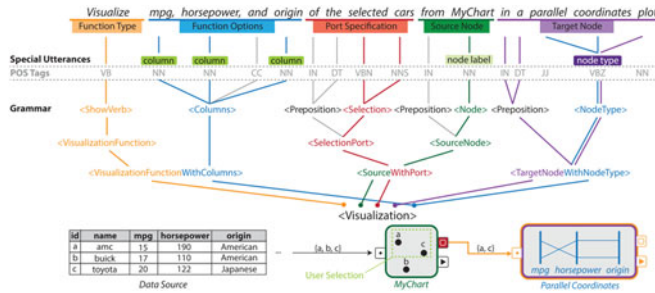
Fig. 4. Semantics parsing in FlowSense [273]. The derivation of the query is shown as a parse tree.

analytic task list. Recently, Shen *et al.* [211] summarized 18 classical analytic tasks by a survey covering both academia and industry.

4.2.2 Intent Inference

Although considerable previous works focus on task modeling, few visualization systems have attempted to infer the user's analytic task before the emergence of natural language interfaces. Gotz and Wen [70] monitored the user's click interactions for implicit signals of user intent. Steichen *et al.* [233] and Gingerich *et al.* [68] used the eye-gazing patterns of users while interacting with a given visualization to predict the user's analytic task. Battle *et al.* [16] investigated the relationship between latency, task complexity, and user performance. However, these behavior-based systems are limited to few pre-defined tasks; generalization for automatic visualization systems does not exist yet.

Rather than inferring the analytic task through the user's behavior, systems supporting NL interaction depend on understanding the NL utterances to analyze the user's intent

TABLE 4
Comparison of Commonly Used NLP Toolkits

Toolkit	CoreNLP [157]	NLTK [21]	OpenNLP [1]	SpaCy [81]	Stanza [182]	Flair [7]	GoogleNLP [3]
Programming language	Java	Python	Java	Python	Python	Python	Python
Tokenization	✓	✓	✓	✓	✓	✓	✓
Sentence segmentation	✓	✓	✓	✓	✓	✓	✓
Part-of-speech tagging	✓	✓	✓	✓	✓	✓	✓
Parsing	✓	✓	✓	✓	✓	✓	✓
Lemmatization	✓	✓	✓	✓	✓	✓	✓
Named entities recognition	✓	✓	✓	✓	✓	✓	✓
Coreference resolution	✓	✓	✓	-	✓	-	-
Entity linking	✓	✓	-	✓	✓	-	✓
Chunker	-	-	✓	✓	-	-	-
Sentiment	✓	-	-	-	✓	-	✓
Text classification	-	✓	-	✓	-	✓	✓
Train custom model	-	-	✓	✓	✓	✓	✓

TABLE 5
Ten Low-Level Analytic Tasks Proposed by Amar *et al.* [9]

Task	Description	Representative papers
Characterize Distribution	Given a set of data cases and a quantitative attribute of interest, characterize the distribution of that attribute's values.	[9], [61], [143], [174], [194], [196], [225], [254], [273]
Cluster	Given a dataset, find clusters of similar attribute values.	[9], [61], [194], [212], [225], [273]
Compute Derived Value	Given a dataset, compute an aggregate numeric representation of the data.	[9], [61], [86], [143], [194], [225], [254], [273]
Correlate	Given a dataset and two attributes, determine useful relationships between the values of those attributes.	[9], [61], [86], [143], [174], [194], [196], [212], [225], [254], [273]
Determine Range	Given a dataset and a data attribute, find the span of values within the set.	[9], [61], [194], [225], [273]
Filter	Find data cases satisfying the given concrete conditions on attribute values.	[9], [61], [174], [194], [225], [254], [273]
Find Anomalies	Identify any anomalies within a given set of data cases with respect to a given relationship or expectation, e.g., statistical outliers.	[9], [61], [85], [194], [196], [212], [225], [254], [273]
Find Extremum	Find data cases possessing an extreme value of an attribute over its range.	[9], [61], [85], [113], [143], [194], [225], [254], [273]
Retrieve Value	Given a set of specific cases, find attributes of those cases.	[9], [61], [85], [113], [174], [194], [225], [254], [273]
Sort	Given a dataset, rank them according to some ordinal metric.	[9], [61], [194], [225], [254], [273]

They have been widely applied for automatic visualization creation.

since they may hint at the user's analysis goals. Most systems infer the analytic tasks by comparing the query tokens to a predefined list of task keywords [63], [83], [174], [205], [239], [273]. For example, NL4DV [174] identifies five low-level analytic tasks: *Correlation*, *Distribution*, *Derived Value*, *Trend*, and *Filter*. A task keyword list is integrated internally (e.g., *Correlation* task includes 'correlate,' 'relationship,' etc., *Distribution* task includes 'range,' 'spread,' etc.). NL4DV also leverages POS tags and query parsing results to model the relationship between query phrases and populate task details. For conversational analysis systems, Nicky [116] interprets task information in conversations based on a domain-specific ontology. Evizeon [83] and Orko [231] support follow-up queries through conversational centering [71], which is a model commonly used in linguistic conversational structure but only for the *filter* task. ADE [38] supports the analytic process via task recommendation invoked by inferences about user interactions in mixed-initiative environments. The recommendation is also accompanied by a natural language explanation. Instead of simply matching keywords, Fu *et al.* [61] maintained a dataset of NL queries for visual data analytics and trained a multi-label task classification model based on BERT [49], an advanced pre-trained NLP model.

Discussion. Although most V-NLI systems support task inference, as shown in Table 3 (column *Task Inference*), the tasks integrated in the system are limited (e.g., a subset of ten low-level tasks [9]). More tasks with hierarchical modeling can be considered to better cover the user's intent, as well as tasks for analyzing specific data types (e.g., tree [235], map [54], and graph [127]). Besides, rule-based approaches account for the majority; various advanced learning models provide a great opportunity to infer analytic tasks in an unbiased and rigorous manner.

4.3 Data Attributes Inference

A dataset contains numerous data attributes. However, the user may only be interested in several certain data attributes in a query. The systems should be able to extract data attributes that are mentioned both explicitly (e.g., directly refer to attribute names) and implicitly (e.g., refer to an attribute's values or alias). Illustrated by an example, NL4DV [174] maintains an alias map and allows developers to configure aliases (e.g., *GDP* for *Gross domestic product* and *investment* for *production budget*). It iterates through the generated N-grams discussed in Section 4.1, checking for both syntactic and semantic similarity between N-grams and a lexicon composed of data attributes, aliases, and values. For syntactic similarity, NL4DV adopts cosine similarity function; for semantic similarity, it computes the Wu-Palmer similarity score based on WordNet [164]. If the similarity score reaches the threshold, the corresponding data attributes will be extracted and further presented on the visualization. Most V-NLI systems have taken a similar approach, and they just differ in the integrated rules. Besides, Analyza [50] utilizes additional heuristics to derive information from data attributes and expands the lexicon with a proprietary knowledge graph. Recently, Liu *et al.* [143] proposed a deep learning-based pipeline, ADVISor. It uses BERT [49] to generate embeddings of both NL queries and table headers, which are further used by a deep neural networks model to decide

data attributes, filter conditions, and aggregation type. Similarly, Luo *et al.* [150] presented an end-to-end solution using a transformer-based [248] sequence-to-sequence (seq2seq) model [12]. In the flow of visual analytical conversations, data attributes can also be extracted through co-reference resolution (Section 9.2).

Discussion. During the survey and evaluation of state-of-the-art systems, we found that most V-NLIs are primarily geared to support specification-oriented queries like examples in their manual (e.g., "Show me acceleration and horsepower across origins"), from which it is easy to extract data attributes. However, when it comes to underspecified or vague queries (e.g., include synonym, abbreviation, and terminology in different fields), these systems often fail to generate appropriate visualizations. So improving the robustness of V-NLIs should be one of the focuses in future work.

4.4 Default for Underspecified Utterances

Numerous systems support the *filter* task [63], [83], [174], [205], [273], aiming to select data attributes and ranges. An input query will be underspecified if it lacks enough information for the system to infer data attributes and perform filtering. Presenting defaults in the interface by inferring the underspecified utterances can be an effective option to address this issue. However, the area has received relatively little attention, as shown in Table 3 (column *Default for Underspecified Utterances*). Within the research, vague modifiers like "tall" and "cheap" are a prevalent kind of underspecification in human language. Hearst *et al.* [78] made the first step toward design guidelines for dealing with vague modifiers based on existing cognitive linguistics research and a crowdsourcing study. Tableau's Ask Data [2] internally leverages lightweight and intermediate language, Arklang [208], to infer underspecified details. It emphasizes how the linguistic context of previous utterances affects the interpretation of a new utterance. Sentifiers [207] can determine which data attributes and filter ranges to associate the vague predicates using word co-occurrence and sentiment polarities. As shown in Fig. 5, when analyzing the earthquakes dataset, the user inputs "where is it unsafe" and Sentifiers automatically associates "unsafe" with the *magnitude* attribute. A top N filter of *magnitude* six and higher is applied, and similar negative sentiment polarities are marked red on the map.

Discussion. Although the aforementioned approaches are useful, some more complex interpretations are still not supported in current V-NLIs, such as combinations of vague modifiers. So when encountering ambiguity, apart from formulating a sensible default, human interaction (e.g., ambiguity widgets) is another effective method (see Section 8.1). In addition, users' explicit judgments for the same utterance can differ significantly. A more adaptive solution may be to personalize the inferencing logic by identifying individuals' unique analytic tasks.

5 DATA TRANSFORMATION

After extracting data attributes, various data transformation operations can be made to transform raw data into focused data. This section will introduce how V-NLI systems transform raw data for data insights, with additional discussions on a closely related topic, NLI for databases.
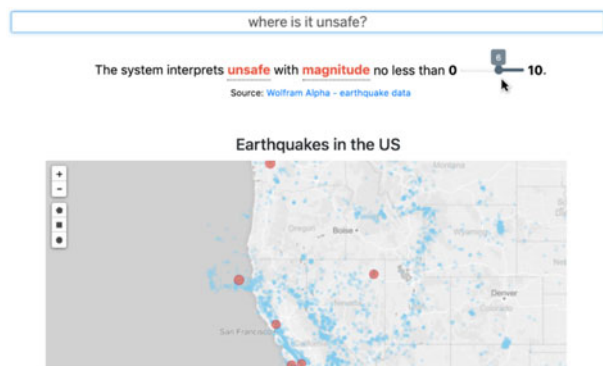
Fig. 5. Vague modifier interpretation in Sentifiers [63]. The system associates the vague modifier "unsafe" with *magnitude* attribute.

5.1 Transformation for Data Insights

A consensus is that the purpose of visualization is insights, not pictures. A "boring" dataset may become "interesting" after data transformations (e.g., aggregation, binning, and grouping). Therefore, identifying the data transformation information is another important characteristic in V-NLI. To describe related works, we categorize tabular data into four types: temporal, quantitative, nominal, and ordinal, which are widely adopted in the visualization community [168]. For temporal data, systems can support binning based on temporal units [62], [148], [165], [169], [174], [205], [273]. For instance, Setlur *et al.* [205] developed a temporal analytics module based on temporal entities and expressions defined in TimeML [91]. To parse temporal expressions, this module incorporates the following temporal token types: Temporal Units (e.g., years, months, days, hours and minutes, and their abbreviations.), Temporal Prepositions (e.g., 'in,' 'during,' 'near,' and 'around.'), and Temporal Connectives (e.g., 'before' and 'after'). For quantitative data, various aggregation functions (e.g., count, average, and sum) can be performed, binning and grouping operations are also commonly used [10], [38], [41], [50], [63], [83], [93], [121], [148], [159], [178], [205], [208], [225], [273], [284]. For example, DeepEye [148] includes various aggregation functions in the visualization language. Arklang [208], which is an intermediate language to resolve NL utterances to formal queries, integrates a finite set of aggregation operators. Eviza [205] enables aggregation using regular spatial bins. Unlike continuous data, nominal and ordinal data are usually used as grouping metrics. As an end-to-end system, ncNet [150] supports more complex data transformation types (e.g., relational join, GroupBY, and OrderBY) by applying Neural Machine Translation (NMT) models. After extracting relevant information, the commonly used visualization specification languages (e.g., Vega [198], Vega-Lite [197], and VizQL [234]) all provide various data transformation primitives to realize. Besides, to our knowledge, most V-NLIs are designed to deal with tabular data, but there are also several systems that focus on specific data types (e.g., network [231] and map [204]). However, they mostly only display the raw data without data transformations.

Discussion. The back-end engine can directly perform calculations and visualize the results if the data transformation information can be extracted from the user's NL queries. Otherwise, despite advances in technologies for data management and analysis [17], [67], it remains time-consuming to inspect a dataset and construct a visualization that allows meaningful analysis to begin. So a practical system should be in a position to automatically recommend data insights for users. Fortunately, various systems have been designed to generate data facts in the context of visual data exploration, such as DataSite [42], Voder [225], Wrangler [99], Quickinsights [53], Foresight [46], and SpotLight [75]. They can serve as back-end engines for insight-driven recommendations.

5.2 NLI for Database

Natural Language Interface for Database (NLI4DB), or Text-to-SQL, is a task to automatically translate a user's query text in natural language into an executable query for databases like SQL [6]. NLI4DB is strongly related to data transformation in V-NLI, and V-NLI can augment NLI4DB with effective visualizations of query results. Besides, not all the queries about a dataset need a visualization as a response. For instance, one might ask, "What was the GDP of the USA last year?" or "Which country won the most gold medals in the Tokyo Olympics?". In such cases, NLI4DB can directly compute and present values in response to the user's questions instead of displaying through visualizations that require the user to interpret for answers. Generally, there are three types of methods for NLI4DB: symbolic approach, statistical approach, and connectionist (neural network) approach [175]. The symbolic approach utilizes explicit linguistic rules and knowledge to process the user's query, which dominated the area in the early years. DISCERN [221] is a representative example and integrates a model that extracts information from scripts, lexicon, and memory. The statistical approach exploits statistical learning methods such as hidden Markov models [141] and probabilistic context-free grammars [87]. From the view of the connectionist approach, the Text-to-SQL task is a kind of special Machine Translation (MT) problem. However, it is hard to be directly handled with standard end-to-end neural machine translation models due to the absence of detailed specification in the user's query [73]. Another problem is the out-of-domain items mentioned in the user's query due to the user's unawareness of the ontology set. Aiming at the two problems, IRNet [73] proposes a tree-shaped intermediate representation synthesized from the user's query. It is later fed into an inference model to generate SQL statements with a domain knowledge base. Furthermore, to capture the special format of SQL statements, SQLnet [268] adopts a universal sketch as a template and predicts values for each slot. Wang *et al.* [252] employed a two-stage pipeline to subsequently predict the semantic structure and generate SQL statements with structure-enhanced query text. Recently, TaPas [80] extends BERT's architecture and trains from weak supervision to answer questions over tables. For more details, please refer to the survey [6], [290] and related papers in Table 2.

