Calliope: Automatic Visual Data Story Generation from a Spreadsheet

Danqing Shi, Xinyue Xu, Fuling Sun, Yang Shi and Nan Cao

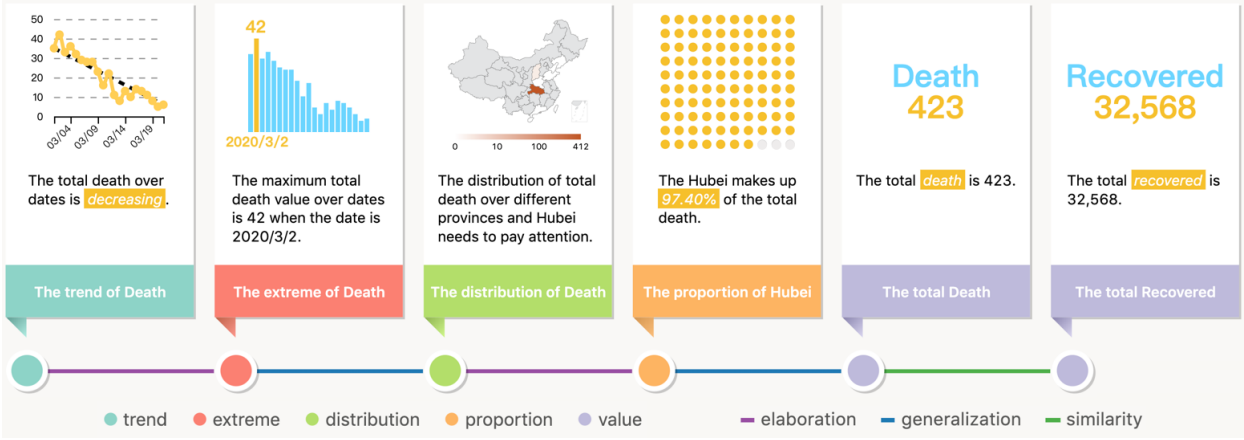

Fig. 1. A full-automatically generated visual data story about COVID-19 virus propagation in March 2020 in China. The whole story consists of six data facts shown in the ***storyline*** mode. The story first illustrates an overall trend of the total death followed by the elaborations of data details in Hubei, which is the most influenced province in China. Finally, the story is summarized by the total death and recovered in March, showing the improved situation in China.

Abstract—Visual data stories shown in the form of narrative visualizations such as a poster or a data video, are frequently used in data-oriented storytelling to facilitate the understanding and memorization of the story content. Although useful, technique barriers, such as data analysis, visualization, and scripting, make the generation of a visual data story difficult. Existing authoring tools rely on users' skills and experiences, which are usually inefficient and still difficult. In this paper, we introduce a novel visual data story generating system, Calliope, which creates visual data stories from an input spreadsheet through an automatic process and facilities the easy revision of the generated story based on an online story editor. Particularly, Calliope incorporates a new logic-oriented Monte Carlo tree search algorithm that explores the data space given by the input spreadsheet to progressively generate story pieces (i.e., data facts) and organize them in a logical order. The importance of data facts is measured based on information theory, and each data fact is visualized in a chart and captioned by an automatically generated description. We evaluate the proposed technique through three example stories, two controlled experiments, and a series of interviews with 10 domain experts. Our evaluation shows that Calliope is beneficial to efficient visual data story generation.

Index Terms—Information Visualization, Visual Storytelling, Data Story

1 Introduction

Visual data story is a representation of a series of meaningfully connected story pieces (i.e., data facts) in the form of a narrative visualization [23, 40]. Such form of representation is frequently used in a data-driven storytelling process due to its efficiency in terms of supporting the comprehension and memorization of the telling content [3, 20]. Examples include various data news created by journalists in New York Times and Prof. Hans Rosling's talks on human development trends. Although useful, creating a visual data story requires an author to have a variety of skills, including data analysis, visualization, and scripting, which are usually difficult to perform for an ordinary user.

Technical barriers motivate the rapid development of a variety of authoring tools for creating various types of narrative visualizations, such as data videos [1, 38], infographics [52], and annotated charts [35], to represent a data story. These tools usually incorporate built-in data analysis and visualization components to help with the story creation process. However, users must be involved to manually extract the data insights, write scripts, and assemble story pieces logically. Such involvement is usually inefficient and the quality of the resulting story still relies on the users' experiences and skills. Therefore, instead of developing another interactive authoring tool, a more intelligent system that can automatically generate a high-quality data story directly from the input data and support flexible story editing functions is desired by ordinary users who have limited data knowledge.

However, designing such a technique is difficult given numerous challenges to be addressed. First, automatically finding a set of informative elementary story pieces for building up a data story is usually difficult. It not only requires a fast exploration in an enormous data space to create candidate facts, but also needs qualitative measurement to estimate the importance of each fact to make them comparable, enabling the selection of important story pieces. Second, a data story is usually presented by narrative visualizations accompanied with text narrations and annotations to precisely express the message and avoid ambiguity [23, 40]. However, automatically generating human-

• *Danqing Shi, Xinyue Xu, Fuling Sun, Yang Shi, and Nan Cao are with Intelligent Big Data Visualization Lab at Tongji University.*

E-mail: {sdq, xuxinyue, fulingsun, yangshi.idvx}@tongji.edu.cn, nan.cao@gmail.com. Nan Cao is the corresponding author.

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

understandable visual and textual representations for each data fact in a story and matching them to each other is a challenging task due to the complexity of the data and the automatic generation problem. Third, all the facts must be organized in a narratively logical order to make the story understandable and meaningful when generating a story. However, computing to generate a narrative logic based on structured data remains a difficult problem supported by limited previous research in the visualization field.

To address the above challenges, we introduce Calliope[1], an intelligent system designed for automatically generating visual data stories from a spreadsheet. In the system, we introduce a logic-oriented Monte Carlo tree search (MCTS) algorithm to explore the space given by the input data and generate potential data facts for the story in a logical context while the exploration. The importance of the generated facts in each search step will be estimated by their information quantity, which is calculated based on the information theory [41] and their pattern significance calculated based on auto-insight techniques [12, 46, 51]. To facilitate a fast revision on the generated story, Calliope provides an interactive story editor, through which a user can easily edit on the general story logic and the details of each data fact. The final story can be further published on cloud for communication and sharing. The major contributions of the paper are as follows:

- **System.** We introduce the first system, to the best of our knowledge, that is designed to automatically generate visual data stories. The system also provides authoring and communication functions that enable easy story editing and sharing.
- **Story Generation Algorithm.** We introduce a logic-oriented Monte Carlo tree search algorithm that explores the data space to generate a series of data facts in logical context to build a story. The algorithm avoids the time-consuming enumeration of the data space via a reward function and a logic filter to ensure the quality of the generation results. The run-time of each searching step can also be precisely controlled to ensure efficiency.
- **Story Information Measurement.** We introduce the first method, to the best of our knowledge, which can precisely estimate the information quantity of a data story. Specifically, we calculate the self-information of each story fact and estimate the content of the story based on the information entropy by using and extending the definitions introduced in the information theory.
- **Evaluation.** We demonstrate the utility of the proposed system via three example stories generated based on real-world data, two controlled experiments designed to verify the generated logic, and a series of interviews with 10 expert users respectively from three areas, including data journalism, business intelligence, and the visualization research community.

2 Related Work

In this section, we review the recent studies that are most relevant to our work, including data-driven storytelling, automatic data visualization, and natural language generation.

2.1 Data-Driven Storytelling

Data-driven storytelling is a rapidly developing research direction that focuses on techniques for enhancing data understanding, information expression, and communication. Narrative visualization is one promising approach frequently used for data-driven story telling [47]. Recently, the visualization community has extensively investigated storytelling and narrative visualization techniques [40, 47]. According to Segel and Heer [40], narrative visualization can be largely classified into seven genres, including magazine style, annotated chart, partitioned poster, flow chart, comic strip, slideshow, and video. Evidence shows that an effective composition and visual narrative of the story can guide readers through the data and improve the comprehension and memory [3, 20]. To compose effective data-driven stories, Hullman *et al.* [20] identified several key design actions in the sequential story creation process, including context definition, facts selection, modality selection, and order selection. Lee *et al.* also decomposed the creation process of a data-driven story into three major steps: insights finding, scripting, and communicating to the audience [23]. These valuable study results guide the designs of many data story creation systems including Calliope, which is introduced in this paper.

Users commonly experience difficulty in creating a data-driven story due to technical barriers, which motivate the design and development of various authoring tools. General tools such as Ellipsis [37] allows a user to directly integrate visualizations to an illustrative story. Recent studies focus on designing tools to generate a specific type of visual narrative. For example, ChartAccent [35] and InfoNice [52] are respectively designed for creating infographics and annotated charts. Narvis [50] is introduced to extract the combination of visual elements of a visualization and organize them as a slideshow to help with the narrative interpretation of a visualization design. DataClips [1] is designed to help users generate data videos. Various authoring tools that are specifically designed to create narrative visualization for certain types of data. For example, Timeline Storyteller [4], is a visual storytelling tool designed specifically for time-oriented data. Several visualization tools are also introduced to bridge the gap between visual analysis and storytelling [6, 18], but they still target on expert users.