Discussion. V-NLI can augment NLI4DB with well-designed visualizations, and NLI4DB can complement V-NLI with exact value answers. First, the two fields can draw on each other as they share similar principles. Second, we can combine the two technologies to generate visualizations that "contain" answers. Third, there are many benchmarks in the NLI4DB community like WikiSQL [287] and Spider [274], which can be utilized to further design V-NLI benchmarks.

6 VISUAL MAPPING

This section will discuss how V-NLIs perform visual mapping in: spatial substrate, graphical elements, and graphical properties.

6.1 Spatial Substrate

Spatial substrate is the space to create visualizations. It is important to determine the layout configurations to apply in the spatial substrate, such as which axes to use. Some V-NLI systems support explicitly specifying layout information like inputting "show *GDP* series at y axis and *year* at x axis grouped by *Country Code*". If the mapping information is not clarified in the query, the search space will be very huge, in which some combinations of data attributes and visual encodings may not generate a valid visualization. For instance, the encoding type "*Y*-axis" is inappropriate for categorical attributes. Fortunately, there are many design rules either from traditional wisdom or users to help prune meaningless visualizations. These design rules are typically given by experts. Voyager [259], [260], [261], DIVE [86], Show me [154], Polaris [234], Profiler [100], DeepEye [149], and Draco [168] all contribute to the enrichment of design rules with more data types. Besides, many systems [168], [247], [261] also allow users to specify their interested data attributes and assign them on certain axes, which is a more direct way to perform visual mapping. In Falx [250], users can specify visualizations with examples of visual mapping, and the system automatically infers the visualization specification and transforms data to match the design. With the development of machine learning in recent years, advanced models can be applied for more effective visual mapping. VizML [84] identifies five key design choices while creating visualizations, including mark type, X or Y column encoding, and whether X or Y is the single column represented along that axis or not. For a new dataset, 841 dataset-level features extracted are fed into neural network models to predict the design choices. Luo *et al.* [150] proposed an end-to-end approach that applies the transformer-based seq2seq [12] model to directly map NL queries and data to chart templates. In addition to single chart, considering multiple visualizations, Evizeon [83] uses a grid-based layout algorithm to position new charts, while Voder [225] allows the user to choose a slide or dashboard layout.

Discussion. When performing visual mappings in the spatial substrate, the vast majority of systems are limited to 2-dimension space (along x and y axes). However, considering more axes types, creating 3-dimensional (e.g., add z axes) and even hyper-dimensional (e.g., parallel coordinate) representations is possible.

6.2 Graphical Elements

Graphical element (e.g., points, lines, surfaces, and volumes), which is usually named mark or chart type, is an important part of visualizations. Choosing an appropriate mark can convey information more efficiently. Similar to deciding the layout, some V-NLI systems allow users to specify marks like inputting "create a scatterplot of *mpg* and *cylinders*", which is a simple way to map mark information of visualizations. However, in most cases, the mark information is not accessible. So after parsing the NL queries, most systems

TABLE 6
Mappings Between Data Attribute Combinations (N: Numeric, C: Categorical), Analytic Tasks, and Chart Types in Voder [225]

Attributes	Task(s)	Visualization
N	Find Extremum	Strip plot
	Characterize Distribution	Strip plot
		Box plot
		Histogram
	Find Anomalies	Strip plot
		Box plot
C	Find Extremum	Bar chart Domut
	Characterize Distribution	chart
NxN	Corelation	Scatterplot
	Characterize Distribution	
CxN	Characterize Distribution (+Derived Value)	Bar chart Domut chart
	Find Extremum(+Derived Value)	
	Find Extremum	Strip plot Scatterplot
CxC	Find Extremum	Stacked bar chart
	Characterize Distribution (+Find Extremum)	Scatterplot+Size
NxNxN	Characterize Distribution	Scatterplot+Size
CxNxN	Correlation	Scatterplot
	Correlation(+Filter)	Scatterplot+Color
	Characterize Distibution (+Filter)	Scatterplot+Color Scatterplot+Size
CxCxN	Find Extremum	Strip plot+Color Scatterplot+Color
	Find Extremum(+Derived Value)	Strip plot+Color Scatterplot+Color Scatterplot+Size

integrate predefined rules or leverage Visualization Recommendation (VisRec) technologies to deal with the extracted elements. Voyager [260], [261] applies visualization design best practices drawn from prior research [153], [154], [192], [203] and recommends mark types based on the data types of the x and y channels. Srinivasan *et al.* [225] developed the heuristics and mappings between data attribute combinations, analytic tasks, and chart types in Table 6 through iterative informal feedback from fellow researchers and students. TaskVis [211] integrates 18 classical low-level analysis tasks with their appropriate chart types by a survey both in academia and industry. Instead of considering common visualization types, Wang *et al.* [253] developed rules for automatic selection of line graphs or scatter plots to visualize trends in time series, while Sarikaya *et al.* [196] deepened into scatter charts. To better benefit from design guidance provided by empirical studies, Moritz *et al.* [168] proposed a formal framework that models visualization design knowledge as a collection of answer set programming constraints.

Discussion. V-NLIs mostly maintain an alias map for data attribute inference (see Section 4.3) but often ignore mark inference. Therefore, though they can explicitly interpret mark information from the user's NL queries, they may fail when the information is implicitly expressed (e.g., "point" instead of "scatterplot"). Besides, as shown in Table 3 (column *Visualization Type*), existing works focus more on traditional chart types, more chart types, and even interactive visualizations can be extended in future work.

6.3 Graphical Properties

Jacques Bertin first identified seven "retinal" graphical properties (visual encoding properties) in visualizations: position, size, value, texture, color, orientation, and shape. Some other types are later expanded in the community, such as "gestalt" properties (e.g., connectivity and grouping) and animation [153], [243]. For visual mapping in V-NLI, color, size, and shape are most commonly applied to graphical elements, making them more noticeable. The rule-based approach is also dominant here, and human perceptual effectiveness metrics are usually considered. For example, the cardinality of variables in the color/size/shape channel should not be too high (e.g., 100). Otherwise, the chart would be too messy to distinguish. The aforementioned works also developed various design rules for graphical properties, especially for color, size, and shape channels in legend [86], [100], [149], [154], [168], [211], [234], [260], [261]. Besides, Text-to-Viz [41] integrates a color module that aims to generate a set of color palettes for a specific infographic. Similarly, InfoColorizer [275] recommends color palettes for infographics in an interactive and dynamic manner. Wu *et al.* [264] proposed a learning-based method to automate layout parameter configurations, including orientation, bar bandwidth, max label length, etc. Liu *et al.* [146] explored data-driven mark orientation for trend estimation in scatterplots. CAST [65] and Data Animator [240] enable interactive creation of chart animations.

Discussion. For graphical properties, color, size, and shape account for the majority in current V-NLIs. Although some other forms have been investigated deeply, they are rarely used in V-NLI (e.g., orientation and texture). Besides, color is a tricky choice to use. On one hand, it is unfriendly to those who are color-blind. On the other hand, it may involve cultural issues in certain scenarios (e.g., red is a "festive" color in China, whereas it may mean "warning" in many Western societies). So color needs to be used with the discretion of color vision deficiencies and cultural meaning.

7 VIEW TRANSFORMATION

After visual mapping, the generated visualization specifications can be rendered through a library (e.g., D3 [23]). View transformations are also supported here. The three commonly used view transformation types are location probes which reveal additional information through locations, viewpoint controls that scale or translate a view, and distortions that modify the visual structure [28]. Surprisingly, this stage is rarely used in the context of our survey. To our knowledge, there have been few works that focus on natural language interaction for view transformation. It is usually realized through multimodal interaction, as shown in Table 3 (column *View Transformation*). For example, Orko [231] facilitates both natural language and direct manipulation input to explore network visualization. Fig. 6 is the user interface of Orko, and it shows an example that explores a network of European soccer players. When the user says "Show connection between Ronaldo and Bale", the system will zoom in to display the details of nodes. It also allows users to perform view transformation (e.g., zooming and panning) through a pen or finger. Similarly, Valletto [103], InChorus [226], Data@Hand [115], DataBreeze [227], Power BI's Q&A [5], and Tableau's Ask data [2] all support standard view transformations. Besides, the
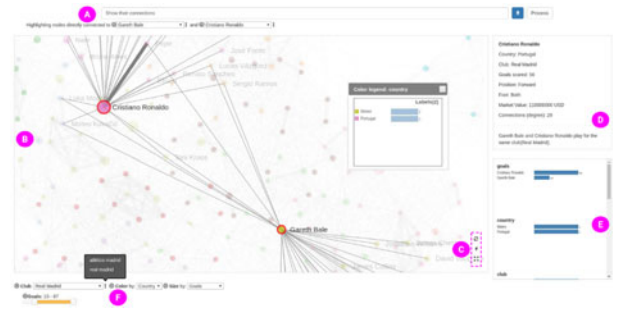

Fig. 6. User interface of Orko [231] to explore a network of European soccer players.

software visualization community also presents several closely related works. Bieliauskas *et al.* [20] proposed an interaction approach with software visualizations based on a conversational interface. It can automatically display best fitting views according to meta information from natural language sentences. Seipel *et al.* [201] explored natural language interaction for software visualization in Augmented Reality (AR) with Microsoft HoloLens device, which can provide various view transformation interactions through gesture, gaze, and speech.

Discussion. View transformation has attracted less attention in the community. Only a few V-NLI systems support NL control over view transformations, and they are mainly limited to viewpoint navigation. Future improvements can focus on other aspects (e.g., animation [240], data-GIF [215], and visual distortions [176]).

8 HUMAN INTERACTION

This section will discuss how V-NLI allows users to provide feedback to visualization with human interactions. The primary purpose is to help users better express their intents.

8.1 Ambiguity Widgets

Due to the vague nature of natural language, the system may fail to recognize the user intent and extract data attributes [161]. Considerable research has been devoted to addressing the ambiguity and underspecification of NL queries. Mainstream approaches can be divided into two categories. One is to generate appropriate defaults by inferencing underspecified natural language utterances, discussed in Section 4.4. The other is to return the decision right to users through ambiguity widgets. Ambiguity widget is a kind of interactive widget that allows users to input through a mouse. Datatone [63] first integrates ambiguity widgets and presents a mixed-initiative approach to manage ambiguity in the user query. As shown in Fig. 7, for the query in (b), DataTone detects data ambiguities about *medal*, *hockey*, and *skating*. Three corresponding ambiguity widgets are presented for users to correct the system choices in (c). Additional design decision widgets like color and aggregation are also available in (e). Eviza [205], Evizeon [83], NL4DV [174], and AUDiaL [169] all borrow the idea of DataTone and expand the ambiguity widgets to richer forms (see Table 3 (column *Ambiguity Widgets*)), such as maps and calendars. DIY [173] enables users to interactively assess the response from NLI4DB systems for correctness.

Fig. 7. Ambiguity widgets in Datatone [63]. The user can correct the imprecise system decisions caused by ambiguity.

Fig. 8. Autocompletion in Sneak pique [206], The user is prompted with widgets that provide appropriate previews of the underlying data.

Discussion. Resolving underspecified queries through ambiguity widgets mainly consists of two steps: detecting ambiguity and presenting widgets to the user. For the first step, most systems now still leverage heuristics for algorithmic resolution of ambiguity. A general probabilistic framework may handle a broader range of ambiguities. For the second, richer interactive widgets can be adopted to enhance the user experience.

8.2 Autocompletion and Command Suggestion

Users may be unaware of what operations the system can perform and whether a specific language structure is preferred in the system. Although advanced text understanding models give users the freedom to express their intents, system discoverability to help formulate analytical questions is still an indispensable part of V-NLI. Discoverability entails *awareness*: making users aware of what operations the system can perform, and *understanding*: educating users about how to phrase queries that can be interpreted correctly by the system [224]. Generally, current V-NLI systems pay relatively little attention to system discoverability. Major attempts include autocompletion and command suggestion. This characteristic offers interpretable hints or suggestions matching the visualizations and datasets, which is considered a fruitful interaction paradigm for information sense-making. When using V-NLI of Tableau [2] or Power BI [5], the autocompleted content will be presented as we are typing, especially when there is a parsing error. They are mostly reminders of commonly used queries, such as data attributes and aggregation functions. Suggestions will not intrusively appear when the user has formulated a valid query. Similarly, Bacci *et al.* [10] and Yu *et al.* [273] both adopted template-based approaches to support autocompletion in their V-NLIs. In addition to text prompts, data previews can be more useful across all autocompletion variants. Deeper into this area, three crowdsourcing studies conducted by Setlur *et al.* [206] indicated that users prefer widgets for previewing numerical, geospatial, and temporal data while textual autocompletion for hierarchical and categorical data. On this basis, they built a design probe, Sneak Pique. As shown in Fig. 8, when analyzing a dataset of coronavirus cases, the user is prompted with widgets that provide appropriate previews of the underlying data in various types. The system also supports toggling from a widget to a

corresponding text autocompletion dropdown. Most recently, Srinivasan *et al.* [229] proposed Snowy, a prototype system that generates and recommends utterance recommendations for conversational visual analysis by suggesting data features while implicitly making users aware of which input the NLI supports.

Discussion. Current utterance realization approaches are all template-based, which just works effectively for a small set of tasks and can hardly apply to large-scale systems. Going forward, another important consideration that needs further research is which commands to show and when and how the suggestions will be presented. In addition, existing technologies can only handle keyboard input. There remains an open area for the discoverability of speech-based V-NLI. The tone of voice can also provide insights into the user's sentiments [147].