The above tools assume that the story content is manually created, resulting in inefficiency. By contrast, Calliope supports automatic story generation and flexible story editing functions, which ensure the quality, lower the barrier, and improve the efficiency of visual narration.

2.2 Automatic Data Visualization

Studies on automatic visualization have experienced three stages, including visualization chart recommendation, automatic data mapping generation, and auto-insights. To recommend a chart given an input data, early studies employed a rule-based method, which checks the data types to make a suggestion [17, 29]. A recent study [19] trained a classification model based on a collection of "data feature - chart type" pairs extracted from a visualization design corpus. As a result, given data features, a proper chart type can be selected. In Calliope, we select a chart for each data fact based on its fact type and data fields.

To visualize data in a chart, one must determine the detailed data mapping strategy. To this end, various techniques are introduced. Draco [33] uses an optimization model to find the best data mapping strategy under a set of constraints formulated by several common visual design guidelines. DeepEye [28] enumerates all possible data mapping strategies and uses a decision tree to select the good ones. Data2Vis [11] "translates" the data into a visual encoding scheme based on a sequence-to-sequence deep model. Text-to-Viz [8] employs natural language processing techniques to identify and parse data entities, such as numbers, portions, and data scopes from an input text, and convert them into a statistic diagram. Shi *et al.* [42] explored the chart design space via a reinforcement learning model and generated a sequence of data mapping approaches regarding an analytical task. Calliope encodes different data fact fields in a chart following a rule-based method.

Recent studies focused on extracting data patterns and representing them in charts to reveal data insights, i.e., auto-insights [12, 46]. The extracted insights can be quantitatively measured based on their statistical significance [12]. Visual analysis techniques were also developed to support auto-insights. For example, SeeDB [49] finds and illustrates the most interesting trend in the data by exploring various data mapping strategies. Foresight [10] extracts and visualizes insights from multidimensional data using rule-based methods. DataShot [51] randomly visualizes a set of automatically generated data facts as a fact-sheet. Inspired by DataShot, Calliope also borrowed the auto-insights techniques to generate data facts, but made a step further by organizing the facts in a logical order to generate a meaningful data story.

2.3 Natural Language Generation

Recently, studies on natural language generation (NLG) demonstrate the capability of producing descriptive text from various types of data [16, 32, 48]. Many methods use templates to generate sentences [24, 45], and techniques that automatically enrich a template

[1] Calliope is available online at `https://datacalliope.com`

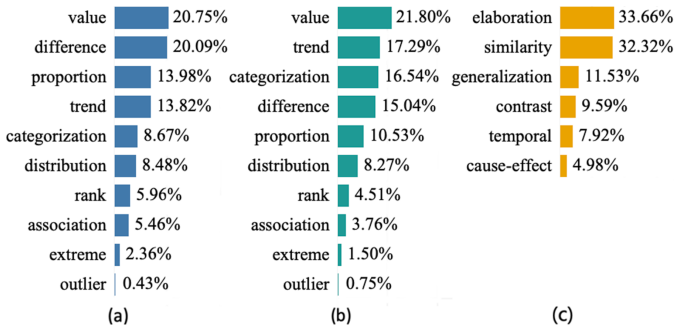

Fig. 2. The statistics on the labeled data video corpus showing the percentage of (a) data facts, (b) fact types used as the start points, and (c) narrative relationships.

Fact Types	Bar	Line	ISOType	Map	Scatter plot	Pie	Area	Bubble	Text	Table	Box plot	Treemap
Value	56		26	40					204	9		
Difference	99	37	23	39					46			
Proportion	45		68	9		131			58			
Trend	60	170	12	17			14					
Categorization	24		4	12			4					5
Distribution	40		28	107			12					
Rank	24			15		3			22	5		
Association		17		11	9			2	1			
Extreme	12	2	4	10					4			
Outlier			1	1			4		2		2	

Table 1. The frequency of visualization charts regarding the 10 fact types.

were developed [13, 57]. Recent studies leveraged deep learning models to generate textual content from scratch [14, 15]. Several methods create an intermediate structure based on a recurrent neural network [31, 55, 56], and others use the auto-encoder architecture to generate diverse sentences from a latent space [25, 26]. Among various techniques, those aim to generate text content based on structured data are the most relevant to our work [2, 21, 30]. For example, commercial software, such as PowerBI[2] and Quill[3] describe important data facts based on a set of templates to help interpret the data and the corresponding visualization. A number of visual auto-insights systems, such as DataSite [9], DataShot [51], and Voder [44], use the template-based NLG to generate captions for visualization charts. In Calliope, we also employ the template-based method to generate captions for each chart, but to ensure the readability and avoid ambiguity, we define a syntax for each fact type that regulates the generation results.

3 Design of the Calliope System

This section introduces the design of Calliope system. We first introduce the formal definition of a data story, then survey on a collection of data videos to help us understand how a story is generated by human designers. After that, we summarize the design requirements and introduce the architecture design of the system.

3.1 Data Story

Data story is a set of story pieces that are meaningfully connected to support the author's communication goal [23]. A story piece is a fact backed up by data, and it is usually visualized by succinct but expressive charts, accompanied with annotations (labels, pointers, text, etc.) and

[2] https://powerbi.microsoft.com
[3] https://narrativescience.com

Fact Types	r_s	r_t	r_c	r_a	r_e	r_g
Value	45.6	8.9	0.0	4.2	26.8	14.5
Difference	41.6	6.7	0.0	5.8	31.1	14.8
Proportion	52.1	7.3	0.0	5.2	22.4	13.0
Trend	34.7	9.4	8.2	7.1	28.2	12.4
Categorization	37.7	3.4	0.0	3.4	47.5	7.8
Distribution	49.0	12.1	0.0	4.4	22.3	12.1
Rank	43.8	11.7	0.0	6.6	34.3	3.6
Association	31.0	5.6	15.1	7.1	26.2	15.1
Extreme	51.8	5.6	0.0	3.7	25.9	13.0
Outlier	20.0	10.0	0.0	10.0	40.0	20.0

Table 2. The likelihood of coherence relations, including similarity (r_s), temporal (r_t), contrast (r_c), cause-effect (r_a), elaboration (r_e), and generalization (r_g) used after the 10 fact types.

narrations to express the message and avoid the ambiguity. We design Calliope to automatically generate visual data stories by following this definition. Formally, a data story $\mathcal{S}$ consists of a sequence of data facts that are connected by coherent relations (denoted as $r_i \in \mathcal{R}$) $\{f_1, r_1, \cdots, f_{n-1}, r_{n-1}, f_n\}$ with each fact $f_i \in F$. We will use these notations throughout the paper.

3.2 Preliminary Survey

Before designing the system, it is necessary to understand how a data story is created by a human designer. To this end, as data video is a frequently used narrative visualization form, we collected a set of 602 data videos from YouTube and Vimeo by searching keywords, such as "animated infographic", "data video", and "motion infographic". A total of 230 high-quality videos were selected and manually segmented into 4186 story pieces. The fact and chart types of 2583 data-related story pieces and the coherence relations used for connecting two succeeding pieces were labeled for analysis. Here, we borrowed the definition of fact types introduced in DataShot [51] and coherence relations introduced in [53, 54] to label our data.

As a result, even some simple statistics are able to help us answer a number of questions that are critical to the design of an automatic story generation system. For example, Fig. 2 suggests the frequently used fact types, the fact types frequently used as start points, and the frequently used coherence relations regarding our data video corpus. Table 1 suggests the preferred visualization charts given a fact type. Table 2 suggests the narrative logic probably used following each fact type. These results guide the design of the story generation algorithm and will be further described later.

3.3 System Design

Our goal is to design a system that can automatically generate high-quality initial data stories directly from an input spreadsheet and support flexible story editing functions to lower the technical barriers of creating a data story. To achieve this goal, a number of key requirements need to be fulfilled:

- **R1 Generating "successful" stories.** The most important thing for the system is to ensure the quality of story generation. Among various factors that contribute to a successful narrative artifact, the key is understandability [36], which is usually determined by the narrative logic, and the meaningful and believable content [23]. Therefore, the system should be intelligent enough to automatically generate meaningful stories logically with correct, i.e., believable, data backups.
- **R2 Efficient story generation.** The system should be able to efficiently generate a data story within a reasonable period of time that is affordable to the users. Therefore, the generation time should be controllable to grantee the efficiency of the system while keeping the quality of the story.
- **R3 Expressive story representation.** As suggested in [23], the generated visual data story should be expressively represented in

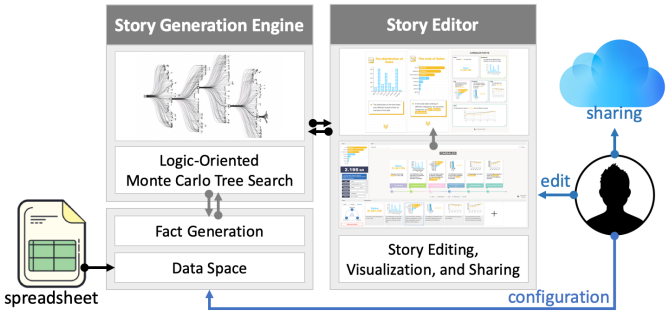

Fig. 3. Calliope system consists of two modules : the story generation engine and the story editor.

both visual and textual forms to precisely express the message and avoid ambiguity. Here, simple but intuitive charts [51], as well as precise and meaningful narratives [23] should be guaranteed to reduce users' learning efforts.

R4 **Easy story editing.** The system should provide flexible interactions to support comprehensive editing of the generated storyline, text narration, visual representation, and the corresponding data facts, so that a user can further refine and adjust an automatically generated story based on their own requirements.

R5 **Easy communication and sharing.** The visual and textual representations of a data story should be probably aligned and adaptive laid out to fit into different devices such as a laptop, tablets, and smartphones to facilitate an easy story exploration, communication, and sharing.

To fulfill these requirements, the design of Calliope system consists of two modules (Fig. 3): (1) the story generation engine and (2) the story editor. The story generation engine is designed based on a logic-oriented Monte Carlo tree search process, in which a story is gradually generated fact by fact while searching through the data space defined by an input spreadsheet. The whole search process is guided by narrative logic and a reward function that measures the importance of facts to ensure the quality of the generated story (**R1**). In addition, the time spent on each searching step is configurable, which guarantees the generation efficiency (**R2**). The generated story is visualized in the story editor as a series of captioned visualization charts (**R3**), whose data facts, caption, chart type, and logic orders can be revised according to user preferences (**R4**). The final visual data story can be represented in three modes to fit different devices (**R5**).

4 Story Generation Engine

In this section, we first formally define a data fact and its importance measurement. We then describe the details of the proposed automatic story generation algorithm.

4.1 Data Facts

Data facts are the elementary building blocks of a data story. Each of these facts represents a piece of information extracted from the data. We first give a formal definition of data facts by simplifying the concepts introduced in [7, 51] to guarantee a clear semantic and then introduce a novel method used to estimate the importance of a given data fact.

Definition A data fact is designed to measure a collection of data items in a subspace of an input dataset based on a measurable data field. The data items can be further divided into groups via a breakdown method. Formally, a fact, $f_i \in F$, is defined by a 5-tuple:

$$\begin{aligned} f_i &= \{type, subspace, breakdown, measure, focus\} \\ &= \{t_i, s_i, b_i, m_i, x_i\} \end{aligned}$$

where ***type*** (denoted as t_i) indicates the type of information described by the fact. As summarized in Table 3, Calliope includes 10 fact types; ***subspace*** (denoted as s_i) describes the data scope of the fact, which is defined by a set of data filters in the following form:

$$\{\{\mathcal{F}_1 = \mathcal{V}_1\}, \cdots, \{\mathcal{F}_k = \mathcal{V}_k\}\}$$

where $\mathcal{F}_i$ and $\mathcal{V}_i$ respectively indicate a data field and its corresponding value selected to filter the data. By default, the subspace is the entire dataset. ***breakdown*** (denote as b_i) is a set of temporal or categorical data fields based on which the data items in the subspace are further divided into groups; ***measure*** (denote as m_i) is a numerical data field based on which we can retrieve a data value or calculate a derived value, such as count, sum, average, minimum, or maximum, by aggregating the subspace or each data group; ***focus*** (denote as x_i) indicates a set of specific data items in the subspace that require attention. Despite the above five fields, certain facts may also have a ***derived value*** (denoted as V_d) such as a textual summary of the trend (i.e., "increasing" or "decreasing") or the specific difference value between two cases described by a difference fact, or the correlation coefficient computed for an association fact as shown in Table 3. These values help with a more insightful description of the fact.

When compared to the concepts introduced in [7, 51], the above definition simplified and restricted the fact fields to ensure a clear semantic expression of the data that avoids redundancy, overlaps, and ambiguity. Specifically, when compared to [51], we removed the fact fields that are irrelevant to the fact semantics and treated "aggregation" as an operation on "measures" instead of a fact type to avoid duplicated fact definitions. In addition, as summarized in Table 3, we add constraints on each fact field to ensure a clear semantics. For example, the facts in "distribution" and "trend" types are designed to capture the data patterns given by the measures of different data groups in the subspace. Both fact types can be differentiated by their ways of breaking down a subspace: the subspace in a "trend" fact must be divided by a temporal field, whereas the subspace in a "distribution" fact can only be divided by a categorical field. Thus, each fact can be described by a syntax which is used for generating a textual description of the fact.

To understand the above concepts, let's consider the following examples. Given a dataset about the COVID-19 virus outbreak in China, the data fact, $\{$*"distribution"*, $\{\{$*Country* = *"China"*$\}\}$, $\{$*Province*$\}$, $\{$*sum(Infections)*$\}$, $\{$*Province*=*"Hubei"*$\}\}$, describes "the distribution of the total number of infections over all provinces when the country is China (subspace) and Hubei needs to pay attention" regarding to the syntax of the distribution fact. Similarly, the data fact, $\{$*"trend"*, $\{\{$*Province* = *"Hubei"*$\}\}$, $\{$*Date*$\}$, $\{$*sum(Infections)*$\}$, $\{$*Date*=*"2020-1-24"*$\}\}$, indicates "the changing trend of the total number of infections over different dates when province is Hubei and the values of 2020-1-24 need to pay attention".

Importance Score We estimate the importance of a data fact $f_i = \{t_i, s_i, b_i, m_i, x_i\} \in F$ based on its *self-information* (denoted as $I(f_i)$) weighted by its *pattern significance* (i.e., $S(f_i) \in [0, 1]$) as follows:

$$I_s(f_i) = S(f_i) \cdot I(f_i) \tag{1}$$

In particular, $I(f_i) \in [0, \infty]$ is defined based on the information theory and can be measured in "bit" using the following formula:

$$I(f_i) = -log_2(P(f_i)) \tag{2}$$

where $P(f_i)$ indicates the occurrence probability of the fact given the input data. A data fact with a lower occurrence probability in the data space has a higher self-information value as it reveals uncommon patterns which are usually more meaningful and interesting. $P(f_i)$ is formally determined by the occurrence probability of the fact's ***subspace*** s_i, the probabilities of selecting x_i as the ***focus*** in s_i, and the probabilities of choosing m_i ($P(m_i|t_i)$) and b_i ($P(b_i|t_i)$) to measure and break down s_i given a fact type t_i:

$$P(f_i) = P(m_i|t_i) \cdot P(b_i|t_i) \cdot P(s_i) \cdot P(x_i|s_i) \tag{3}$$

where $P(m_i|t_i)$ and $P(b_i|t_i)$ is defined regarding the data type constraints of m_i and b_i as summarized in Table 3. For example, when the fact type is "Value", $P(m_i|Value)$ is $1/N$, where N is the total number of numerical fields in the data. Similarly, $P(b_i|Difference)$ is $1/(C+T)$ where C, T are the total number of categorical and temporal fields in the data. Moreover, in Formula (3), $P(x_i|s_i)$ is defined as the proportion of the focused data items in the subspace, i.e.,

Fact Type (t_i)	Subspace (s_i)	Breakdown (b_i)	Measure (m_i)	Focus (x_i)	Derived Value (V_d)	Syntax
Value	*	×	N	×	derived value	The {{agg}} {{m_i}} is {{V_d}} when {{s_i}}.
Difference	*	C/T	N	=2	difference value	The difference between {{x_i[1]}} and {{x_i[2]}} regarding to their {{agg}} {{m_i}} is {{V_d}} when {{s_i}}.
Proportion	*	C/T	N	=1	percentage	The {{x_i}} accounts for {{V_d}} of the {{agg}} {{m_i}} when {{s_i}}.
Trend	*	T	N	≥ 0	increasing/decreasing	The {{V_d}} trend of {{agg}} {{m_i}} over {{b_i}}(s) when {{s_i}} and the values of {{x_i}} needs to pay attention.
Categorization	*	C	×	≥ 0	number of categories	There are {{V_d}} {{b_i}}(s) which are {{C_1}}, {{C_2}}, {{...}}, and {{C_n}} when {{s_i}}, among which {{x_i}} needs to pay attention.
Distribution	*	C	N	≥ 0	×	The distribution of the {{agg}} {{m_i}} over {{b_i}}(s) when {{s_i}} and {{x_i}} needs to pay attention.
Rank	*	C/T	N	=3 (top 3)	×	In the {{agg}} {{m_i}} ranking of different {{b_i}}(s), the top three {{b_i}}(s) are {{x_i[0]}}, {{x_i[1]}}, {{x_i[2]}}, when {{s_i}}.
Association	*	C/T	$N \times N$	×	correlation coefficient	The Pearson correlation between the {{m_i[1]}} and the {{m_i[2]}} is {{V_d}} when {{s_i}}.
Extreme	*	C/T	N	=1 (max/min)	maximum/minimum	The {{V_d}} value of the {{agg}} {{m_i}} is {{x_i}} when {{s_i}}.
Outlier	*	C/T	N	=1 (outlier)	outlier score	The{{agg}} {{m_i}} of {{x_i}} is an outlier when compare with that of other {{b_i}}(s) when {{s_i}}.