8.3 Multimodal Interaction

Van Dam [244] envisioned post-WIMP user interfaces as "one containing at least one interaction technique not dependent on classical 2D widgets such as menus and icons." With the advancements in hardware and software technologies that can be used to support novel interaction modalities, researchers are empowered to take a step closer to post-WIMP user interfaces, enabling users to focus more on their tasks [128]. A qualitative user study conducted by Saktheeswaran *et al.* [195] also found that participants strongly prefer multimodal input over unimodal input. In recent years, many works have examined how multiple input forms (e.g., mouse, pen, touch, keyboard, and speech) can be combined to provide a more natural and engaging user experience. For instance, Orko [231] is a prototype visualization system that combines both natural language interface and direct manipulation to assist visual exploration and analysis of graph data. Valletto [103] allows users to specify visualizations through a speech-based conversational interface, multitouch gestures, and a conventional GUI interface. InChorus [226] is designed to maintain interaction consistency across different visualizations, and it supports pen, touch, and speech input on tablet devices. Its interface components are shown in Fig. 9, including typical WIMP components (A, B, C), speech command display area (D), and content supporting pen and touch (E, F, G). Data-Breeze [227] couplings pen, touch, and speech-based multimodal interactions with flexible unit visualizations, enabling a novel data exploration experience. RIA [284] is an intelligent multimodal conversation system to aid users in exploring large and complex datasets, which is powered by an
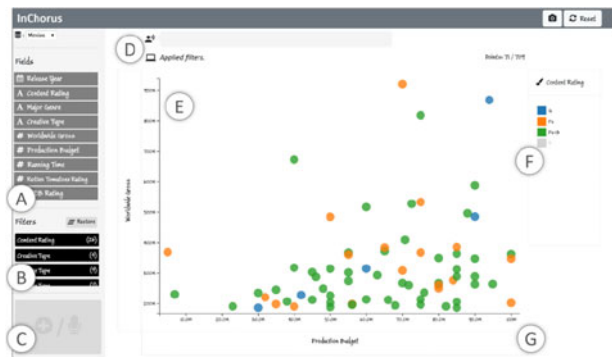
Fig. 9. Multimodal interaction interface in InChorus [226], which allows pen, touch, and speech input on tablet devices.

Fig. 10. Conversational V-NLI in Evizeon [83]. The user has a back-and-forth information exchange with the system.

optimization-based approach to visual context management. Data@Hand [115] leverages the synergy of speech and touch input modalities for personal data exploration on the mobile phone. Srinivasan *et al.* [224] proposed to leverage multi-modal input to enhance discoverability and the human-computer interaction experience.

Discussion. V-NLI is an essential part in the context of post-WIMP interaction with visualization systems. Typing and speech are both commonly used modalities to input NL queries. However, speech has unique challenges from a system design aspect, such as triggering speech input, lack of assistive features like autocompletion, and transcription errors [128], which deserve more attention in the future. Besides, limited research has incorporated human gesture recognition and tracking technology [94] to facilitate visualization creation, especially with large displays.

9 DIALOGUE MANAGEMENT

This section will discuss dialogue management in V-NLI. Given the conversational nature of NLIs, users may frequently pose queries depending on their prior queries. By iterating upon their questions, the user can dive deep into their aspects of interest on a chart and refine existing visualizations.

9.1 Analytical Conversations

Conversional NLIs have been widely applied in many fields to prompt users to open-ended requests (e.g., recommender systems [102] and intelligent assistants [219]). With conversational agents, users can also reinforce their understanding of how the system parses their queries. In the visualization community, Eviza [205] and Evizeon [83] are two representative visualization systems that provide V-NLI for visual analysis cycles. Fig. 10 shows a back-and-forth information exchange process between Evizeon [83] and the user when analyzing the measles dataset. The system supports various forms of NL interactions with a dashboard applying pragmatic principles. For example, when the user first types "measles in the UK", all the charts presented are filtered to cases of measles in the UK (Fig. 10a). Then, the user types "show me the orange spike", and Evizeon will add details to the spike in the orange line as it understands that this is a reference to visual properties of the line chart (Fig. 10b). Similarly, Articulate2 [121] is intended to automatically transform the user queries into visualizations using a full-fledged conversational interface. Analyza [50] combines
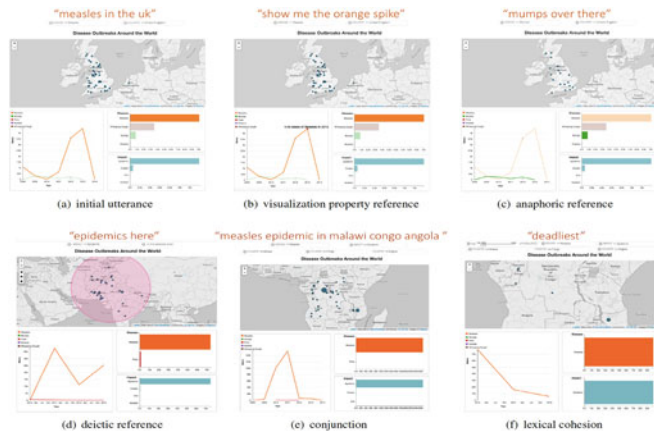
two-way conversation with a structured interface to enable effective data exploration. Besides, Fast *et al.* [58] proposed a new conversational agent, Iris, which can generate visualizations from and combine the previous commands, even non-visualization related. Ava [134] uses controlled NL queries to program data science workflow.

Conversational transitions model, which describes how to transition visualization states during the analytical conversation, is an essential part to enable the aforementioned interactive conversations. In the early years, Grosz *et al.* [71] explored how the context of a conversation adjusts over time to maintain coherence through transitional states (retain, shift, continue, and reset). On this basis, Tory and Setlur [242] introduced a conversational transitions model (see Fig. 11) that emerged from their analysis in the design of Tableau's V-NLI feature, Ask data. After interpreting a generated visualization, a user may formulate a new NL query to continue the analytical conversation. The user's transitional goal means how the user wishes to transform the existing visualization to answer a new question. The model contains the following transitional goals: elaborate, adjust/pivot, start new, retry, and undo, which in turn drive user actions. The visualization states (select attributes, transform, filter, and encode) are finally updated, and new visualizations are presented to the user. An essential insight during the analysis of Tory and Setlur was that applying transitions to filters alone (like Evizeon [83] and Orko [231]) is insufficient. The user's intent around transitions may apply to any aspects of a visualization state.

Discussion. Some V-NLI systems leverage a VisRec engine to generate visualizations. However, it is insufficient for analytical conversation. A more intelligent system should infer the user's transitional goals based on their interactions and then respond to each visualization state accordingly. Besides, empirical studies [77] have shown that users tend to prefer additional content beyond the exact answers when asking questions in such conversational interfaces, which had significant implications for designing insight-driven conversation interfaces in the future.

9.2 Co-Reference Resolution

Co-reference Resolution (CR) aims to find linguistic expressions (called mentions) in a given text that refers to the
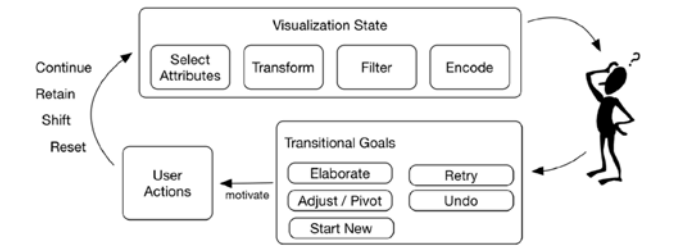
Fig. 11. Conversational transitions model [242] that describes how to transition a visualization state during an analytical conversation.

same entity [236]. CR is an important sub-task in dialogue management as the user usually uses pronouns to refer to certain visual elements. A typical scenario is to identify the entity to which a pronoun in the text refers. The task also involves recognizing co-reference relations between multiple noun phrases [181]. Table 7 shows examples of co-reference types. Before the booming development of deep learning-based language models, human-designed rules, knowledge, and features dominated CR tasks [280]. Nowadays, most state-of-the-art CR models are neural networks and employ an end-to-end fashion, where the pipeline includes encoding context, generating representations for all potential mentions, and predicting co-reference relations [131], [96]. Benefiting from the powerful understanding and predicting ability of language models like BERT [49] and GPT-3 [25], reliance on annotated span mentions is becoming weaker [117]. A CR task can even be treated as a token prediction task [118]. Several existing models exist to deal with this situation. Besides, the usage of structured knowledge bases [279], [145] and higher-order information [132] has been proven to be beneficial to the overall performance of CR tasks.

Co-reference resolution is also an essential task in the multimodal sense as users may even pose queries that are the follow-up to their direct manipulations on the interface. When the user faces a graphical representation, each element in the representation can become a referent. The system should have a declarative representation for all these potential referents and find the best match to make multimodal interaction more smooth. Articulate2 [120] addresses

this issue by leveraging Kinect to detect deictic gestures on the virtual touch screen in front of a large display. If referring expressions are detected, and Kinect has detected a gesture, information about any objects pointed to by the user will be stored. The system then can find the best match between properties of each relevant entity. The properties of visualizations and objects keeping track of include statistics and trends in the data, title, mark, and any more prominent objects within the visualization (e.g., hot-spots, street names, and bus stops). In multimodal systems [103], [115], [195], [224], [226], [227], [231], users can focus on specific visual elements in the visualization by selecting with mouse, pen, or finger. The follow-up queries can automatically link to related visual elements or data.

Discussion. At the natural language level, existing V-NLIs mostly leverage NLP toolkits to perform co-reference resolution. Although useful, they lack detailed modeling of visualization elements, as discussed in Section 4.1. At the multimodal level, in addition to commonly used modalities (e.g., mouse, pen, and finger), it may enable a better user experience by integrating more modalities like eye-gaze and gesture.

10 PRESENTATION

In most cases, V-NLI accepts natural language as input and outputs well-designed visualizations. As shown in Table 3 (column *Visualization Type*), the output is not limited to traditional charts but also involves other richer forms (e.g., maps, networks, and infographics). In addition to visual presentation, an emerging theme is to complement visualizations with natural language. The previous sections have discussed natural language as an input modality. This section will introduce using natural language as an output modality.

10.1 Annotation
Annotation plays a vital role in explaining and emphasizing key points in the dataset. The systems should generate valuable natural language statements and map the text to a visualization appropriately. Annotation tools were first applied to complement news articles. Kandogan [101] introduced the concept of just-in-time descriptive analytics that helps users easily understand the structure of data. Given a piece of news, Contextifier [89] first computes clusters, outliers, and trends in line charts and then automatically produces annotated stock visualizations based on the information. Although it offers a promising example of how human-produced news visualizations can be complemented in a specific context, it is only suitable for stock sequence data. NewsViews [64] later extends Contextifier to support more types of appropriate data such as time series and georeferenced data. In addition, several works incorporate the visual elements in the chart into annotations, synchronizing with textual descriptions. As shown in Fig. 12, Vis-Annotator [122] leverages the Mask R-CNN model to identify visual elements in the target visualization, along with their visual properties. Textual descriptions of the chart are synchronously interpreted to generate visual search requests. Based on the identified information, each descriptive sentence is displayed beside the described focal areas as

TABLE 7
Examples of Co-Reference Types

Type	Example	Resolution
Pronoun-Noun	Show me the most popular **movie** in **New York** and **its** rating.	its → movie's
Noun-Noun	How many **Wal-Mart stores** in **Seattle** and how's the annual profits for the **malls**?	malls → Wal-Mart
Pronoun-Pronoun	The **manchurian tigers** have been heavily hunted, which caused a dramatic drop of **their** existing number and **they** may finally get extinct.	they, their → manchurian tigers
Pronoun-Pronoun	**We** are all the **Canadians** while **Lily** was born in the **U.S.** and **she** immigrated to **Canada** two years ago.	We → Canadians, she → Lily
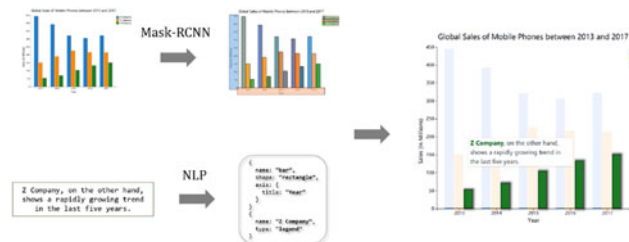
Fig. 12. Visualizations with annotations generated by Vis-Annotator [122].

Fig. 13. Coupling from the text about NBA game report to stories [162].

annotations. Similarly, Click2Annotate [33] and Touch2Annotate [34] are both semi-automatic annotations generators. To be interactive, Calliope [213] and ChartAccent [190] support creating stories via interactive annotation generation and placement. Most recently, ADVISor [143] can generate visualizations with annotations to answer the user's NL questions on tabular data.

Discussion. Annotation can turn data into a story. However, the scalability of current works is weak as most of them are template-based. To improve the system usability, advanced NLG models [29] can be leveraged to assist generating appropriate textual annotations. Extending existing systems to support richer data types and annotation forms is also a meaningful direction. Besides, it would be helpful to devise an annotation specification language for additional flexibility.

10.2 Narrative Storytelling

Narrative storytelling gives data a voice and helps communicate insights more effectively. A data story means a narrative to the data that depicts actors and their interactions, along with times, locations, and other entities [200]. Constructing various visual and generated textual (natural language) elements in visualizations is vital for narrative storytelling. To address this issue, the template is the most commonly applied method. For instance, Text-to-Viz [41] is designed to generate infographics from NL statements with proportion-related statistics. It builds 20 layout blueprints that describe the overall look of the resulting infographics. For each layout blueprint, Text-to-Viz enumerates all extracted segments by text analyzer and then generates all valid infographics. DataShot [254], TSIs [26], Chen *et al.* [36], and Retrieve-Then-Adapt [183] all use similar template-based approaches for narrative storytelling. With the advances of deep learning, some works leverage generative adversarial networks (GAN) [69] to synthesize layouts [139], [286]. Those learning-based methods do not rely on hand-crafted features. However, due to limited training data, they often fail to achieve comparable performance with rule-based methods. Likewise, storytelling has been extensively used to visualize narrative text produced by online writers and journalism media. Metoyer *et al.* [162] proposed a novel approach to generate data-rich stories. It first extracts narrative components (who, what, when, where) from the text and then generates narrative visualizations by integrating the supporting data evidence with the text. As shown in Fig. 13, given a highlighted NBA game report sentence, the system generates an initialized story dashboard based on the interpretation information. Similarly, Story

Analyzer [165] extracts subjects, actions, and objects from the narrative text to produce interrelated and user-responsive story dashboards. RIA [284] dynamically derives a set of layout constraints (e.g., ensure visual balance and avoid object occlusion) to tell spatial stories.

Discussion. Although the aforementioned systems are promising and work relatively well, they are restricted to one specific type of information respectively. More types of information (e.g., timeline and map) can be combined to create more expressive stories. Besides, similar to annotation, template-based methods also dominate narrative storytelling. It will be interesting to explore learning-based approaches to further improve the story quality as well as formulating datasets specially for this topic.