Table 3. Definitions of the 10 types of data facts, where N, C, T respectively indicate the numerical, categorical, and temporal data types. {agg} is one of the following terms that indicates an aggregation method: total number of (i.e., count), total value of (i.e., sum), average, maximum, and minimum.

$P(x_i|s_i) = count(x_i)/count(s_i)$, with the assumption that the probability of selecting each data item as a focus in the subspace is equivalent. In our design, all the data items in s_i are focused by default, i.e., $P(x_i|s_i) = 1$ when the ***focus*** field is unspecified. To calculate $P(s_i)$, we first assume s_i consists of k data filters, i.e., $\{\{\mathcal{F}_1 = \mathcal{V}_1\}, \cdots, \{\mathcal{F}_k = \mathcal{V}_k\}\}$ and there are m independent data fields that can be used for formulating a subspace. In this way, $P(s_i)$ is defined as follows:

$$P(s_i) = \frac{1}{\sum_{i=0}^{m} C(m,i)} \cdot \prod_{j=1}^{k} P(\mathcal{F}_j = \mathcal{V}_j) \tag{4}$$

where the first term indicates the probability of choosing the fields $\mathcal{F}_1, \cdots, \mathcal{F}_k$ from the input data to formulate the subspace s_i. $C(m,i)$ is an i-combination over a set of m possible data fields. $\sum_{i=0}^{m} C(m,i)$ summarizes the number of all possible cases for formulating a subspace. In this way, the first term in Formula (4) indicates the method we use for formulating the current subspace s_i is just one possible case. The second term in Formula (4) indicates the probability of using the corresponding values $\mathcal{V}_1, \cdots, \mathcal{V}_k$ on the selected fields to filter the data. This probability is directly given by the products of the proportions of the data that satisfy each filter conditions, i.e., $\{\mathcal{F}_j = \mathcal{V}_j\}$.

In Formula (1), $S(f_i) \in [0,1]$ estimates the significance of the data patterns described by the fact f_i, which is calculated based on auto-insight techniques [12, 46, 51]. The detailed methods are described in the supplemental material. It worth mentioning that a significant pattern may not necessary have a high self-information value. Only using both measurements as shown in Formula (1) will guarantee a comprehensive estimation. Under this definition, the importance of a fact is only determined by its data content and irrelevant to how frequently a type of fact is used in data stories (Fig. 2(a)).

4.2 Story Generation Algorithm

In Calliope, we introduce an intelligent algorithm that generates data facts from an input spreadsheet and threads them logically to create a meaningful data story. However designing such an algorithm is challenging. The story design space, formulated by a collection of data facts generated from the input spreadsheet, could be extremely large due to the huge number of possible combinations of the fact fields even based on a small dataset. The algorithm cannot generate all facts first and then pick up those important ones to build the story as users cannot afford a long waiting time. We addressed this issue by introducing an efficient logic-oriented searching algorithm based on the Monte Carlo tree search (MCTS) [5, 43]. The algorithm can efficiently explore a large space that contains a huge number of states via a searching process organized by a tree structure and guided by a reward function towards a logic direction.

Algorithm Overview In general, the algorithm explores the design space by dynamically constructing a searching tree $\mathcal{T}$. As shown in Fig. 4(a), each node in the tree is a data fact f_i and each directed edge indicates a logic relation r_i. A data story $\mathcal{S} = \{f_1, r_1, \cdots, f_{n-1}, r_{n-1}, f_n\}$ is thus represented by a path starting from the root. A reward function is designed to estimate the quality of each path in the tree. The reward scores are marked on the last node in paths. A node shared by multiple paths is weighted by the maximum reward. These scores are used to guide the exploration of the design space.

The tree $\mathcal{T}$ is gradually generated through a searching process as described in Algorithm 1. In particular, the algorithm takes a spreadsheet $\mathcal{D}$, i.e., a data table, and a goal $\mathcal{G}$, such as generating a story with a desired information quantity or length as the inputs and automatically generates a story $\mathcal{S}$ that fulfills the goal. Initially, it randomly generates a set of facts in types that are frequently used as the starting point in a data story (Fig. 2(b)). These facts usually reveal general and common data patterns which may already known by the audience as the background of the story. Among these facts, the most important one, denoted as f_0, is used as the root of $\mathcal{T}$. In the next, the algorithm generates a story by iteratively searching more informative and significant data facts to elaborate the story via four major steps: ***selection***, ***expansion***, ***simulation***, and ***back-propagation***. The first step finds a

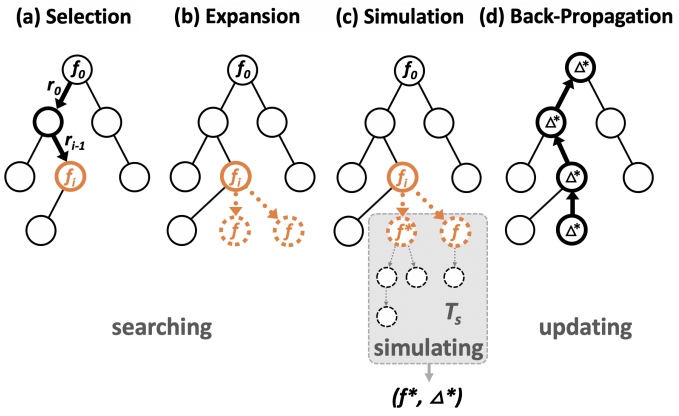

Fig. 4. An iteration of the logic-oriented Monte Carlo tree search consists of four steps, including (a) selection, (b) expansion, (c) simulation, and (d) back-propagation.

Algorithm 1: Logic-Oriented Monte Carlo Tree Search

Input : $\mathcal{D}, \mathcal{G}$
Output : $\mathcal{S} = \{f_1, r_1 \cdots, f_{n-1}, r_{n-1}, f_n\}$

1 $f_0 \leftarrow$ Initialize($\mathcal{D}$); $\mathcal{T} \leftarrow \{f_0\}$; $\mathcal{S} \leftarrow \{\}$;
2 **while** $\mathcal{G}$ *is not fulfilled* **do**
/* 1.***selection*** */
3 $f_i \leftarrow select(\mathcal{T})$;
/* 2.***expansion*** */
4 $F_i \leftarrow$ Expand(f_i);
/* 3.***simulation*** */
5 $\mathcal{T}_s \leftarrow \{f_i\}$; $F \leftarrow F_i$; $f_p \leftarrow f_i$;
6 **while** *within time limitation* **do**
// Calculate the reward of each node in F in context of $\mathcal{T}_s$ and find the node $f \in F$ with the highest reward Δ. The design space will be explored in direction of $f_p \rightarrow f$ in the simulation process.
7 $f, \Delta \leftarrow$ Reward($\mathcal{T}_s, F$);
// Add f in the simulation tree $\mathcal{T}_s$ as a child of f_p and update reward of the relevant nodes in $\mathcal{T}_s$ based on Δ. After that, find the node $f^* \in F_i$ with the highest reward in $\mathcal{T}_s$, where $f_i \rightarrow f^*$ determines the best searching direction found so far.
8 $f^*, \Delta^* \leftarrow$ BackPropagation($\mathcal{T}_s, f, \Delta$);
// Select the next node f_p and expand it in the simulation tree for a further exploration
9 $f_p \leftarrow select(\mathcal{T}_s)$; $F \leftarrow$ Expand(f_p);
10 **end**
/* 4.***back-propagation*** */
11 BackPropagation($\mathcal{T}, f^*, \Delta^*$);
12 $\mathcal{S} \leftarrow \mathcal{P}^* = \{f_1, r_1, \cdots, f_{n-1}, r_{n-1}, f_n\}$;
13 **end**
14 **return** $\mathcal{S}$;

node f_i with the largest reward in $\mathcal{T}$, from which the next searching step will be performed (*line 3*, Fig. 4(a)). The second step searches the design space by creating a set of data facts (denoted as F_i), that is logically relevant to f_i (*line 4*, Fig. 4(b)). The third step finds the best searching direction $f_i \rightarrow f^*, f^* \in F_i$ with the largest reward Δ^* through a simulation process (*lines 5 - 11*, Fig. 4(c)). This process simulates the cases in which each $f \in F_i$ is expanded in a simulation tree $\mathcal{T}_s$ rooted at f_i to help the algorithm explore the space a few steps further, so that the different searching directions can be estimated in advance. The simulation runs within a time limit to ensure the efficiency of the algorithm (**R2**). In the last step, the tree $\mathcal{T}$ is updated via a back-propagation process, in which the weights of the relevant nodes are updated based on Δ^* and f^* is formally added into $\mathcal{T}$ as a child of f_i (*line 11*, Fig. 4(d)). Finally the path with the highest reward in $\mathcal{T}$, $\mathcal{P}^*$, is identified as the best story generated at the current iteration (*line 12*). The algorithm stops when the goal $\mathcal{G}$ is fulfilled.

Logic-Oriented Node Expansion Expanding a selected node f_i in the search tree $\mathcal{T}$ to elaborate the story design space is a critical step in the aforementioned searching algorithm. The expansion should generate a set of nodes that is logically relevant to f_i to gradually generate a meaningful data story through the searching process. To this end, we investigated how a set of commonly used coherence relations (denoted as $\mathcal{R}$) [53, 54] was used in data stories during our preliminary survey. As a result, Table 2 summarizes the likelihood, $P(r_i|f_i)$, of each relation r_i occurring after a fact f_i regarding to their fact types, which guides the node expansion process. In particular, during the expansion, we create a set of data facts regarding each coherence relation. The proportion of the newly generated facts is given by $P(r_i|f_i)$, and each new fact, f_{i+1}, is generated by the following rules:

- ***Similarity*** indicates two succeeding facts are logically parallel to each other. Therefore, f_{i+1} can be generated by a variety of methods, such as modifying the measure / breakdown / focus field without changing the subspace.
- ***Temporal*** relation communicates the ordering in time of events or states. In this case, we generate f_{i+1} by changing the value of the temporal filter in f_i's subspace to a succeeding time.
- ***Contrast*** indicates a contradiction between two facts. For simplicity, we only check the contradictions in two types of facts, i.e., trend and association. f_{i+1} is generated by modifying the subspace of f_i to form a data contradiction in measures. For example, the sales trends of a product increases, but that of another product decreases. The sales number of a product is positively associated with its price, but the association is negative in case of another product. In these examples, the subspace is determined by different products.
- ***Cause-Effect*** indicates the later event is caused by the former one. In multidimensional data, a causal relation can be determined between dimensions based on the data distribution. In this way, f_{i+1} can be generated by changing the measure field m_i of f_i to another numerical field in the spreadsheet that is most likely caused by m_i in accordance with causal analysis [39].
- ***Elaboration*** indicates a relation in which a latter fact f_{i+1} adds more details to the previous one f_i. In this way, we create f_{i+1} by shrinking the scope of f_i's subspace via adding more constraints (i.e., filters) or setting a focus to "zoom" f_i into a more specific scope.
- ***Generalization*** indicates f_{i+1} is an abstraction of the previous f_i, which is in opposite to elaboration. Therefore, we create f_{i+1} by enlarging the scope of f_i's subspace via removing constraints.

Reward Function We propose a reward function that estimates the quality of each generated story $\mathcal{S}$ via three criteria, i.e., diversity $D(\mathcal{S})$, logicality $L(\mathcal{S})$, and integrity (i.e, data coverage) $C(\mathcal{S})$ based on the story's information entropy $H(\mathcal{S})$:

$$reward(\mathcal{S}) = \{\gamma_1 \cdot D(\mathcal{S}) + \gamma_2 \cdot L(\mathcal{S}) + \gamma_3 \cdot C(\mathcal{S})\} \cdot H(\mathcal{S}) \quad (5)$$

where $\gamma_i \in [0,1], \sum_i \gamma_i = 1$ are the weighting parameters given by users to balance different criteria. All the criteria are also normalized to $[0,1]$. $H(\mathcal{S})$ is the story's information entropy that indicates the expected self-information of the data facts in the story, which is used as the basis of the reward and formally defined as follows:

$$H(\mathcal{S}) = \sum_{i=1}^{n} P(f_i) \cdot I_s(f_i) = -\sum_{i=1}^{m} P(f_i) \cdot S(f_i) \cdot \log_2(P(f_i)) \quad (6)$$

where, $I_s(f_i)$ is the fact's importance score defined in Formula (1). The definition of each story estimation criterion is described as follows:

- ***Diversity*** estimates the variance of the fact types in $\mathcal{S}$. Rich fact types will make the story vivid and attractive. Diversity is given by two terms as follows:

$$D(\mathcal{S}) = \frac{n}{\min(|\mathcal{S}|, 10)} \cdot \frac{-\sum_{i=1}^{n} p_i \cdot \ln(p_i)}{\ln(n)} \quad (7)$$

  where n indicates the total number of fact types used in $\mathcal{S}$ whose maximum value is known as 10; p_i is the proportion of the i-th fact type in the story. When $D(\mathcal{S})$ is maximized, the first term encourages to use more fact types in a story, and the second term, a normalized Shannon diversity index [34], ensures that different fact types can be evenly used in the story.
- ***Logicality*** estimates the logical coherence of a story. A higher logicality score indicates the story is more coherent and easier to follow. Logicality is defined by the averaged likelihood of each coherent relation r_i occurred after each fact f_i in the story:

$$L(S) = \frac{1}{n-1} \sum_{r_i, f_i \in S} P(r_i|f_i) \quad (8)$$

- ***Integrity*** is the data coverage rate of $\mathcal{S}$. A larger integrity, indicates the story more comprehensively represents the input data. Integrity is defined as:

$$C(\mathcal{S}) = \frac{count(\bigcup_{i=0}^{n-1} f_i)}{\mathcal{N}} \quad (9)$$

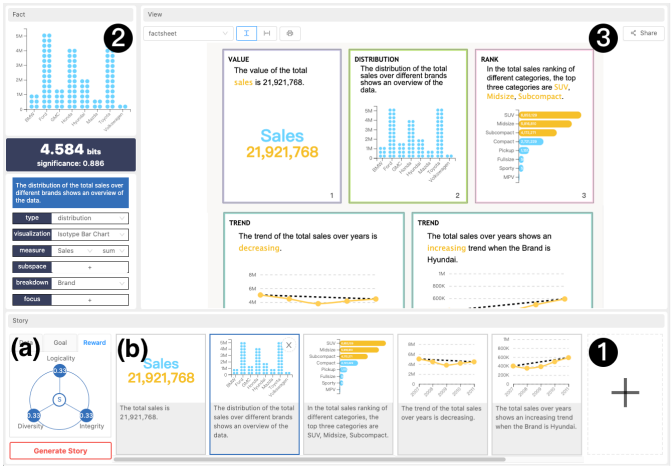

Fig. 5. The story editor of Calliope system consists of three major views: (1) the storyline view for story configuration, generation, and storyline editing, (2) the fact view for fact editing, and (3) the story visualization view for the visual data story preview and sharing (***factsheet*** mode).

where the molecule is the total number of data items in the spreadsheet used in the story, and $\mathcal{N}$ is the total number of data items.

5 Data Story Editor

In this section, we introduce the design of the story editor and the methods used for visualizing a data story.

5.1 User Interface

The story editor, as shown in Fig. 5, consists of three major views: the storyline view (Fig. 5-1), fact view (Fig. 5-2), and story visualization view (Fig. 5-3). In the storyline view, a user can upload a spreadsheet, set the story generation goal, and adjust the reward function in a group of configuration panels (Fig. 5-1(a)). The generated data facts are shown in a row (Fig. 5-1(b)), in which a user can remove a fact or change the generated narrative order based on his/her own preferences. Each fact is visualized by a chart and captioned by a generated text description (**R3**). When a fact is selected, the data details on each of its fields, importance scores, and visual and textual representations, will be shown in the fact view (**R4**). The generated data story can be visualized in the story visualization view through three visualization modes: (1) ***storyline*** mode (Fig. 1), (2) ***swiper*** mode (Fig. 6(a)), and (3) ***factsheet*** mode (Fig. 6(b)). These modes are respectively designed for representing the story on laptops/tablets, smartphones, and printouts to facilitate a flexible story communication and sharing (**R5**). A user can easily switch between different modes in the story visualization view via a drop down menu.

5.2 Visualizing a Data Story

A data story generated by the engine is visualized as a sequence of charts through two steps: *showing a data fact* and *showing a story*. The first step maps a data fact to a chart while the second step organizes a sequence of charts in an expressive layout as a story.

Showing a Data Fact Benefiting from the simple and clear definition of each fact type introduced in Section 4.1, Calliope is able to directly convert a data fact into a captioned chart that incorporates both the visual and textual representations. Specifically, the caption is generated based on the syntax defined in Table 3, and the fact is automatically visualized in two steps by following a rule-based approach. In particular, the system first selects the most frequently used chart regarding the fact type in Table 1. After that, it selects a subset of data from the input spreadsheet regarding the filters given by the ***subspace*** field and then maps the ***breakdown*** field(s) to the categorical channel(s) and the ***measure*** field(s) to the numerical channel(s) in the chart. Finally, the data values indicated by the ***focus*** field are highlighted in the chart. Fig. 5(2) illustrates an example of showing a difference fact in a captioned bar chart.

Showing a Story The story visualization view provides a variety of visualization modes to represent the generated data story for different application scenarios. In particular, a summarization is first provided to give a textual briefing of the story to help users obtain data insights at a glance. This step shows the data coverage rate, the total number of data facts in the story, and the generated textual narrative of the story. The ***storyline*** and ***swiper*** visualization modes are respectively designed to facilitate an efficient story exploration on tablets and smartphones. In particular, the captioned charts showing data facts in the story are horizontally aligned in a row to represent the narrative order in the ***storyline*** mode and are shown one at a time in the ***swiper*** mode. A user can swipe the touch screen to explore the story through these representations. Finally, the ***factsheet*** mode is designed to show the story in the form of a poster that could be easily printed out.