10.3 Natural Language Generation for Visualization

In the past years, there has been a body of research on automatically generating descriptions or captions for charts. The majority of early works are rule-based [39], [44], [57], [166], [167]. While modern pipelines generally include a chart parsing module and a subsequent caption generation module. Chart parsing module [8], [180], [199] deconstructs the original charts and extracts valuable elements such as text, axis, and lines using relevant techniques like text localization, optical character recognition, and boundary tracing. Deshpande *et al.* [48] proposed a novel method for chart parsing which adopts a question answering approach to query key elements in charts. The organized elements are subsequently fed into the caption generation module [144], [177] to output captions. A common shortcoming of the models above is the demand for manually designed procedures or features. For example, Al-Zaidy *et al.* [8] relied on pre-defined templates to generate sentences, and Chart-to-Text [177] needs additional annotated tabular data to describe elements in a given chart. Benefiting from the development of deep learning, several end-to-end models have been proposed recently [177], [184], [223]. FigJAM [184] employs ResNet-50 [76] to encode a chart as a whole image and uses OCR to encode text to generate slot values. They are along with the image feature vectors as initialization of a LSTM network to generate captions. Spreafico *et al.* [223] exploited the encoder-decoder LSTM architecture to take time-series data as input and generate corresponding captions. Obeid *et al.* [177] applied a transformer-based model [248] for generating chart summaries. Most recently,
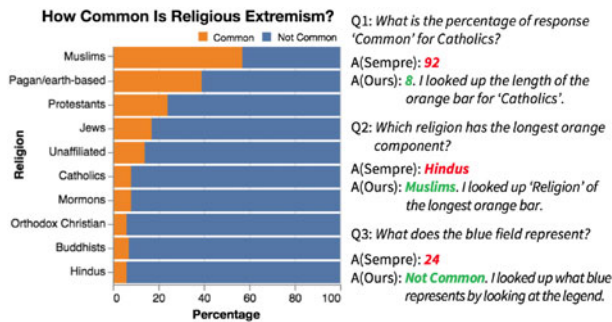
Fig. 14. The automatic VQA pipeline [110] answers the three questions correctly (marked in green) and gives correct explanations.

Kim *et al.* [111] explored how readers gather takeaways when considering charts and captions together. Results suggest that the takeaways differ when the caption mentions visual features of differing prominence levels, which provides valuable guidance for future research.

Discussion. So far, most related works in NLG4Vis can only apply to simple charts (e.g., bar charts, scatterplots, and line charts). Further research can pay attention to improving the chart type coverage, as well as the language quality and descriptive ability. Besides, a more fundamental direction is to develop large datasets covering diverse domains and chart types.

10.4 Visual Question Answering

Visual Question Answering (VQA) is a semantic understanding task that aims to answer questions based on a given visualization and possibly along with the informative text. In this subsection, we only focus on images with charts or infographics rather than images with natural objects and scenes in the computer vision community. Generally, visualizations and questions are respectively encoded and then fused to generate answers [158]. For visualizations, Kim *et al.* [110] leveraged a semantic parsing model, Sempre [282], to develop an automatic chart question answering pipeline with visual explanations describing how the answer was produced. Fig. 14 shows that the pipeline answers all three questions correctly (marked in green) and gives correct explanations of how it obtains the answer. While for infographics, the OCR module additionally needs to identify and tokenize the text, which serves as facts for answer generation [30], [97], [98], [158], [222]. Later models employ more sophisticated fusion modules, among which the attention mechanism has reached great success. LEAF-Net [30] comprises chart parsing, question, and answer encoding according to chart elements followed by an attention network is proposed. STL-CQA [222] applies a structural transformer-based learning approach that emphasizes exploiting the structural properties of charts. Recently, with the development of multimodal transformer [140], [237], Mathew *et al.* [158] proposed a pipeline where the question and infographic are mapped into the same vector space and simply added as the input to stacks of transformer layers. Besides, they also delivered a summary of VQA datasets for reference.

Discussion. Some systems extend existing QA engines (e.g., Sempre) to answer questions about charts; as a result, the interpretation ability is limited as they are not designed for visualization. A more proper way is to develop specific

models based on visualization datasets. However, the range of questions that these learning-based approaches can handle may be restricted to the diversity of datasets. So it may be promising to transfer existing models to other areas and iteratively extend the dataset.

11 RESEARCH OPPORTUNITY

By conducting a comprehensive survey under the guidance of an information visualization pipeline proposed by Card *et al.* [28], we found that, as a rapidly growing field, V-NLI faces many thorny challenges as well as open research opportunities. In this section, we organize these from a macro-perspective into five aspects: knowledge, model, interaction, presentation, and dataset, with additional discussions on applications. Our target is to cover the critical gaps and emerging topics that deserve more attention.

11.1 Domain Knowledge

A major limitation of most existing V-NLI systems is the absence of domain knowledge. We have conducted an evaluation on four state-of-the-art open-source V-NLIs, both academic [174], [273] and commercial [2], [5]. We found that none of the systems can recognize that *nitrogen dioxide* and NO_2 have the same meaning, nor can they recognize the relationship between *age* and *birth year*. Therefore, the support of domain knowledge is crucial, especially for extracting data attributes in the NL query. CogNet [249], a knowledge base dedicated to integrating various existing knowledge bases (e.g., FrameNet, YAGO, Freebase, DBpedia, Wikidata, and ConceptNet), may be useful for broadening the repertoire of utterances supported. Specifically, these knowledge bases can be wholly or partly used to create word embeddings. With the help of embedding results, the similarity between N-grams of the NL query and data attributes names can be computed at both the syntactic and semantic levels. Naturally, it is conceivable that data attributes with high similarity or whose similarity exceeds a certain threshold will be extracted.

11.2 NLP Model

11.2.1 Application of More Advanced NLP Models

The performance of V-NLI depends to a great extent on NLP models. As shown in Table 3 (column *NLP Toolkit or Technology* and column *Recommendation Algorithm*), most of the existing V-NLI systems just apply hand-crafted grammar rules or typical NLP toolkits for convenience. Recently, several state-of-the-art NLP models have reached human performance in specific tasks, such as ELMO [179], BERT [49], GPT-3 [25], and CPM-2 [283]. Some works have leveraged the advances for data visualization [251], [262]. However, few works have applied them in V-NLI. During our survey, we also found that existing works provided limited support for free-form NL input. To construct a more robust system, a promising direction is to apply more advanced NLP models to learn universal grammar patterns from a large corpus of conversations in visual analysis. In the process, we believe that the existing high-quality text datasets will help train and evaluate robust NLP models for V-NLI. Besides, the community can pay more attention to the end-

to-end approach due to its increased efficiency, cost cutting, ease of learning, and smarter visualization inference ability (e.g., chart type selection, data insight mining, and missing data prediction). To this end, advanced neural machine translation models [43] can be applied with various optimization schemes. NLP models of multiple languages also deserve attention since all V-NLI systems we found can only support English. In fact, numerous NLP models can deal effectively with other languages (e.g., Chinese, French, and Spanish) and can be leveraged to develop V-NLI to serve more people who speak different languages.

11.2.2 Deep Interpretation of Dataset Semantics

Semantic information plays an important role in V-NLI. The state-of-the-art systems have considered leveraging semantic parsing toolkits like SEMPRE [282] to parse NL queries [110], [273]. However, just considering the semantics of the NL query is limited, and the semantics of the dataset should also be taken into consideration. The existing technologies for data attribute matching only confine to the letter-matching level but do not go deep into the semantic-matching level, as described in Section 4.3. For example, when the analyzed dataset is about movies, a full-fledged system should be able to recognize that the *Name* attribute in the dataset refers to the movie name and automatically associate it with other attributes appearing in the query. A promising way to augment semantic interpretation ability is to connect with recent semantic data type detection models for visualization like Sherlock [90], Sato [278], ColNet [31], DCOM [156], EXACTA [267], and Doduo [238]. Incorporating such connections will help better infer attribute types upon initialization and may also reduce the need for developers to manually configure attribute aliases. Besides, the aforementioned models are limited to a fixed set of semantic types. An additional essential direction is to extend the existing semantic data type detection models to support more semantic types. Additionally, with the contextually deep interpretation of dataset semantics, supporting queries for multi-table schema would be an interesting research hotspot.

11.3 Interaction

11.3.1 Richer Prompt to Improve System Discoverability

The user may not be aware of what input is valid and what chart types are supported by the system with open-ended textboxes. As described in Section 8.2, discoverability has received relatively little attention in current V-NLIs compared to other characteristics. The existing practices have revealed that real-time prompts can effectively help the user better understand system features and correct input errors in time when typing queries [206], [224], [273] (see Table 3 (column *Autocompletion*)). The most common method is template-based text autocompletion [273]. However, this method is relatively monotonous in practice and does not work for spoken commands. Several works offer the user prompt widgets of data previews to speed up the user's input speed [206], but the supporting prompt widget types are limited. We believe that richer interactive widgets with prompts and multimodal synergistic interaction can greatly improve the system's

usability. Showing provenance of prompt behavior can also enhance the interpretability of visualization results. Concerning this issue, Setlur *et al.* [206] conducted a crowdsourcing study regarding the efficacy of autocompletion suggestions. The insights drawn from the studies are of great value to inspire the future design of V-NLI.

11.3.2 Take Advantage of the User's Interaction History

The intent behind NL queries can be satisfied by various types of charts. These visualization results are too broad that existing systems can hardly account for varying user preferences. Although several conversational V-NLI systems have been proposed [83], [121], [205] to analyze NL queries in context, few systems have taken the user's interaction history into account. Recently, Zehrung *et al.* [276] conducted a crowdsourcing study analyzing trust in humans versus algorithmically generated visualization recommendations. Based on the results, they suggested that the recommendation system should be customized according to the specific user's information search strategy. Personalized information derived from historical user interaction and context can provide a richer model to satisfy the user's analytic tasks. A large number of innovative models in the recommendation system [281] can also be applied for reference. Besides, Lee *et al.* [130] recently deconstructed categorization in visualization recommendation. Several works [79], [209] have studied the associations and expectations about verbalization and visualization reported by users. Integrating the information for further modeling of the user's interaction history is another interesting research topic.

11.4 Presentation

We hypothesize that we would not truly have succeeded in democratizing access to visual analysis through natural language until we design systems that use NL both as input and output modality. For this vision, a promising direction is to combine all the aforementioned characteristics (e.g., VQA, NLG, and annotation) to design a hybrid system. In addition, existing V-NLI systems mostly only support 2D visualization. Nowadays, Immersive Analytics (IA) is a quickly evolving field that leverages immersive technologies for data analysis [56], [60]. In the visualization community, several works have augmented static visualizations with virtual content [35], [124]. Reinders *et al.* [189] studied blind and low vision people's preferences when exploring interactive 3D printed models (I3Ms). However, there have been no systems that support NLI for data visualization in an immersive way. A related work for reference is that Lee *et al.* [125] proposed the concept of *data visceralization* and introduced a conceptual pipeline. It would be interesting to enrich this pipeline with the integration of V-NLI. Besides, as described in Section 7, view transformations are rarely involved in existing systems. With immersive technologies, more view transformations can be explored and integrated into V-NLI to provide an immersive interactive experience.

11.5 Dataset

There is a widely recognized consensus that large-scale data collection is an effective way to facilitate community

TABLE 8
Summary of Existing V-NLI Datasets

Name	Publication	NL queries	Data tables	Benchmark	Other Contributions	Website
Kim *et al.* [110]	CHI'20	629	52		VQA with explanations	https://github.com/dhkim16/VisQA-release
Quda [61]	arXiv'20	14,035	36		Three Quda applications	https://freenli.github.io/quda/
NLV [228]	CHI'21	893	3	✓	Characterization of utterances	https://nlvcorpus.github.io/
nvBench [151]	SIGMOD'21	25,750	780	✓	NL2SQL-to-NL2VIS and SEQ2VIS	https://github.com/TsinghuaDatabaseGroup/nvBench

development (e.g., ImageNet [47] for image processing and GEM [66] for natural language generation). Although there are various datasets for general NLP tasks, they can hardly be directly applied to provide training samples for V-NLI models. In the visualization community, several works have begun to collect large datasets for V-NLI, as shown in Table 8. To assist the deployment of learning-based techniques for parsing human language, Fu *et al.* [61] proposed Quda, containing diverse user queries annotated with analytic tasks. Srinivasan *et al.* [228] conducted an online study to collect NL utterances and characterized them based on their phrasing type. VizNet [85] is a step forward in addressing the need for large-scale data and visualization repositories. Kim *et al.* [110] collected questions people posed about various bar charts and line charts, along with their answers and explanations to the questions. Fully considering the user's characteristics, Kassel *et al.* [104] introduced a linguistically motivated two-dimensional answer space that varies in the level of both support and information to match the human language in visual analysis. The major limitation of them is that the query types contained are not rich enough. Besides, unfortunately, a benchmark or the ground truth of a given dataset is usually unavailable. Although Srinivasan *et al.* [228] made an initial attempt to present a collection of NL utterances with the mapping visualizations, Luo *et al.* [151] synthesized NL-to-Vis benchmarks by piggybacking NL-to-SQL benchmarks and produced a NL-to-Vis benchmark, nvBench, the supporting visualization types and tasks are limited. Therefore, collecting large-scale datasets and creating new benchmarks that support more tasks, domains, and interactions will be an indispensable research direction in the future.

11.6 Application

Apart from facilitating data analysis, much attention has been paid to benefit real-world applications by integrating visual analysis and natural language interface. Huang *et al.* [88] proposed a query engine to convert, store and retrieve spatial uncertain mobile trajectories via intuitive NL input. Leo John *et al.* [133] proposed a novel V-NLI to promote medical imaging and biomedical research. With an audio travel podcast as input, Crosscast [266] identifies geographic locations and descriptive keywords within the podcast transcript through NLP and text mining techniques. The information is later used to select relevant photos from online repositories and synchronize their display to align with the audio narration. MeetingVis [214] leverages ASR and NLP techniques to promote effective meeting summaries in team-based workplaces. PathViewer [255] leverages ideas from flow diagrams and NLP to visualize the sequences of intermediate steps that students take. Since V-NLI

can be easily integrated as a module into the visualization system, more applications can be explored in the future.

12 CONCLUSION

The past two decades have witnessed the rapid development of visualization-oriented natural language interfaces, which act as a complementary input modality for visual analytics. However, the community lacks a comprehensive survey of related works to guide follow-up research. We fill the gap, and the resulting survey gives a comprehensive overview of what characteristics are currently concerned and supported by V-NLI. We also propose several promising directions for future work. To our knowledge, this paper is the first step towards reviewing V-NLI in a novel and systematic manner. We hope that this paper can better guide future research and encourage the community to think further about NLI for data visualization.