6 Evaluation

We evaluate Calliope system via (1) three examples to showcase the quality of the generated story, (2) two controlled experiments to estimate the usefulness and quality of a generated logic, and (3) domain expert interviews to estimate the usability of the system.

6.1 Example Stories

We demonstrate three visual data stories generated by Calliope respectively based on three real-world datasets as shown in Fig. 1 and Fig. 6, which are described as follows:

Fig. 1 shows a story generated based on a ***COVID-19*** dataset (903 rows, 5 columns). The data record the recent numbers of daily infections, deaths, and healings of COVID-19 in China from March 1st to March 21st. The generated story illustrates the daily mortality in China decreased in March (*Fact 1*), and the largest number was 42 occurred on March 2nd (*Fact 2*). Hubei was the most affected province (*Fact 3*). The total death in Hubei accounted for 97.4% of that in China (*Fact 4*), which was 423 (*Fact 5*). A large number of patients recovered in March (*Fact 6*), showing the improving situation in China.

Fig. 6(a) shows a story generated based on a ***Car Sales*** dataset (275 rows, 4 columns), including the sales records of different automobile brands around the financial crisis in 2008. The story shows that in 2007-2011, 21,921,768 cars were sold in total (*Fact 1*). The top three sellers were Ford, Toyota, and Honda (*Fact 2*). The difference was huge when comparing the best and worst sellers (*Fact 3*). Specifically, SUV was the best selling model (*Fact 4*) which sold 6,764,065 more than MPV (*Fact 5*). Generally, the sales records decreased during the final financial crisis (*Fact 6*).

Fig. 6(b) presents a dataset about ***Startup Failures***. The data (1234 rows, 6 columns) record a set of companies closed during or after the raising tide of "new economics" in China from 2010 to 2019. Each startup company is described from six criteria, including its broken year, location, industry, funded status, survival time, and the main cause of failure. The story shows that numerous companies were closed in recent years (*Fact 1*), and most of them were located in Eastern China (*Fact 2*). The most dangerous fields were e-commerce, social media, and local business (*Fact 3*). Most companies closed in these fields were still in the early stages before the series A+ round (*Fact 4*), and some even closed without receiving any investment (*Fact 5*). Regarding the reasons, "no business model" and "no market need" were the most frequently occurring problems in these startup companies (*Fact 6*).

6.2 Evaluation of the Generated Logic

We verified the usefulness and the quality of the generated logic in a story via two controlled experiments.

Experiment I: Usefulness We first estimated whether the logic generated by Calliope helps with the understanding of a data story. To this end, we compared our generation results with the factsheets generated by DataShot [51], in which a set of selected data facts was randomly organized.

Data. We collected the same datasets illustrated in Fig. 4 (C, D) in DataShot, including CarSales and SummerOlympics. Based on these two datasets, we generated two factsheets using Calliope automatically and picked two cases from the DataShot paper directly as the baseline.

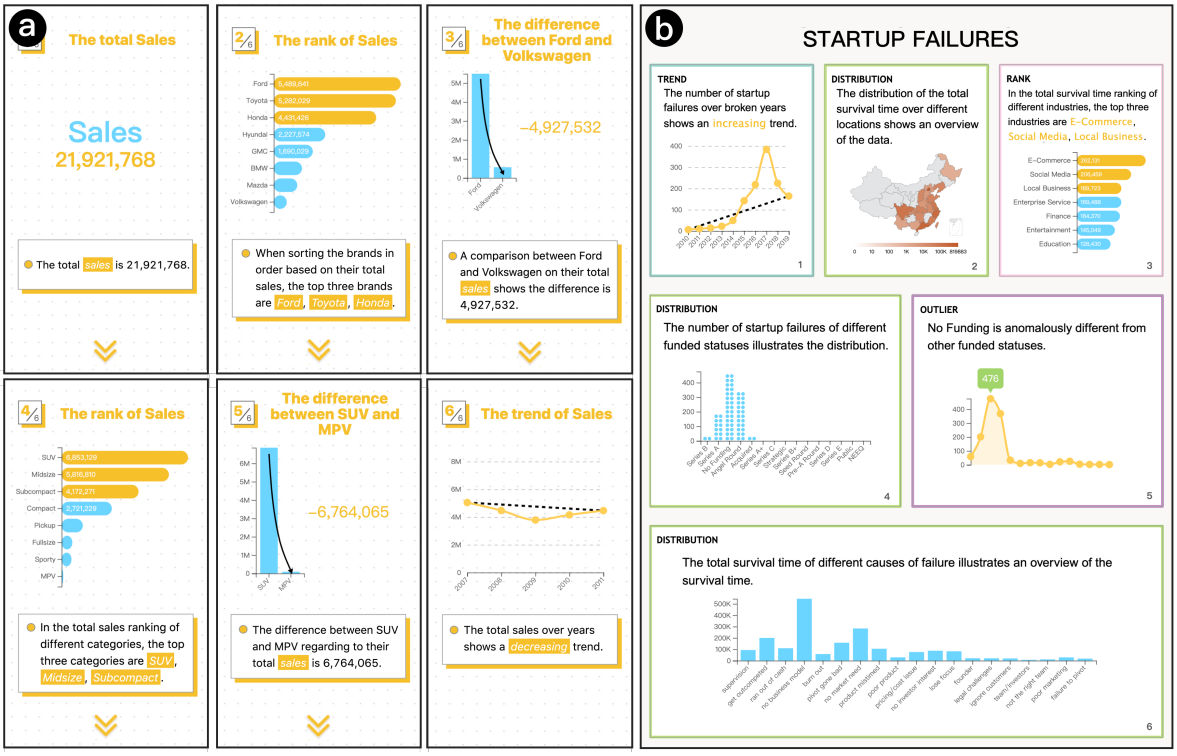

Fig. 6. Two data story examples generated by Calliope: (a) a story about car selling records around economic crisis in 2008 shown in the ***swiper*** mode and (b) a story about startup failures after the tide of "new economics" in China shown in the ***factsheet*** mode.

To ensure a more fair study setting, we made sure these generated stories from different systems contained similar data facts, and we also revised the design of each factsheet with the same style.

Procedure and Tasks. We recruited 16 college students (12 females) ages 22 - 26 years old (M=23.94, SD=1.43) as participants. Each participant was presented with two factsheets of different topic from our data, one by Calliope and one by DataShot. We counterbalanced the presentation order of the factsheets for a fair comparison. The participants were asked to read the factsheets carefully and compare two factsheets from five specific aspects including logicality, comprehension, memorability, engagement, and dissemination. The experiment lasted approximately 40 minutes for each participant.

Results. In terms of *Logicality* and *Memorability*, Calliope received more positive feedback than DataShot. One participant commented that "*I can smoothly follow it's (Calliope) logic from whole to part, as it first introduces the overall information about Olympic golds and then zooms in on specific sports and countries.*" Another participant said, "*it's much easier to remember the story generated by Calliope, as the annual car sales in different brands is presented step by step in a proper order*". Regarding *Comprehension*, *Engagement*, and *Dissemination*, Calliope performs comparably to Datashot. One participant said, "*I enjoy the simple and beautiful visualization of both factsheets and would love to share them on social media if the data is relevant.*"

Experiment II : Quality The second experiment was designed to evaluate the quality of the generated logic. To this end, we objectively estimated consistency of the logic orders respectively given by users and generated by Calliope based on the same set of data facts.

Procedure and Tasks. We first shuffled the order of a set of data facts in a visual data story generated by Calliope and then asked a group of users to restore the logic order by reading the chart and description of each fact. Finally, we checked the consistency between the human-generated logic orders and those produced by Calliope based on Kendall's τ_b correlation [22]. This measure was introduced to estimate the consistency of the element orders between two sequences, whose value lies in $[-1, 1]$ with "-1" indicating a completely reverse order and "1" indicating the orders are identical. To ensure a fair and comprehensive comparison, we generated 12 data stories based on the aforementioned three datasets, four stories per dataset. Each story contained six data facts whose orders were shuffled for the experiment.

A group of 20 participants (17 female) aged 22-30 years old ($M = 26, SD = 2.63$) were recruited for Experiment II. All of the participants reported that they have fundamental knowledge about data visualization or experience in data-oriented storytelling. The experiments started by a brief introduction about the data, and the participants were asked to reorder the data facts of all the aforementioned 12 shuffled stories via an interactive user interface. We also encouraged the participants to fully explore the data and try their best to understand the data insights represented by each data fact.

Results. We calculated the average Kendall's τ_b value on each dataset, and the result showed that the logic orders generated by Calliope were consistent with those generated by our participants (***Car Sales***: $\mu = 0.487, \sigma = 0.29$; ***COVID-19***: $\mu = 0.648, \sigma = 0.327$; ***Startup Failures***: $\mu = 0.63, \sigma = 0.295$). We also leveraged the sequences generated by humans as a ground truth and calculated Kendall's τ_b values between the random results and human generated results as the baseline. As a result, the baseline is 0.015, which indicates these two orders are relatively irrelevant (as the value is close to 0). By comparing the τ_b values, we found that the logic orders generated by Calliope are much more consistent to humans than the baseline.

6.3 Expert Interview

To further evaluate the usability of Calliope system, a series of interviews with three groups of expert users from different areas were performed. The first group included four data journalists (denoted as J1-J4) from different news medias in China. They had over 3.5 years' working experiences on average, and were very familiar with the data-oriented storytelling and had technical skills for creating a visual data story. The second group comprised three data analysts (denoted as D1-D3) from an international IT consultant company. Their major job was to analyze customers' data and write analysis reports. BI tools such as Tableau and PowerBI were frequently used in their daily work. The third group consisted of three senior visualization researchers (denoted as V1-V3), all of whom had experiences on publishing papers in major visualization conferences such as IEEE VIS and EuroVis.

Procedure The interviews were performed via an online meeting system. Each interview started by a 10-minutes introduction about the system. After that, the experts were encouraged to use the online Calliope system on their own. After fully explored the functions of the system, the experts were asked to generate a data story based on one of the three datasets introduced in Experiment II. To arouse their interests, we let the journalists explore the ***COVID-19*** dataset as it is a recent and important news topic. We let the data analysts use the ***Car Sales*** data given its similarity to the business data analyzed in their work. We let the visualization researchers explore the ***Startup Failures*** data as it is the most complex one. They were also encouraged to edit the generated story and share their findings with us. The experts were asked to finish a questionnaire, followed by an interview after they fully explored the functionalities of Calliope system. The whole process lasted for about one hour and the interviews were recorded for later analysis.

Results Fig. 7 shows that questionnaire results, Calliope was rated relatively high in terms of the generated story, visualization, and system. A large number of positive comments were recorded during the interview. We summarized the interview results due to the page limitation.

Data Story. All the experts agreed that the generated data story was able to express useful data insights. The visualization researcher V2 mentioned that "*the data facts in the story are quite clear and are well-organized*". J1 commented that "*the story starts from an overview, followed by a series of data details, ..., It's the way we frequently adopted when writing a news story. It's amazing that now it can be automatically generated*". J3 also noted that *"the logical order [of the story] can help readers get into the points"*. D1 observed that the ordered data facts can help with the efficient exploration of the data, *"[Automatically] showing the appropriate dimensions and measures in a sequence of visualizations can definitely guide the data exploration, ..., it's helpful when you have no idea about how to get started"*.

Visualization. In terms of visualization, all the experts were satisfied with the design of three visualization modes in Calliope, *"it's a very nice and thoughtful design"* (J1-J4). Especially, they felt that the ***storyline*** mode provides a good overview (J1-J3, D1, D3, V1, V2). All the experts believed that the ***swiper*** mode was neat and helpful when viewing the story on a smartphone while the ***factsheet*** mode supports easy printing of a story. All the journalists felt that editing was an important function, *"it (editing function) allows us to create a high-quality story quickly based on the generation results"* (J1). They also felt generating stories by interactively changing the reward was *"interesting"* and *"inspiring"*. J2 suggested that *"we usually write stories from different perspectives, and it can facilitate my ideation process"*.

System. Most experts (J1-J3, D1, D3, V2, V3) mentioned that the system was useful for users who are not skilled at data analysis or visualization. The data journalists J1-J4 especially appreciated the efficient story generation and editing function of Calliope. J4 said, *"with this tool I can quickly create a story by first generating a draft and then revising it accordingly"*. The data analysts felt the system is powerful to help them efficiently preview an unknown data, *"with this tool, I can quickly find where to start when getting a [new] spreadsheet"* (D1). The visualization researchers believed that our system lowers the barriers of creating a data story. V3 said, "when compared with other data story authoring tools, this system is much more smart as it requires limited knowledge about data analysis and visualization design".

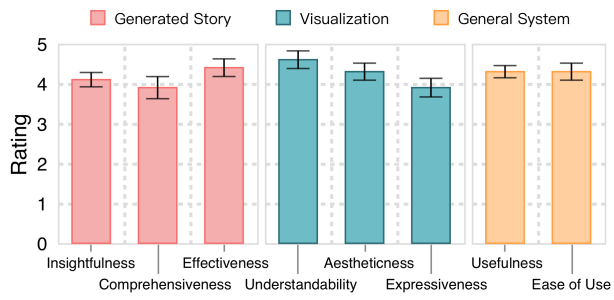

Fig. 7. The ratings of generated story, visualization design, and system from different criteria based on a 5-point Likert scale given by 10 expert users, where 5 is the best and 1 is the worst.

7 Limitations and Future Work

Despite the above positive results from the evaluation, we would also like to summarize and discuss several limitations that was found during the design and implementation process and mentioned by our expert users during the interview. We hope highlight these limitations will help point out several potential future research directions and inspire new studies by following our work.

Supporting a Better Textual Narrative. During the interview, many data journalists (J1,J2,J4) felt the generated captions were too rigid to be used especially in a data news. More diverse and insightful descriptions were desired. Moreover, the current results also contained some grammar errors, which also need to be addressed (J1,2 D1, V2,3)). However, all of them acknowledged that the current results, although unsatisfactory, were still useful for a rapid preview and briefing of the input data. In the future, it is necessary to leverage more advanced techniques introduced in the field of nature language processing to generate text narratives in the data story in a higher quality.

Understanding Data Semantics. After using Calliope, although impressed, some experts (J1-J3, V2) expect a more intelligent tool that can even understand the semantics of the data to better generate the story content and logic. We acknowledge this is a key limitation of the current system and understanding the underlying semantics of the data is critical for generating a meaningful and insightful data story. This is a promising research direction that is worth a further exploration. To address the issue, one could leverage or develop more advanced AI techniques or could also introduce sophisticated interactive feedback mechanism to keep user in the generation loop and leverage human intelligence to steer data quality [27] and guide the underlying generation algorithm/model to better understand data semantics.

Enriching Visualization. Several experts (J3, D1, D2) would like to have a slides mode and dashboard mode to support more application cases. J1 and V3 also pointed out that some of the current generated visual encoding are notably simple, and a chart should encode more information at a time. For example, when showing a line chart, the size of point could also be used to encode data, and a stack bar chart could be used to show an additional categorical field in the data. In addition, Calliope cannot deal with hierarchical or relational datasets, which are also desired functions (V1,V2). Providing more advanced visualization representations for a story is also a valuable future work.

Performance Issues. The current system design and implementation have some performance bottlenecks that worth a future study. In particular, the calculation of data fact significance, i.e., $S(f_i)$ in Formula (1) consists of statistical computations, which is usually slow and thus limits the number of facts that can be explored in each searching iteration, thus affecting the generation quality.

8 Conclusion

We have presented Calliope, a novel system designed for automatically generating visual data stories from a spreadsheet. The system incorporates a novel logic-oriented Monte Carlo tree search algorithm to create a data story by gradually generating a sequence of data facts in a logical order while exploring the data space. The importance of a fact is measured by its information quantity and its statistical significance. Each fact in the story is visualized in a chart with an automatically generated caption. A story editor is introduced in the system to facilitate the easy and efficient editing of a generated story. The proposed technique was evaluated via three example cases, two controlled experiments, and a series of interviews with 10 expert users from different domains. The evaluation showed the power of Calliope system and revealed several limitations of the current system, which will be addressed in the future.

Acknowledgments

Nan Cao is the corresponding author. This work was supported by NSFC 62072338 and NSF Shanghai 20ZR1461500.

REFERENCES

[1] F. Amini, N. H. Riche, B. Lee, A. Monroy-Hernandez, and P. Irani. Authoring data-driven videos with dataclips. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):501–510, 2016.

[2] A. Belz. Automatic generation of weather forecast texts using comprehensive probabilistic generation-space models. *Natural Language Engineering*, 14(4):431–455, 2008.

[3] M. A. Borkin, Z. Bylinskii, N. W. Kim, C. M. Bainbridge, C. S. Yeh, D. Borkin, H. Pfister, and A. Oliva. Beyond memorability: Visualization recognition and recall. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):519–528, 2015.

[4] M. Brehmer, B. Lee, N. H. Riche, D. Tittsworth, K. Lytvynets, D. Edge, and C. White. Timeline storyteller: The design & deployment of an interactive authoring tool for expressive timeline narratives. In *Proceedings of the Computation + Journalism Symposium*, pp. 1–5, 2019.

[5] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton. A survey of Monte Carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1):1–43, 2012.

[6] S. Chen, J. Li, G. Andrienko, N. Andrienko, Y. Wang, P. H. Nguyen, and C. Turkay. Supporting story synthesis: Bridging the gap between visual analytics and storytelling. *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2019.

[7] Y. Chen, J. Yang, and W. Ribarsky. Toward effective insight management in visual analytics systems. In *2009 IEEE Pacific Visualization Symposium (PacificVis)*, pp. 49–56. IEEE, 2009.

[8] W. Cui, X. Zhang, Y. Wang, H. Huang, B. Chen, L. Fang, H. Zhang, J.-G. Lou, and D. Zhang. Text-to-viz: Automatic generation of infographics from proportion-related natural language statements. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):906–916, 2019.