REFERENCES

[1] Apache OpenNLP, 2017. [Online]. Available: http://opennlp.apache.org/

[2] Ask data, 2019. [Online]. Available: https://www.tableau.com/products/new-features/ask-data

[3] Google NLP, 2018. [Online]. Available: https://cloud.google.com/natural-language/

[4] IBM Watson Analytics, 2014. [Online]. Available: http://www.ibm.com/analytics/watson-analytics

[5] Microsoft Power BI, 2018. [Online]. Available: https://docs.microsoft.com/en-us/power-bi/create-reports/power-bi-tutorial-q-and-a

[6] K. Affolter, K. Stockinger, and A. Bernstein, "A comparative survey of recent natural language interfaces for databases," *VLDB J.*, vol. 28, no. 5, pp. 793–819, 2019.

[7] A. Akbik, D. Blythe, and R. Vollgraf, "Contextual string embeddings for sequence labeling," in *Proc. 27th Int. Conf. Comput. Linguistics*, 2018, pp. 1638–1649.

[8] R. A. Al-Zaidy and C. L. Giles, "Automatic extraction of data from bar charts," in *Proc. 8th Int. Conf. Knowl. Capture*, 2015, pp. 1–4.

[9] R. Amar, J. Eagan, and J. Stasko, "Low-level components of analytic activity in information visualization," in *Proc. IEEE Symp. Informat. Vis.*, 2005, pp. 111–117.

[10] F. Bacci, F. M. Cau, and L. D. Spano, "Inspecting data using natural language queries," in *Proc. Int. Conf. Comput. Sci. Appl.*, 2020, vol. 12254, pp. 771–782.

[11] B. Bach *et al.*, "Design patterns for data comics," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2018, pp. 1–12.

[12] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–15.

[13] C. Baik, H. V. Jagadish, and Y. Li, "Bridging the semantic gap with SQL query logs in natural language interfaces to databases," in *Proc. Int. Conf. Data Eng.*, 2019, pp. 374–385.

[14] F. Basik et al., "DBPal: A learned NL-interface for databases," in *Proc. Int. Conf. Manage. Data*, 2018, pp. 1765–1768.

[15] H. Bast and E. Haussmann, "More accurate question answering on freebase," in *Proc. ACM Int. Conf. Informat. Knowl. Manage.*, 2015, pp. 1431–1440.

[16] L. Battle, R. J. Crouser, A. Nakeshimana, A. Montoly, R. Chang, and M. Stonebraker, "The role of latency and task complexity in predicting visual search behavior," *IEEE Trans. Vis. Comput. Graphics*, vol. 26, no. 1, pp. 1246–1255, Jan. 2020.

[17] L. Battle and C. Scheidegger, "A structured review of data management technology for interactive visualization and analysis," *IEEE Trans. Vis. Comput. Graphics*, vol. 27, no. 2, pp. 1128–1138, Feb. 2021.

[18] Y. Belinkov and J. Glass, "Analysis methods in neural language processing: A survey," *Trans. Assoc. Comput. Linguistics*, vol. 7, pp. 49–72, 2019.

[19] S. Bergamaschi et al., "Combining user and database perspective for solving keyword queries over relational databases," *Inf. Syst.*, vol. 55, pp. 1–19, 2016.

[20] S. Bieliauskas and A. Schreiber, "A conversational user interface for software visualization," in *Proc. IEEE Work. Conf. Softw. Vis.*, 2017, pp. 139–143.

[21] S. Bird, "NLTK: The natural language toolkit," in *Proc. ACL Interactive Poster Demonstration Sessions*, 2006, pp. 214–217.

[22] L. Blunschi et al., "SODA: Generating SQL for business users," *Proc. VLDB Endowment*, vol. 5, no. 10, pp. 932–943, 2012.

[23] M. Bostock, V. Ogievetsky, and J. Heer, "D3: Data-driven documents," *IEEE Trans. Vis. Comput. Graphics*, vol. 17, no. 12, pp. 2301–2309, Dec. 2011.

[24] M. Brehmer and T. Munzner, "A multi-level typology of abstract visualization tasks," *IEEE Trans. Vis. Comput. Graphics*, vol. 19, no. 12, pp. 2376–2385, Dec. 2013.

[25] T. Brown et al., "Language models are few-shot learners," in *Proc. Adv. Neural Informat. Process. Syst.*, 2020, pp. 1877–1901.

[26] C. Bryan, K. L. Ma, and J. Woodring, "Temporal summary images: An approach to narrative visualization via interactive annotation generation and placement," *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 1, pp. 511–520, Jan. 2017.

[27] Z. Bylinskii et al., "Understanding infographics through textual and visual tag prediction," 2017, *arXiv:1709.09215*.

[28] S. Card, J. Mackinlay, and B. Shneiderman, *Readings in Information Visualization: Using Vision to Think*. San Mateo, CA, USA: Morgan Kaufmann, 1999.

[29] A. Celikyilmaz, E. Clark, and J. Gao, "Evaluation of text generation: A survey," pp. 1–75, 2021, *arXiv:2006.14799*.

[30] R. Chaudhry et al., "LEAF-QA: Locate, encode attend for figure question answering," in *Proc. Winter Conf. Appl. Comput. Vis.*, 2020, pp. 3501–3510.

[31] J. Chen, E. Jiménez-Ruiz, I. Horrocks, and C. Sutton, "ColNet: Embedding the semantics of web tables for column type prediction," in *Proc. 33rd AAAI Conf. Artif. Intell.*, 2019, pp. 29–36.

[32] S. Chen et al., "Supporting story synthesis: Bridging the gap between visual analytics and storytelling," *IEEE Trans. Vis. Comput. Graphics*, vol. 26, no. 7, pp. 2499–2516, Jul. 2020.

[33] Y. Chen, S. Barlowe, and J. Yang, "Click2Annotate: Automated insight externalization with rich semantics," in *Proc. IEEE Symp. Vis. Anal. Sci. Technol.*, 2010, pp. 155–162.

[34] Y. Chen, J. Yang, S. Barlowe, and D. H. Jeong, "Touch2Annotate: Generating better annotations with less human effort on multi-touch interfaces," in *Proc. Extended Abstract Hum. Factors Comput. Syst.*, 2010, pp. 3703–3708.

[35] Z. Chen et al., "Augmenting static visualizations with PapARVis designer," in *Proc. Extended Abstract Hum. Factors Comput. Syst.*, 2020, pp. 1–12.

[36] Z. Chen et al., "Towards automated infographic design: Deep learning-based auto-extraction of extensible timeline," *IEEE Trans. Vis. Comput. Graphics*, vol. 26, no. 1, pp. 917–926, Jan. 2020.

[37] J. Choo et al., "VisIRR: Visual analytics for information retrieval and recommendation with large-scale document data," in *Proc. IEEE Symp. Vis. Anal. Sci. Technol.*, 2014, pp. 243–244.

[38] K. Cook et al., "Mixed-initiative visual analytics using task-driven recommendations," in *Proc. IEEE Symp. Vis. Anal. Sci. Technol.*, 2015, pp. 9–16.

[39] M. Corio and G. Lapalme, "Generation of texts for information graphics," in *Proc. Eur. Workshop Natural Lang. Gener.*, 1999, pp. 49–58.

[40] K. Cox et al., "A multi-modal natural language interface to an information visualization environment," *Int. J. Speech Technol.*, vol. 4, no. 3, pp. 297–314, 2001.

[41] W. Cui et al., "Text-to-Viz: Automatic generation of infographics from proportion-related natural language statements," *IEEE Trans. Vis. Comput. Graphics*, vol. 26, no. 1, pp. 906–916, Jan. 2020.

[42] Z. Cui, S. K. Badam, M. A. Yalçin, and N. Elmqvist, "DataSite: Proactive visual data exploration with computation of insight-based recommendations," *Informat. Vis.*, vol. 18, no. 2, pp. 251–267, 2019.

[43] R. Dabre, C. Chu, and A. Kunchukuttan, "A survey of multilingual neural machine translation," *ACM Comput. Surv.*, vol. 53, no. 5, pp. 1–38, 2020.

[44] S. Demir, S. Carberry, and K. F. McCoy, "Generating textual summaries of bar charts," in *Proc. Int. Natural Lang. Generation Conf.*, 2008, pp. 7–15.

[45] S. Demir, S. Carberry, and K. F. McCoy, "Summarizing information graphics textually," *Comput. Linguistics*, vol. 38, no. 3, pp. 527–574, 2012.

[46] Ç. Demiralp, P. J. Haas, S. Parthasarathy, and T. Pedapati, "Foresight: Recommending visual insights," *VLDB Endowment*, vol. 10, no. 12, pp. 1937–1940, 2017.

[47] J. Deng et al., "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[48] A. P. Deshpande and C. N. Mahender, "Summarization of graph using question answer approach," in *Proc. Adv. Intell. Syst. Comput.*, 2020, pp. 205–216.

[49] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2019, pp. 4171–4186.

[50] K. Dhamdhere et al., "Analyza: Exploring data with conversation," in *Proc. Int. Conf. Intell. User Interfaces*, 2017, pp. 493–504.

[51] V. Dibia and C. Demiralp, "Data2Vis: Automatic generation of data visualizations using sequence-to-sequence recurrent neural networks," *IEEE Comput. Graphics Appl.*, vol. 39, no. 5, pp. 33–46, Sep./Oct. 2019.

[52] E. Dimara and C. Perin, "What is Interaction for Data Visualization?," *IEEE Trans. Vis. Comput. Graphics*, vol. 26, no. 1, pp. 119–129, Jan. 2020.

[53] R. Ding et al., "Quickinsights: Quick and automatic discovery of insights from multi-dimensional data," in *Proc. Int. Conf. Manage. Data*, 2019, pp. 317–332.

[54] I. K. Duncan, S. Tingsheng, S. T. Perrault, and M. T. Gastner, "Task-based effectiveness of interactive contiguous area cartograms," *IEEE Trans. Vis. Comput. Graphics*, vol. 27, no. 3, pp. 2136–2152, Mar. 2021.

[55] J. Eisenschlos, S. Krichene, and T. Müller, "Understanding tables with intermediate pre-training," in *Proc. Find. Assoc. Comput. Linguistics*, 2020, pp. 281–296.

[56] B. Ens et al., "Grand Challenges in Immersive Analytics," in *Proc. Extended Abstr. Hum. Factors Comput. Syst.*, 2021, pp. 1–17.

[57] M. Fasciano and G. Lapalme, "PostGraphe: A system for the generation of statistical graphics and text Important factors in the generation process," in *Proc. Int. Natural Lang. Generation Conf.*, 1996, pp. 51–60.

[58] E. Fast et al., "Iris: A conversational agent for complex tasks," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2018, pp. 1–12.

[59] L. Ferres, G. Lindgaard, and L. Sumegi, "Evaluating a tool for improving accessibility to charts and graphs," in *Proc. Int. ACM SIGACCESS Conf. Comput. Accessibility*, 2010, pp. 83–90.

[60] A. Fonnet and Y. Prie, "Survey of immersive analytics," *IEEE Trans. Vis. Comput. Graphics*, vol. 27, no. 3, pp. 2101–2122, Mar. 2021.

[61] S. Fu et al., "Quda: Natural language queries for visual data analytics," 2020, *arXiv:2005.03257*.

[62] J. Fulda, M. Brehmel, and T. Munzner, "TimeLineCurator: Interactive authoring of visual timelines from unstructured text," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 1, pp. 300–309, Jan. 2016.

[63] T. Gao et al., "DataTone: Managing ambiguity in natural language interfaces for data visualization," in *Proc. Annu. ACM Symp. User Interface Softw. Technol.*, 2015, pp. 489–500.

[64] T. Gao et al., "NewsViews: An automated pipeline for creating custom geovisualizations for news," in Proc. Extended Abstr. Hum. Factors Comput. Syst., 2014, pp. 3005–3014.

[65] T. Ge, B. Lee, and Y. Wang, "CAST: Authoring data-driven chart animations," in Proc. Extended Abstr. Hum. Factors Comput. Syst., 2021, pp. 1–15.

[66] S. Gehrmann et al., "The GEM benchmark: Natural language generation, its evaluation and metrics," in Proc. Workshop Natural Lang. Gener. Eval. Metrics, 2021, pp. 96–120.

[67] A. Ghosh et al., "A comprehensive review of tools for exploratory analysis of tabular industrial datasets," Vis. Inform., vol. 2, no. 4, pp. 235–253, 2018.

[68] M. Gingerich and C. Conati, "Constructing models of user and task characteristics from eye gaze data for user-adaptive information highlighting," in Proc. AAAI Conf. Artif. Intell., 2015, pp. 1728–1734.

[69] I. Goodfellow et al., "Generative adversarial nets," in Proc. Int. Conf. Neural Informat. Process. Syst., 2014, pp. 2672–2680.

[70] D. Gotz and Z. Wen, "Behavior-driven visualization recommendation," in Proc. Int. Conf. Intell. User Interfaces, 2009, pp. 315–324.

[71] B. J. Grosz and C. L. Sidner, "Attention, intentions, and the structure of discourse," Comput. Linguistics, vol. 12, no. 3, pp. 175–204, 1986.

[72] J. Gu, Z. Lu, H. Li, and V. O. Li, "Incorporating copying mechanism in sequence-to-sequence learning," in Proc. Annu. Meeting Assoc. Comput. Linguistics, 2016, pp. 1631–1640.

[73] J. Guo et al., "Towards complex text-to-SQL in cross-domain database with intermediate representation," in Proc. Annu. Meeting Assoc. Comput. Linguistics, 2019, pp. 4524–4535.

[74] I. Gur, S. Yavuz, Y. Su, and X. Yan, "DialSQL: Dialogue based structured query generation," in Proc. Annu. Meeting Assoc. Comput. Linguistics, 2018, pp. 1339–1349.

[75] C. Harris et al., "Insight-centric visualization recommendation," 2021, arXiv:2103.11297.

[76] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 770–778.

[77] M. Hearst and M. Tory, "Would you like a chart with that? Incorporating visualizations into conversational interfaces," in Proc. IEEE Visualization Conf., 2019, pp. 1–5.

[78] M. Hearst, M. Tory, and V. Setlur, "Toward interface defaults for vague modifiers in natural language interfaces for visual analysis," in Proc. IEEE Visualization Conf., 2019, pp. 21–25.

[79] R. Henkin and C. Turkay, "Words of estimative correlation: Studying verbalizations of scatterplots," IEEE Trans. Vis. Comput. Graphics, vol. 14, no. 8, pp. 1–15, Sep. 2020.

[80] J. Herzig et al., "TaPas: Weakly supervised table parsing via pre-training," in Proc. Annu. Meeting Assoc. Comput. Linguistics, 2020, pp. 4320–4333.

[81] M. Honnibal and I. Montani, "spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing," 2017.

[82] A. K. Hopkins, M. Correll, and A. Satyanarayan, "VisuaLint: Sketchy in situ annotations of chart construction errors," Comput. Graph. Forum, vol. 39, no. 3, pp. 219–228, 2020.

[83] E. Hoque, V. Setlur, M. Tory, and I. Dykeman, "Applying pragmatics principles for interaction with visual analytics," IEEE Trans. Vis. Comput. Graphics, vol. 24, no. 1, pp. 309–318, Jan. 2018.

[84] K. Hu et al., "VizML: A machine learning approach to visualization recommendation," in Proc. Extended Abstr. Hum. Factors Comput. Syst., 2019, pp. 1–12.

[85] K. Hu et al., "VizNet: Towards a large-scale visualization learning and benchmarking repository," in Proc. Extended Abstr. Hum. Factors Comput. Syst., 2019, pp. 1–12.

[86] K. Hu, D. Orghian, and C. Hidalgo, "DIVE: A mixed-initiative system supporting integrated data exploration workflows," in Proc. Workshop Hum. Loop Data Anal., 2018, pp. 1–7.

[87] B. Huang, G. Zhang, and P. C.-Y. Sheu, "A natural language database interface based on a probabilistic context free grammar," in Proc. IEEE Int. Workshop Semantic Comput. Syst., 2008, pp. 155–162.

[88] Z. Huang et al., "A natural-language-based visual query approach of uncertain human trajectories," IEEE Trans. Vis. Comput. Graphics, vol. 26, no. 1, pp. 1–11, Jan. 2019.

[89] J. Hullman, N. Diakopoulos, and E. Adar, "Contextifier: Automatic generation of annotated stock visualizations," in Proc. Extended Abstr. Hum. Factors Comput. Syst., 2013, pp. 2707–2716.

[90] M. Hulsebos et al., "Sherlock: A deep learning approach to semantic data type detection," in Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2019, pp. 1500–1508.

[91] R. Ingria et al., "TimeML: Robust specification of event and temporal expressions in text," New Directions Question Answering, vol. 3, pp. 28–34, 2003.

[92] S. Iyer et al., "Learning a neural semantic parser from user feedback," in Proc. Annu. Meeting Assoc. Comput. Linguistics, 2017, pp. 963–973.

[93] Q. Jiang, G. Sun, Y. Dong, and R. Liang, "DT2VIS: A focus+context answer generation system to facilitate visual exploration of tabular data," IEEE Comput. Graph. Appl., vol. 41, no. 5, pp. 45–56, Sep./Oct. 2021.

[94] S. Jiang et al., "Emerging wearable interfaces and algorithms for hand gesture recognition: A survey," IEEE Rev. Biomed. Eng., vol. 15, pp. 85–102, 2022.

[95] M. Joshi et al., "SpanBERT: Improving pre-training by representing and predicting spans," Trans. Assoc. Comput. Linguistics, vol. 8, pp. 64–77, 2020.

[96] M. Joshi, O. Levy, L. Zettlemoyer, and D. Weld, "BERT for coreference resolution: Baselines and analysis," in Proc. Conf. Empir. Methods Natural Lang. Process., 2019, pp. 5803–5808.

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

[98] S. E. Kahou et al., "FigureQA: An annotated figure dataset for visual reasoning," in Proc. Int. Conf. Learn. Representations, 2018, pp. 1–20.

[99] S. Kandel, A. Paepcke, J. Hellerstein, and J. Heer, "Wrangler: Interactive visual specification of data transformation scripts," in Proc. Extended Abstr. Hum. Factors Comput. Syst., 2011, pp. 3363–3372.

[100] S. Kandel et al., "Profiler: Integrated statistical analysis and visualization for data quality assessment," in Proc. Int. Work. Conf. Adv. Vis. Interfaces, 2012, pp. 547–554.

[101] E. Kandogan, "Just-in-time annotation of clusters, outliers, and trends in point-based data visualizations," in Proc. IEEE Symp. Vis. Anal. Sci. Technol., 2012, pp. 73–82.

[102] J. Kang et al., "Understanding how people use natural language to ask for recommendations," in Proc. ACM Conf. Recommender Syst., 2017, pp. 229–237.

[103] J. F. Kassel and M. Rohs, "Valletto: A multimodal interface for ubiquitous visual analytics," in Proc. Extended Abstr. Hum. Factors Comput. Syst., 2018, pp. 1–6.

[104] J. F. Kassel and M. Rohs, "Talk to me intelligibly: Investigating an answer space to match the user's language in visual analysis," in Proc. Designing Interactive Syst. Conf., 2019, pp. 1517–1529.

[105] T. Kato, M. Matsushita, and E. Maeda, "Answering it with charts: Dialogue in natural language and charts," in Proc. Int. Conf. Comput. Linguistics, 2002, pp. 1–7.

[106] P. Kaur, M. Owonibi, and B. Koenig-Ries, "Towards visualization recommendation-a semi-automated domain-specific learning approach," in Proc. GI-Workshop Grundlagen von Datenbanken, 2015, pp. 30–35.

[107] S. Kerpedjiev, G. Carenini, S. F. Roth, and J. D. Moore, "AutoBrief: A multimedia presentation system for assisting data analysis," Comput. Stand. Interfaces, vol. 18, no. 6/7, pp. 583–593, 1997.

[108] N. Kerracher and J. Kennedy, "Constructing and evaluating visualisation task classifications: Process and considerations," Comput. Graph. Forum, vol. 36, no. 3, pp. 47–59, 2017.

[109] A. Key, B. Howe, D. Perry, and C. Aragon, "VizDeck: Self-organizing dashboards for visual analytics," in Proc. Int. Conf. Manage. Data, 2012, pp. 681–684.

[110] D. H. Kim, E. Hoque, and M. Agrawala, "Answering questions about charts and generating visual explanations," in Proc. Extended Abstr. Hum. Factors Comput. Syst., 2020, pp. 1–13.

[111] D. H. Kim, V. Setlur, and M. Agrawala, "Towards understanding how readers integrate charts and captions: A case study with line charts," in Proc. Extended Abstr. Hum. Factors Comput. Syst., 2021, pp. 1–11.

[112] H. Kim et al., "Thumbnails for data stories: A survey of current practices," in Proc. IEEE Visualization Conf., 2019, pp. 116–120.

[113] Y. Kim and J. Heer, "Assessing effects of task and data distribution on the effectiveness of visual encodings," Comput. Graph. Forum, vol. 37, no. 3, pp. 157–167, 2018.

[114] Y. Kim and J. Heer, "Gemini: A grammar and recommender system for animated transitions in statistical graphics," IEEE Trans. Vis. Comput. Graphics, vol. 27, no. 2, pp. 485–494, Feb. 2021.

[115] Y.-H. Kim, B. Lee, A. Srinivasan, and E. K. Choe, "Data@Hand: Fostering visual exploration of personal data on smartphones leveraging speech and touch interaction," in *Proc. Extended Abstr. Hum. Factors Comput. Syst.*, 2021, pp. 1–17.

[116] R. Kincaid and G. Pollock, "Nicky: Toward a virtual assistant for test and measurement instrument recommendations," in *Proc. IEEE Int. Conf. Semantic Comput.*, 2017, pp. 196–203.

[117] Y. Kirstain, O. Ram, and O. Levy, "Coreference resolution without span representations," 2021, *arXiv:2101.00434*.

[118] V. Kocijan et al., "A surprisingly robust trick for the winograd schema challenge," *Proc. 57th Annu. Meet. Assoc. Comput. Linguistics*, 2019, pp. 4837–4842.

[119] N. Kong and M. Agrawala, "Graphical overlays: Using layered elements to aid chart reading," *IEEE Trans. Vis. Comput. Graphics*, vol. 18, no. 12, pp. 2631–2638, Dec. 2012.

[120] A. Kumar et al., "Multimodal coreference resolution for exploratory data visualization dialogue: Context-based annotation and gesture identification," in *Proc. Workshop Semantics Pragmatics Dialogue*, 2017, pp. 41–51.

[121] A. Kumar et al., "Towards a dialogue system that supports rich visualizations of data," in *Proc. Annu. Meeting Special Int. Group Discourse Dialogue*, 2016, pp. 304–309.

[122] C. Lai et al., "Automatic annotation synchronizing with textual description for visualization," in *Proc. Extended Abstr. Hum. Factors Comput. Syst.*, 2020, pp. 1–13.

[123] S. Lalle, D. Toker, and C. Conati, "Gaze-driven adaptive interventions for magazine-style narrative visualizations," *IEEE Trans. Vis. Comput. Graphics*, vol. 27, no. 6, pp. 2941–2952, Jun. 2021.

[124] R. Langner, M. Satkowski, W. Büschel, and R. Dachselt, "MARVIS: Combining mobile devices and augmented reality for visual data analysis," in *Proc. Extended Abstr. Hum. Factors Comput. Syst.*, 2021, pp. 1–17.

[125] B. Lee, D. Brown, B. Lee, C. Hurter, S. Drucker, and T. Dwyer, "Data visceralization: Enabling deeper understanding of data using virtual reality," *IEEE Trans. Vis. Comput. Graphics*, vol. 27, no. 2, pp. 1095–1105, Feb. 2021.

[126] B. Lee, P. Isenberg, N. H. Riche, and S. Carpendale, "Beyond mouse and keyboard: Expanding design considerations for information visualization interactions," *IEEE Trans. Vis. Comput. Graphics*, vol. 18, no. 12, pp. 2689–2698, Dec. 2012.

[127] B. Lee et al., "Task taxonomy for graph visualization," in *Proc. AVI Workshop BEyond Time Errors Novel Eval. Methods Informat. Visualization*, 2006, pp. 1–5.

[128] B. Lee, A. Srinivasan, P. Isenberg, and J. Stasko, "Post-WIMP interaction for information visualization," *Found. Trends Hum.-Comput. Interaction*, vol. 14, no. 1, pp. 1–95, 2021.

[129] D. J. L. Lee, A. Quamar, E. Kandogan, and F. Özcan, "Boomerang: Proactive insight-based recommendations for guiding conversational data analysis," in *Proc. Int. Conf. Manage. Data*, 2021, pp. 2750–2754.

[130] D. J.-L. Lee, V. Setlur, M. Tory, K. G. Karahalios, and A. Parameswaran, "Deconstructing categorization in visualization recommendation: A taxonomy and comparative study," *IEEE Trans. Vis. Comput. Graphics*, to be published, doi: 10.1109/TVCG.2021.3085751.

[131] K. Lee, L. He, M. Lewis, and L. Zettlemoyer, "End-to-end neural coreference resolution," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2017, pp. 188–197.

[132] K. Lee, L. He, and L. Zettlemoyer, "Higher-order coreference resolution with coarse-to-fine inference," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2018, pp. 687–692.

[133] R. J. Leo John et al., "A natural language interface for dissemination of reproducible biomedical data science," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Interv.*, 2018, vol. 11073, pp. 197–205.

[134] R. J. LeoJohn, N. Potti, and J. M. Patel, "Ava: From data to insights through conversation," in *Proc. Biennial Conf. Innov. Data Syst. Res.*, 2017, pp. 1–10.

[135] F. Li and H. V. Jagadish, "NaLIR: An interactive natural language interface for querying relational databases," in *Proc. Int. Conf. Manage. Data*, 2014, pp. 709–712.

[136] F. Li and H. V. Jagadish, "Constructing an interactive natural language interface for relational databases," *Proc. VLDB Endowment*, vol. 8, no. 1, pp. 73–84, 2014.

[137] F. Li and H. V. Jagadish, "Understanding natural language queries over relational databases," *ACM SIGMOD Rec.*, vol. 45, no. 1, pp. 6–13, 2016.

[138] H. Li, Y. Wang, S. Zhang, Y. Song, and H. Qu, "KG4Vis: A knowledge graph-based approach for visualization recommendation," *IEEE Trans. Vis. Comput. Graphics*, vol. 28, no. 1, pp. 195–205, Jan. 2022.

[139] J. Li et al., "LayoutGAN: Generating graphic layouts with wireframe discriminators," 2019, *arXiv:1901.06767*.

[140] L. H. Li et al., "VisualBERT: A simple and performant baseline for vision and language," 2019, *arXiv:1908.03557*.

[141] P. Li et al., "Application of hidden markov model in SQL injection detection," in *Proc. Annu. Comput. Softw. Appl. Conf.*, 2017, pp. 578–583.

[142] H. Lin, D. Moritz, and J. Heer, "Dziban: Balancing agency & automation in visualization design via anchored recommendations," in *Proc. Extended Abstr. Hum. Factors Comput. Syst.*, 2020, pp. 1–12.

[143] C. Liu, Y. Han, R. Jiang, and X. Yuan, "ADVISor: Automatic visualization answer for natural-language question on tabular data," in *Proc. IEEE Pacific Visualization Symp.*, 2021, pp. 11–20.

[144] C. Liu, L. Xie, Y. Han, D. Wei, and X. Yuan, "AutoCaption: An approach to generate natural language description from visualization automatically," in *Proc. IEEE Pacific Visualization Symp.*, 2020, pp. 191–195.

[145] Q. Liu et al., "Commonsense knowledge enhanced embeddings for solving pronoun disambiguation problems in Winograd Schema Challenge," 2016, *arXiv:1611.04146*.

[146] T. Liu et al., "Data-driven mark orientation for trend estimation in scatterplots," in *Proc. Extended Abstr. Hum. Factors Comput. Syst.*, 2021, pp. 1–16.

[147] G. López, L. Quesada, and L. A. Guerrero, "Alexa vs. Siri vs. Cortana vs. Google assistant: A comparison of speech-based natural user interfaces," in *Proc. Int. Conf. Appl. Hum. Factors Ergonom.*, 2017, pp. 241–250.

[148] Y. Luo et al., "DeepEye: Creating good data visualizations by keyword search," in *Proc. Int. Conf. Manage. Data*, 2018, pp. 1733–1736.

[149] Y. Luo, X. Qin, N. Tang, and G. Li, "DeepEye: Towards automatic data visualization," in *Proc. Int. Conf. Data Eng.*, 2018, pp. 101–112.

[150] Y. Luo, N. Tang, G. Li, and et al., "Natural language to visualization by neural machine translation," *IEEE Trans. Vis. Comput. Graphics*, vol. 28, no. 1, pp. 217–226, 2021.

[151] Y. Luo et al., "Synthesizing natural language to visualization (NL2VIS) benchmarks from NL2SQL benchmarks," in *Proc. Int. Conf. Manage. Data*, 2021, pp. 1235–1247.

[152] Y. Ma, A. K. H. Tung, W. Wang, and et al., "ScatterNet: A deep subjective similarity model for visual analysis of scatterplots," *IEEE Trans. Vis. Comput. Graphics*, vol. 26, no. 3, pp. 1562–1576, Mar. 2020.

[153] J. Mackinlay, "Automating the design of graphical presentations of relational information," *ACM Trans. Graph.*, vol. 5, no. 2, pp. 110–141, 1986.