[9] Z. Cui, S. K. Badam, M. A. Yalçin, and N. Elmqvist. Datasite: Proactive visual data exploration with computation of insight-based recommendations. *Information Visualization*, 18(2):251–267, 2019.

[10] Ç. Demiralp, P. J. Haas, S. Parthasarathy, and T. Pedapati. Foresight: Recommending visual insights. *Proceedings of the VLDB Endowment*, 10(12):1937–1940, 2017.

[11] V. Dibia and Ç. Demiralp. Data2vis: Automatic generation of data visualizations using sequence-to-sequence recurrent neural networks. *IEEE Computer Graphics and Applications*, 39(5):33–46, 2019.

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

[13] L. Dou, G. Qin, J. Wang, J.-G. Yao, and C.-Y. Lin. Data2text studio: Automated text generation from structured data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 13–18, 2018.

[14] A. Fan, M. Lewis, and Y. Dauphin. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 889–898, 2018.

[15] A. Fan, M. Lewis, and Y. Dauphin. Strategies for structuring story generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2650–2660, 2019.

[16] D. Galanis and I. Androutsopoulos. Generating multilingual descriptions from linguistically annotated owl ontologies: the naturalowl system. In *Proceedings of the 11th European Workshop on Natural Language Generation (ENLG 07)*, pp. 143–146, 2007.

[17] D. Gotz, Z. When, J. Lu, P. Kissa, N. Cao, W. H. Qian, S. X. Liu, and M. X. Zhou. Harvest: an intelligent visual analytic tool for the masses. In *Proceedings of the First International Workshop on Intelligent Visual Interfaces for Text Analysis*, pp. 1–4, 2010.

[18] S. Gratzl, A. Lex, N. Gehlenborg, N. Cosgrove, and M. Streit. From visual exploration to storytelling and back again. *Computer Graphics Forum*, 35(3):491–500, 2016.

[19] K. Hu, M. A. Bakker, S. Li, T. Kraska, and C. Hidalgo. Vizml: A machine learning approach to visualization recommendation. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–12, 2019.

[20] J. Hullman, S. Drucker, N. H. Riche, B. Lee, D. Fisher, and E. Adar. A deeper understanding of sequence in narrative visualization. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2406–2415, 2013.

[21] P. Jain, A. Laha, K. Sankaranarayanan, P. Nema, M. M. Khapra, and S. Shetty. A mixed hierarchical attention based encoder-decoder approach for standard table summarization. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pp. 622–627, 2018.

[22] M. G. Kendall. The treatment of ties in ranking problems. *Biometrika*, 33(3):239–251, 1945.

[23] B. Lee, N. H. Riche, P. Isenberg, and S. Carpendale. More than telling a story: Transforming data into visually shared stories. *IEEE Computer Graphics and Applications*, 35(5):84–90, 2015.

[24] B. Li, S. Lee-Urban, G. Johnston, and M. Riedl. Story generation with crowdsourced plot graphs. In *AAAI Conference on Artificial Intelligence*, pp. 598–604, 2013.

[25] J. Li, L. Bing, L. Qiu, D. Chen, D. Zhao, and R. Yan. Learning to write stories with thematic consistency and wording novelty. In *AAAI Conference on Artificial Intelligence*, pp. 1715–1722, 2019.

[26] D. Liu and G. Liu. A transformer-based variational autoencoder for sentence generation. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7. IEEE, 2019.

[27] S. Liu, G. Andrienko, Y. Wu, N. Cao, L. Jiang, C. Shi, Y.-S. Wang, and S. Hong. Steering data quality with visual analytics: The complexity challenge. *Visual Informatics*, 2(4):191–197, 2018.

[28] Y. Luo, X. Qin, N. Tang, and G. Li. Deepeye: towards automatic data visualization. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pp. 101–112. IEEE, 2018.

[29] J. Mackinlay, P. Hanrahan, and C. Stolte. Show me: Automatic presentation for visual analysis. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1137–1144, 2007.

[30] J. Mahapatra, S. K. Naskar, and S. Bandyopadhyay. Statistical natural language generation from tabular non-textual data. In *Proceedings of the 9th International Natural Language Generation Conference*, pp. 143–152, 2016.

[31] L. J. Martin, P. Ammanabrolu, X. Wang, W. Hancock, S. Singh, B. Harrison, and M. O. Riedl. Event representations for automated story generation with deep neural nets. In *AAAI Conference on Artificial Intelligence*, pp. 868–875, 2018.

[32] A. Mishra, A. Laha, K. Sankaranarayanan, P. Jain, and S. Krishnan. Storytelling from structured data and knowledge graphs: An nlg perspective. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (Volume 4: Tutorial Abstracts)*, pp. 43–48, 2019.

[33] D. Moritz, C. Wang, G. L. Nelson, H. Lin, A. M. Smith, B. Howe, and J. Heer. Formalizing visualization design knowledge as constraints: Actionable and extensible models in draco. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):438–448, 2018.

[34] H. Ramezani et al. A note on the normalized definition of shannon's diversity index in landscape pattern analysis. *Environment and Natural Resources Research*, 2(4):54–60, 2012.

[35] D. Ren, M. Brehmer, B. Lee, T. Höllerer, and E. K. Choe. Chartaccent: Annotation for data-driven storytelling. In *2017 IEEE Pacific Visualization Symposium (PacificVis)*, pp. 230–239. IEEE, 2017.

[36] M. O. Riedl and R. M. Young. Narrative planning: Balancing plot and character. *Journal of Artificial Intelligence Research*, 39(1):217–268, 2010.

[37] A. Satyanarayan and J. Heer. Authoring narrative visualizations with ellipsis. *Computer Graphics Forum*, 33(3):361–370, 2014.

[38] A. Satyanarayan, B. Lee, D. Ren, J. Heer, J. Stasko, J. Thompson, M. Brehmer, and Z. Liu. Critical reflections on visualization authoring systems. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):461–471, 2019.

[39] U. Schaechtle, K. Stathis, and S. Bromuri. Multi-dimensional causal discovery. In *Twenty-Third International Joint Conference on Artificial Intelligence*, pp. 1649–1655, 2013.

[40] E. Segel and J. Heer. Narrative visualization: Telling stories with data. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1139–1148, 2010.

[41] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948.

[42] D. Shi, Y. Shi, X. Xu, N. Chen, S. Fu, H. Wu, and N. Cao. Task-oriented optimal sequencing of visualization charts. In *2019 IEEE Visualization in Data Science (VDS)*, pp. 58–66, 2019.

[43] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driess-

che, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[44] A. Srinivasan, S. M. Drucker, A. Endert, and J. Stasko. Augmenting visualizations with interactive data facts to facilitate interpretation and communication. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):672–681, 2018.

[45] R. Swanson and A. S. Gordon. Say anything: Using textual case-based reasoning to enable open-domain interactive storytelling. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 2(3):1–35, 2012.

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

[47] C. Tong, R. Roberts, R. Borgo, S. Walton, R. Laramee, K. Wegba, A. Lu, Y. Wang, H. Qu, Q. Luo, and X. Ma. Storytelling and visualization: An extended survey. *Information*, 9(3):65, 2018.

[48] R. Turner, Y. Sripada, and E. Reiter. Generating approximate geographic descriptions. In *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 09)*, pp. 42–49, 2009.

[49] M. Vartak, S. Rahman, S. Madden, A. Parameswaran, and N. Polyzotis. Seedb: Efficient data-driven visualization recommendations to support visual analytics. *Proceedings of the VLDB Endowment*, 8(13):2182–2193, 2015.

[50] Q. Wang, Z. Li, S. Fu, W. Cui, and H. Qu. Narvis: Authoring narrative slideshows for introducing data visualization designs. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):779–788, 2018.

[51] Y. Wang, Z. Sun, H. Zhang, W. Cui, K. Xu, X. Ma, and D. Zhang. Datashot: Automatic generation of fact sheets from tabular data. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):895–905, 2019.

[52] Y. Wang, H. Zhang, H. Huang, X. Chen, Q. Yin, Z. Hou, D. Zhang, Q. Luo, and H. Qu. Infonice: Easy creation of information graphics. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pp. 1–12, 2018.

[53] B. Wellner, J. Pustejovsky, C. Havasi, A. Rumshisky, and R. Saurí. Classification of discourse coherence relations: An exploratory study using multiple knowledge sources. In *Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue*, pp. 117–125, 2006.

[54] F. Wolf and E. Gibson. Representing discourse coherence: A corpus-based study. *Computational Linguistics*, 31(2):249–287, 2005.

[55] J. Xu, X. Ren, Y. Zhang, Q. Zeng, X. Cai, and X. Sun. A skeleton-based model for promoting coherence among sentences in narrative story generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 4306–4315, 2018.

[56] L. Yao, N. Peng, R. Weischedel, K. Knight, D. Zhao, and R. Yan. Plan-and-write: Towards better automatic storytelling. In *AAAI Conference on Artificial Intelligence*, pp. 7378–7385, 2019.

[57] R. Ye, W. Shi, H. Zhou, Z. Wei, and L. Li. Variational template machine for data-to-text generation. In *International Conference on Learning Representations (ICLR)*, 2020.