[154] J. Mackinlay, P. Hanrahan, and C. Stolte, "Show Me: Automatic presentation for visual analysis," *IEEE Trans. Vis. Comput. Graphics*, vol. 13, no. 6, pp. 1137–1144, Nov./Dec. 2007.

[155] S. Madan et al., "Synthetically trained icon proposals for parsing and summarizing infographics," 2018, *arXiv:1807.10441*.

[156] S. Maji, S. S. Rout, and S. Choudhary, "DCoM: A deep column mapper for semantic data type detection," 2021, *arXiv:2106.12871*.

[157] C. Manning et al., "The Stanford CoreNLP natural language processing toolkit," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2014, pp. 55–60.

[158] M. Mathew et al., "InfographicVQA," 2021, *arXiv:2104.12756*.

[159] M. Matsushita, E. Maeda, and T. Kato, "An interactive visualization method of numerical data based on natural language requirements," *Int. J. Hum. Comput. Stud.*, vol. 60, no. 4, pp. 469–488, 2004.

[160] L. McNabb and R. S. Laramee, "Survey of surveys (SoS) - mapping the landscape of survey papers in information visualization," *Comput. Graph. Forum*, vol. 36, no. 3, pp. 589–617, 2017.

[161] R. Metoyer, B. Lee, N. HenryRiche, and M. Czerwinski, "Understanding the verbal language and structure of end-user descriptions of data visualizations," in *Proc. Extended Abstr. Hum. Factors Comput. Syst.*, 2012, pp. 1659–1662.

[162] R. Metoyer, Q. Zhi, B. Janczuk, and W. Scheirer, "Coupling story to visualization: Using textual analysis as a bridge between data and interpretation," in *Proc. Int. Conf. Intell. User Interfaces*, 2018, pp. 503–507.

[163] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. Int. Conf. Learn. Representations*, 2013, pp. 1–12.

[164] G. A. Miller, "WordNet: A lexical database for English," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[165] M. Mitri, "Story analysis using natural language processing and interactive dashboards," *J. Comput. Inf. Syst.*, pp. 1–11, 2020. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/08874417.2020.1774442

[166] V. O. Mittal, J. D. Moore, G. Carenini, and S. Roth, "Describing complex charts in natural language: A caption generation system," *Comput. Linguistics*, vol. 24, no. 3, pp. 431–467, 1998.

[167] P. Moraes, G. Sina, K. McCoy, and S. Carberry, "Generating summaries of line graphs," in *Proc. Int. Natural Lang. Gener. Conf.*, 2014, pp. 95–98.

[168] D. Moritz, C. Wang, G. L. Nelson, and et al., "Formalizing visualization design knowledge as constraints: Actionable and extensible models in Draco," *IEEE Trans. Vis. Comput. Graphics*, vol. 25, no. 1, pp. 438–448, Jan. 2019.

[169] T. Murillo-Morales and K. Miesenberger, "AUDiaL: A natural language interface to make statistical charts accessible to blind persons," in *Proc. Int. Conf. Comput. Helping People With Special Needs*, Berlin, Germany, Springer, 2020, pp. 373–384.

[170] B. Mutlu, E. Veas, and C. Trattner, "VizRec: Recommending personalized visualizations," *ACM Trans. Interact. Intell. Syst.*, vol. 6, no. 4, pp. 1–39, 2016.

[171] M. Nafari and C. Weaver, "Augmenting visualization with natural language translation of interaction: A usability study," *Comput. Graph. Forum*, vol. 32, no. 3, pp. 391–400, 2013.

[172] M. Nafari and C. Weaver, "Query2Question: Translating visualization interaction into natural language," *IEEE Trans. Vis. Comput. Graphics*, vol. 21, no. 6, pp. 756–769, Jun. 2015.

[173] A. Narechania et al., "DIY: Assessing the correctness of natural language to SQL systems," in *Proc. Int. Conf. Intell. User Interfaces*, 2021, pp. 597–607.

[174] A. Narechania, A. Srinivasan, and J. Stasko, "NL4DV: A toolkit for generating analytic specifications for data visualization from natural language queries," *IEEE Trans. Vis. Comput. Graphics*, vol. 27, no. 2, pp. 369–379, Feb. 2021.

[175] N. Nihalani, S. Silakari, and M. Motwani, "Natural language Interface for database: A brief review," *Int. J. Comput. Sci.*, vol. 8, no. 2, pp. 600–608, 2011.

[176] L. G. Nonato and M. Aupetit, "Multidimensional projection for visual analytics: Linking techniques with distortions, tasks, and layout enrichment," *IEEE Trans. Vis. Comput. Graphics*, vol. 25, no. 8, pp. 2650–2673, Aug. 2019.

[177] J. Obeid and E. Hoque, "Chart-to-Text: Generating natural language descriptions for charts by adapting the transformer model," in *Proc. Int. Natural Lang. Gener. Conf.*, 2020, pp. 138–147.

[178] M. Oppermann, R. Kincaid, and T. Munzner, "VizCommender: Computing text-based similarity in visualization repositories for content-based recommendations," *IEEE Trans. Vis. Comput. Graphics*, vol. 27, no. 2, pp. 495–505, Feb. 2021.

[179] M. Peters et al., "Deep contextualized word representations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2018, pp. 2227–2237.

[180] J. Poco and J. Heer, "Reverse-engineering visualizations: Recovering visual encodings from chart images," *Comput. Graph. Forum*, vol. 36, pp. 353–363, 2017.

[181] S. Pradhan et al., "CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2012, pp. 1–40.

[182] P. Qi et al., "Stanza: A Python natural language processing toolkit for many human languages," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 101–108.

[183] C. Qian, S. Sun, W. Cui, J. -G. Lou, H. Zhang, and D. Zhang, "Retrieve-then-adapt: Example-based automatic generation for proportion-related infographics," *IEEE Trans. Vis. Comput. Graphics*, vol. 27, no. 2, pp. 443–452, Feb. 2021.

[184] X. Qian et al., "Generating accurate caption units for figure captioning," in *Proc. Web Conf.*, 2021, pp. 2792–2804.

[185] X. Qian et al., "Learning to recommend visualizations from data," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2021, pp. 1359–1369.

[186] X. Qian et al., "Personalized visualization recommendation," 2021, *arXiv:2102.06343*.

[187] X. Qin, Y. Luo, N. Tang, and G. Li, "Making data visualization more efficient and effective: A survey," *VLDB J.*, vol. 29, no. 1, pp. 93–117, 2020.

[188] A. Quamar et al., "An ontology-based conversation system for knowledge bases," in *Proc. Int. Conf. Manage. Data*, 2020, pp. 361–376.

[189] S. Reinders, M. Butler, and K. Marriott, "Hey Model!" - Natural user interactions and agency in accessible interactive 3D models," in *Proc. Extended Abstr. Hum. Factors Comput. Syst.*, 2020, pp. 1–13.

[190] D. Ren, M. Brehmer, B. Lee, T. Höllerer, and E. K. Choe, "ChartAccent: Annotation for data-driven storytelling," in *Proc. IEEE Pacific Vis. Symp.*, 2017, pp. 230–239.

[191] A. Rind, W. Aigner, M. Wagner, and et al., "Task Cube: A three-dimensional conceptual space of user tasks in visualization design and evaluation," *Informat. Vis.*, vol. 15, no. 4, pp. 288–300, 2016.

[192] S. F. Roth, J. Kolojejchick, J. Mattis, and J. Goldstein, "Interactive graphic design using automatic presentation knowledge," in *Proc. Extended Abstr. Hum. Factors Comput. Syst.*, 1994, pp. 112–117.

[193] D. Saha et al., "ATHENA: An ontologydriven system for natural language querying over relational data stores," *Proc. VLDB Endowment*, vol. 9, no. 12, pp. 1209–1220, 2016.

[194] B. Saket, A. Endert, and C. Demiralp, "Task-based effectiveness of basic visualizations," *IEEE Trans. Vis. Comput. Graphics*, vol. 25, no. 7, pp. 2505–2512, Jul. 2019.

[195] A. Saktheeswaran, A. Srinivasan, and J. Stasko, "Touch? speech? or touch and speech? Investigating multimodal interaction for visual network exploration and analysis," *IEEE Trans. Vis. Comput. Graphics*, vol. 26, no. 6, pp. 2168–2179, Jun. 2020.

[196] A. Sarikaya and M. Gleicher, "Scatterplots: Tasks, data, and designs," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 1, pp. 402–412, Jan. 2018.

[197] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer, "Vega-Lite: A grammar of interactive graphics," *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 1, pp. 341–350, Jan. 2017.

[198] A. Satyanarayan, R. Russell, J. Hoffswell, and J. Heer, "Reactive vega: A streaming dataflow architecture for declarative interactive visualization," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 1, pp. 659–668, Jan. 2016.

[199] M. Savva et al., "ReVision: Automated classification, analysis and redesign of chart images," in *Proc. Annu. ACM Symp. User Interface Softw. Technol.*, 2011, pp. 393–402.

[200] E. Segel and J. Heer, "Narrative visualization: Telling stories with data," *IEEE Trans. Vis. Comput. Graphics*, vol. 16, no. 6, pp. 1139–1148, Nov./Dec. 2010.

[201] P. Seipel, A. Stock, S. Santhanam, A. Baranowski, N. Hochgeschwender, and A. Schreiber, "Speak to your software visualization-exploring component-based software architectures in augmented reality with a conversational interface," in *Proc. IEEE Work. Conf. Softw. Vis.*, 2019, pp. 78–82.

[202] J. Sen et al., "Natural language querying of complex business intelligence queries," in *Proc. Int. Conf. Manage. Data*, 2019, pp. 1997–2000.

[203] J. Seo and B. Shneiderman, "A rank-by-feature framework for interactive exploration of multidimensional data," *Informat. Vis.*, vol. 4, pp. 96–113, 2005.

[204] V. Setlur, S. Battersby, and T. Wong, "GeoSneakPique : Visual autocompletion for geospatial queries," in *Proc. IEEE Vis. Conf.*, 2021, pp. 166–170.

[205] V. Setlur et al., "Eviza: A natural language interface for visual analysis," in *Proc. Annu. ACM Symp. User Interface Softw. Technol.*, 2016, pp. 365–377.

[206] V. Setlur, E. Hoque, D. H. Kim, and A. X. Chang, "Sneak pique: Exploring autocompletion as a data discovery scaffold for supporting visual analysis," in *Proc. Annu. ACM Symp. User Interface Softw. Technol.*, 2020, pp. 966–978.

[207] V. Setlur and A. Kumar, "Sentifiers: Interpreting vague intent modifiers in visual analysis using word co-occurrence and sentiment analysis," in *Proc. IEEE Vis. Conf.*, 2020, pp. 216–220.

[208] V. Setlur, M. Tory, and A. Djalali, "Inferencing underspecified natural language utterances in visual analysis," in *Proc. Int. Conf. Intell. User Interfaces*, 2019, pp. 40–51.

[209] R. Sevastjanova et al., "Going beyond visualization: Verbalization as complementary medium to explain machine learning models," in *Proc. Visualization AI Explainability*, 2018, pp. 1–6.

[210] S. Shekarpour, E. Marx, A.-C. NgongaNgomo, and S. Auer, "SINA: Semantic interpretation of user queries for question answering on interlinked data," *J. Web Semant.*, vol. 30, pp. 39–51, 2015.

[211] L. Shen *et al.*, "TaskVis: Task-oriented visualization recommendation," in *Proc. Eurographics Conf. Vis.*, 2021, pp. 91–95.

[212] D. Shi *et al.*, "Task-oriented optimal sequencing of visualization charts," in *Proc. IEEE Visualization Data Sci.*, 2019, pp. 58–66.

[213] D. Shi, X. Xu, F. Sun, Y. Shi, and N. Cao, "Calliope: Automatic visual data story generation from a spreadsheet," *IEEE Trans. Vis. Comput. Graphics*, vol. 27, no. 2, pp. 453–463, Feb. 2021.

[214] Y. Shi, C. Bryan, S. Bhamidipati, Y. Zhao, Y. Zhang, and K. Ma, "MeetingVis: Visual narratives to assist in recalling meeting context and content," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 6, pp. 1918–1929, Jun. 2018.

[215] X. Shu, A. Wu, J. Tang, B. Bach, Y. Wu, and H. Qu, "What makes a data-GIF understandable?," *IEEE Trans. Vis. Comput. Graphics*, vol. 27, no. 2, pp. 1492–1502, Feb. 2021.

[216] N. Siddiqui and E. Hoque, "ConVisQA: A natural language interface for visually exploring online conversations," in *Proc. Int. Conf. Informat. Vis.*, 2020, pp. 440–447.

[217] T. Siddiqui *et al.*, "ShapeSearch: Flexible patternbased querying of trend line visualizations," *Proc. VLDB Endowment*, vol. 11, no. 12, pp. 1962–1965, 2018.

[218] T. Siddiqui *et al.*, "From sketching to natural language: Expressive visual querying for accelerating insight," *SIGMOD Rec.*, vol. 50, no. 1, pp. 51–58, 2021.

[219] S. Sigtia, E. Marchi, S. Kajarekar, D. Naik, and J. Bridle, "Multitask learning for speaker verification and voice trigger detection," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2020, pp. 6844–6848.

[220] A. Simitsis, G. Koutrika, and Y. Ioannidis, "Précis: From unstructured keywords as queries to structured databases as answers," *VLDB J.*, vol. 17, no. 1, pp. 117–149, 2007.

[221] S. Sinclair, R. Miikkulainen, and S. Sinclair, *Subsymbolic Natural Language Processing: An Integrated Model of Scripts, Lexicon, and Memory*, vol. 73. Cambridge, MA, USA: MIT Press, 1997.

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

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

[224] A. Srinivasan, M. Dontcheva, E. Adar, and S. Walker, "Discovering natural language commands in multimodal interfaces," in *Proc. Int. Conf. Intell. User Interfaces*, 2019, pp. 661–672.

[225] A. Srinivasan, S. M. Drucker, A. Endert, and J. Stasko, "Augmenting visualizations with interactive data facts to facilitate interpretation and communication," *IEEE Trans. Vis. Comput. Graphics*, vol. 25, no. 1, pp. 672–681, Jan. 2019.

[226] A. Srinivasan *et al.*, "InChorus: Designing consistent multimodal interactions for data visualization on tablet devices," in *Proc. Extended Abstr. Hum. Factors Comput. Syst.*, 2020, pp. 1–13.

[227] A. Srinivasan, B. Lee, and J. T. Stasko, "Interweaving multimodal interaction with flexible unit visualizations for data exploration," *IEEE Trans. Vis. Comput. Graphics*, vol. 14, no. 8, pp. 1–15, Aug. 2021.

[228] A. Srinivasan *et al.*, "Collecting and characterizing natural language utterances for specifying data visualizations," in *Proc. Extended Abstr. Hum. Factors Comput. Syst.*, 2021, pp. 1–10.

[229] A. Srinivasan and V. Setlur, "Snowy:Recommending utterances for conversational visual analysis," in *Proc. Annu. ACM Symp. User Interface Softw. Technol.*, 2021, pp. 864–880.

[230] A. Srinivasan and J. Stasko, "Natural language interfaces for data analysis with visualization: Considering what has and could be asked," in *Proc. Eurographics/IEEE VGTC Conf. Vis.*, 2017, pp. 55–59.

[231] A. Srinivasan and J. Stasko, "Orko: Facilitating multimodal interaction for visual exploration and analysis of networks," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 1, pp. 511–521, Jan. 2018.

[232] A. Srinivasan and J. Stasko, "How to ask what to say?: Strategies for evaluating natural language interfaces for data visualization," *IEEE Comput. Graph. Appl.*, vol. 40, no. 4, pp. 96–103, Jul./Aug. 2020.

[233] B. Steichen, G. Carenini, and C. Conati, "User-adaptive information visualization - Using eye gaze data to infer visualization tasks and user cognitive abilities," in *Proc. Int. Conf. Intell. User Interfaces*, 2013, pp. 317–328.

[234] C. Stolte, D. Tang, and P. Hanrahan, "Polaris: A system for query, analysis, and visualization of multidimensional relational databases," *IEEE Trans. Vis. Comput. Graphics*, vol. 8, no. 1, pp. 52–65, Jan./Mar. 2002.

[235] D. Streeb *et al.*, "Task-based visual interactive modeling: Decision trees and rule-based classifiers," *IEEE Trans. Vis. Comput. Graphics*, to be published, doi: 10.1109/TVCG.2020.3045560.

[236] N. Stylianou and I. Vlahavas, "A neural entity coreference resolution review," *Expert Syst. Appl.*, vol. 168, no. 114466, pp. 1–20, 2021.

[237] W. Su *et al.*, "VL-BERT: Pre-training of generic visual-linguistic representations," 2019, *arXiv:1908.08530*.

[238] Y. Suhara *et al.*, "Annotating columns with pre-trained language models," 2021, *arXiv:2104.01785*.

[239] Y. Sun, J. Leigh, A. Johnson, and S. Lee, "Articulate: A semi-automated model for translating natural language queries into meaningful visualizations," in *Proc. Int. Symp. Smart Graph.*, Berlin, Germany: Springer, 2010, pp. 184–195.

[240] J. R. Thompson, Z. Liu, and J. Stasko, "Data animator: Authoring expressive animated data graphics," in *Proc. Extended Abstr. Hum. Factors Comput. Syst.*, 2021, pp. 1–18.

[241] C. Tong *et al.*, "Storytelling and visualization: An extended survey," *Information*, vol. 9, no. 3, pp. 1–42, 2018.

[242] M. Tory and V. Setlur, "Do what I mean, not what I say! Design considerations for supporting intent and context in analytical conversation," in *Proc. IEEE Symp. Vis. Anal. Sci. Technol.*, 2019, pp. 93–103.

[243] B. Tversky, J. B. Morrison, and M. Betrancourt, "Animation: Can it facilitate?," *Int. J. Hum. Comput. Stud.*, vol. 57, no. 4, pp. 247–262, 2002.

[244] A. Van Dam, "Post-WIMP user interfaces," *Commun. ACM*, vol. 40, no. 2, pp. 63–67, 1997.

[245] S. Van DenElzen and J. J. Van Wijk, "Small multiples, large singles: A new approach for visual data exploration," *Comput. Graph. Forum*, vol. 32, no. 3 PART2, pp. 191–200, 2013.

[246] M. Vartak, S. Madden, A. Parameswaran, and N. Polyzotis, "SEEDB: Automatically generating query visualizations," *Proc. VLDB Endowment*, vol. 7, no. 13, pp. 1581–1584, 2014.

[247] M. Vartak *et al.*, "SEEDB: Efficient data-driven visualization recommendations to support visual analytics," *Proc. VLDB Endowment*, vol. 8, no. 13, pp. 2182–2193, 2015.

[248] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Int. Conf. Neural Informat. Process. Syst.*, 2017, pp. 6000–6010.

[249] C. Wang *et al.*, "CogNet: Bridging linguistic knowledge, world knowledge and commonsense knowledge," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 16114–16116.

[250] C. Wang *et al.*, "Falx: Synthesis-powered visualization authoring," in *Proc. Extended Abstr. Hum. Factors Comput. Syst.*, 2021, pp. 1–15.

[251] Q. Wang, Z. Chen, Y. Wang, and H. Qu, "A survey on ML4VIS: Applying machine learning advances to data visualization," *IEEE Trans. Vis. Comput. Graphics*, to be published, doi: 10.1109/TVCG.2021.3106142.

[252] W. Wang, Y. Tian, H. Wang, and W.-S. Ku, "A natural language interface for database: Achieving transfer-learnability using adversarial method for question understanding," in *Proc. Int. Conf. Data Eng.*, 2020, pp. 97–108.

[253] Y. Wang, F. Han, L. Zhu, O. Deussen, and B. Chen, "Line graph or scatter plot? Automatic selection of methods for visualizing trends in time series," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 2, pp. 1141–1154, Feb. 2018.

[254] Y. Wang *et al.*, "DataShot: Automatic generation of fact sheets from tabular data," *IEEE Trans. Vis. Comput. Graphics*, vol. 26, no. 1, pp. 895–905, Jan. 2020.

[255] Y. Wang, W. M. White, and E. Andersen, "PathViewer: Visualizing pathways through student data," in *Proc. Extended Abstr. Hum. Factors Comput. Syst.*, 2017, pp. 960–964.

[256] Y. Wang *et al.*, "InfoNice: Easy creation of information graphics," in *Proc. Extended Abstr. Hum. Factors Comput. Syst.*, 2018, pp. 1–12.

[257] Z. Wang, L. Sundin, D. Murray-Rust, and B. Bach, "Cheat sheets for data visualization techniques," in *Proc. Extended Abstr. Hum. Factors Comput. Syst.*, 2020, pp. 1–13.

[258] G. Wohlgenannt *et al.*, "A comparative evaluation of visual and natural language question answering over linked data," in *Proc. Int. Joint Conf. Knowl. Discov. Knowl. Eng. Knowl. Manage.*, 2019, pp. 473–478.

[259] K. Wongsuphasawat *et al.*, "Towards a general-purpose query language for visualization recommendation," in *Proc. Workshop Hum. Loop Data Anal.*, 2016, pp. 1–6.

[260] K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer, "Voyager: Exploratory analysis via faceted browsing of visualization recommendations," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 1, pp. 649–658, Jan. 2016.

[261] K. Wongsuphasawat *et al.*, "Voyager 2: Augmenting visual analysis with partial view specifications," in *Proc. Extended Abstr. Hum. Factors Comput. Syst.*, 2017, pp. 2648–2659.

[262] A. Wu *et al.*, "AI4VIS: Survey on artificial intelligence approaches for data visualization," *IEEE Trans. Vis. Comput. Graphics*, to be published, doi: 10.1109/TVCG.2021.3099002.

[263] A. Wu *et al.*, "MultiVision: Designing analytical dashboards with deep learning based recommendation," *IEEE Trans. Vis. Comput. Graphics*, vol. 28, no. 1, pp. 162–172, Jan. 2022.

[264] A. Wu *et al.*, "Learning to automate chart layout configurations using crowdsourced paired comparison," in *Proc. Extended Abstr. Hum. Factors Comput. Syst.*, 2021, pp. 1–13.

[265] H. Xia, "Crosspower: Bridging graphics and linguistics," in *Proc. Annu. ACM Symp. User Interface Softw. Technol.*, 2020, pp. 722–734.

[266] H. Xia, J. Jacobs, and M. Agrawala, "Crosscast: Adding visuals to audio travel podcasts," in *Proc. Annu. ACM Symp. User Interface Softw. Technol.*, 2020, pp. 735–746.

[267] Y. Xian *et al.*, "EXACTA: Explainable column annotation," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2021, pp. 3775–3785.

[268] X. Xu, C. Liu, and D. Song, "SQLNet: Generating structured queries from natural language without reinforcement learning," 2017, *arXiv:1711.04436*.

[269] S. Yagcioglu, A. Erdem, E. Erdem, and N. Ikizler-Cinbis, "RecipeQA: A challenge dataset for multimodal comprehension of cooking recipes," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2018, pp. 1358–1368.

[270] N. Yaghmazadeh, Y. Wang, I. Dillig, and T. Dillig, "SQLizer: Query synthesis from natural language," *Proc. ACM Program. Lang.*, vol. 1, pp. 1–26, 2017.

[271] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *IEEE Comput. Intell. Mag.*, vol. 13, no. 3, pp. 55–75, Aug. 2018.

[272] B. Yu and C. T. Silva, "VisFlow - web-based visualization framework for tabular data with a subset flow model," *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 1, pp. 251–260, Jan. 2017.

[273] B. Yu and C. T. Silva, "FlowSense: A natural language interface for visual data exploration within a dataflow system," *IEEE Trans. Vis. Comput. Graphics*, vol. 26, no. 1, pp. 1–11, Jan. 2020.

[274] T. Yu *et al.*, "Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2018, pp. 3911–3921.

[275] L. Yuan, Z. Zhou, J. Zhao, Y. Guo, F. Du, and H. Qu, "InfoColorizer: Interactive recommendation of color palettes for infographics," *IEEE Trans. Vis. Comput. Graphics*, to be published, doi: 10.1109/TVCG.2021.3085327.

[276] R. Zehrung, A. Singhal, M. Correll, and L. Battle, "Vis Ex Machina: An analysis of trust in human versus algorithmically generated visualization recommendations," in *Proc. Extended Abstr. Hum. Factors Comput. Syst.*, 2021, pp. 1–12.

[277] Z. Zeng *et al.*, "An evaluation-focused framework for visualization recommendation algorithms," *IEEE Trans. Vis. Comput. Graphics*, vol. 28, no. 1, pp. 346–356, Jan. 2022.

[278] D. Zhang *et al.*, "SATO: Contextual semantic type detection in tables," *Proc. VLDB Endowment*, vol. 13, no. 11, pp. 1835–1848, 2020.

[279] H. Zhang, Y. Song, Y. Song, and D. Yu, "Knowledge-aware pronoun coreference resolution," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 867–876.

[280] H. Zhang, X. Zhao, and Y. Song, "A brief survey and comparative study of recent development of pronoun coreference resolution," 2020, *arXiv:2009.12721*.

[281] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system," *ACM Comput. Surv.*, vol. 52, no. 1, pp. 1–38, 2019.

[282] Y. Zhang, P. Pasupat, and P. Liang, "Macro grammars and holistic triggering for efficient semantic parsing," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2017, pp. 1214–1223.

[283] Z. Zhang *et al.*, "CPM-2: Large-scale cost-effective pre-trained language models," 2021, *arXiv:2106.10715*.

[284] Z Wen, M. Zhou, and V. Aggarwal, "An optimization-based approach to dynamic visual context management," in *Proc. IEEE Symp. Informat. Vis.*, 2005, pp. 187–194.

[285] W. Zheng *et al.*, "Natural language question/answering," in *Proc. ACM Int. Conf. Informat. Knowl. Manage.*, 2017, pp. 217–226.

[286] X. Zhong, X. Qiao, Y. Cao, and R. W. H. Lau, "Content-aware generative modeling of graphic design layouts," *ACM Trans. Graph.*, vol. 38, pp. 1–15, 2019.

[287] V. Zhong *et al.*, "Seq2SQL: Generating structured queries from natural language using reinforcement learning," 2017, *arXiv:1709.00103*.

[288] M. Zhou *et al.*, "Table2Charts: Recommending charts by learning shared table representations," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2021, pp. 2389–2399.

[289] S. Zhu *et al.*, "A survey on automatic infographics and visualization recommendations," *Vis. Inform.*, vol. 4, no. 3, pp. 24–40, 2020.

[290] F. Özcan *et al.*, "State of the art and open challenges in natural language interfaces to data," in *Proc. Int. Conf. Manage. Data*, 2020, pp. 2629–2636.

Leixian Shen received the bachelor's degree in software engineering from the Nanjing University of Posts and Telecommunications in 2020. He is currently working toward the master's degree with the School of Software, Tsinghua University, Beijing, China. His research interests include data visualization and human computer interaction.

Enya Shen received the BSc degree from the Nanjing University of Aeronautics and Astronautics, in 2008, and the MSc and PhD degrees from the National University of Defense Technology, in 2010 and 2014, respectively. He works as an assistant researcher with School of Software, Tsinghua University. His research interests include data visualization, human computer interaction, augmented reality, and geometry modeling.

Yuyu Luo received the bachelor's degree in software engineering from the University of Electronic Science and Technology of China in 2018. He is currently working toward the PhD degree with the Department of Computer Science, Tsinghua University, Beijing, China. His research interests include data visualization and data cleaning.
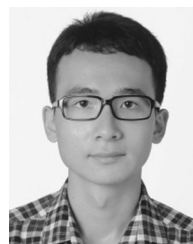
Xiaocong Yang is a fourth-year undergraduate working toward the BS degree with Tsinghua University, and plan to apply for a PhD position in computer science. His research interests include natural language processing and machine learning theory.
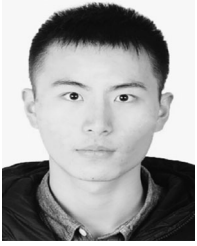
Xuming Hu received the BE degree in computer science and technology from the Dalian University of Technology. He is working toward the PhD degree with Tsinghua University. His research interests include natural language processing and information extraction.

Zhiwei Tai received the BS degree from the School of Software, Tsinghua University in 2020. He is currently working toward the master's degree in software engineering with Tsinghua University. His research interests include data visualization, digital twin, and augmented reality.

Xiongshuai Zhang received the bachelor's degree from the South China University of Techinology, China, in 2019. He is currently working toward the master's degree with the School of Software, Tsinghua University, China. His research interests include data visualization and computer graphics.

Jianmin Wang received the bachelor's degree from Peking University, China, in 1990, and the ME and PhD degrees in computer software from Tsinghua University, China, in 1992 and 1995, respectively. He is a full professor with the School of Software, Tsinghua University. His research interests include big data management and large-scale data analytics. He is leading to develop a big data management system with the National Engineering Lab for Big Data Software.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/csdl.