Technical Disclosure Commons

Defensive Publications Series

January 2020

METHODS AND APPARATUS FOR INTENT AND CONTEXT-BASED AD-HOC DATA DRIVEN REPORTING AND VISUALIZATION

Plamen Nedeltchev

Madhuri Garikapati

Eric Kroll

Mani Kesavan

METHODS AND APPARATUS FOR INTENT AND CONTEXT-BASED AD-HOC DATA DRIVEN REPORTING AND VISUALIZATION

AUTHORS:

Plamen Nedeltchev
Madhuri Garikapati
Eric Kroll
Mani Kesavan

ABSTRACT

Existing executive events, investment planning sessions, regular checkups, service reviews, etc. are mainly based on static content such as templates, PDF files and PPT presentations. However, these templates have static dashboard and predefined scenarios that cannot be changed/altered during the presentation and lack the ability to visually-represent relevant ad-hoc content. As such, presented herein are dynamic (ad-hoc) and context-based visualization systems that are more intuitive than conventional arrangements. The techniques presented herein may, for example, facilitate a more natural decision making process by providing content that can dynamically change based, for example, on the current context, such as important in-depth questions supporting a decision making process, etc. In one example, a user may ask a follow up question and the system is configured to store the context and provide a recursive answer. The system may offer the next best question, based on an embedded convolutional sequence. The techniques presented herein may also use generative models and graph classifiers to convert data into a visual representation.

DETAILED DESCRIPTION

The rapidly increasing need for data-driven decision making is driving decision makers (users) to embrace conventional Business Intelligence (BI) tools to visualize and explore data. However, most BI tools/dashboards fail to reach their intended outcome and lack user adoption because of three (3) main reasons:

- **Dashboards contain too much or too little data:** The dashboards are either excessively loaded with content or fail to provide a complete picture on a specific topic due, for example, to a lack of screen space. Rarely are the dashboards or static content designed following the principles of "white spaces," which experts believe are the secret of effective design and representation of data. As a result, the user or a decision maker is left either overwhelmed or under-informed.
- **Dashboards inherently come with a learning curve**. Users are expected to know how the information hierarchy is laid out before they can navigate to their point of interest. Moreover, users are expected to translate their business question into a series of data questions (e.g., "What fields are of interest to me?;" "How can I slice the data?;" "How can I subset the data?;" "How do I aggregate vs. decompose the data?;" etc.). To determine what is relevant and important, a user has to sift through page-views, drop-down menus, radio buttons and understand the underlying definitions.
- **Dashboards are static**. Answers to various business questions are pre-determined by the content creator. Decision makers cannot find answers to questions that not already built into the original dashboard. Anything that is not prepared or developed in the constraints of the static content typically will require another meeting, extra time, or different set of people. As such, these meetings will often conclude with – "I will get back to you on ..(topic, number, reference, quote).

In other words, with the ever-growing quantity of available data and multi-dimensional relationships in data, it is difficult for executive decision makers to create and understand graphical representations of data that facilitate the decision making process The techniques presented herein outline an intent-driven, interactive data visualization tool that: uses language query models to decipher intent, uses Convolutional Sequence Embedding to predict Next Best Question, and uses generative models to generate various visualizations as the dimensionality and relationships change in real time. Such an interactive data visualization tool that allows dynamic, per-need, ad-hoc and context based navigation of a content facilitates a more natural decision making process that is superior to pre-defined dashboards, templates and PowerPoint presentations. Advancements in artificial intelligence (AI) technologies can be used to enable new and useful capabilities that could change the way people prepare, present and make decisions based on a dynamically available content.

FIG. 1, below, illustrates an example workflow for end-to-end use of an interactive data visualization tool, in accordance with the techniques presented herein, that is configured to detect and utilize user "intent" for generation of graphical displays or other visualization content. It is noted that the word "intent" is used in the context of ML/DL and it should be interpreted as a sequence of digital tokens to label classified data (i.e., "intent"= sequence of digital tokens to label classified data).

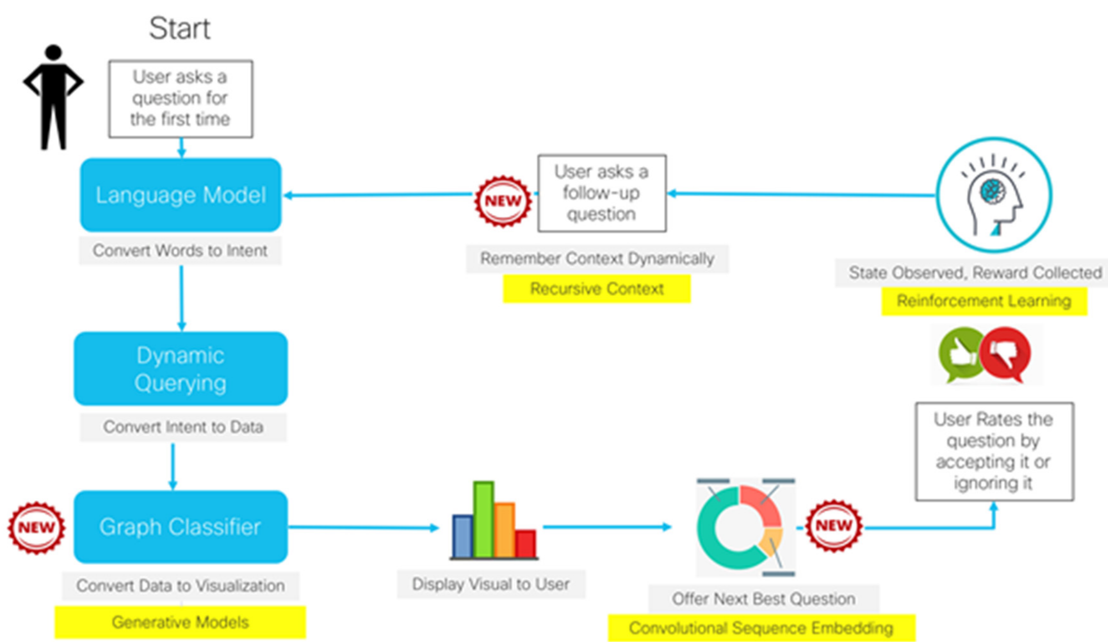

FIG. 1

In other words, with the ever-growing quantity of available data and multi-dimensional relationships in data, it is difficult for executive decision makers to create and understand graphical representations of data that facilitate the decision making process. The techniques presented herein outline an intent-driven, interactive data visualization tool that: uses language query models to decipher intent, uses Convolutional Sequence Embedding to predict Next Best Question, and uses generative models to generate various visualizations as the dimensionality and relationships change in real time. Such an interactive data visualization tool that allows dynamic, per-need, ad-hoc and context based navigation of a content facilitates a more natural decision making process that is superior to pre-defined dashboards, templates and PowerPoint presentations. Advancements in artificial intelligence (AI) technologies can be used to enable new and useful capabilities that could change the way people prepare, present and make decisions based on a dynamically available content.

In FIG. 1, the “Language Model” block detects user intent (keywords) and the “Dynamic Querying” block converts keywords into a query language. The three (3) novel components in this architecture are the “Dynamic, Recursive Context block,” the “Next Best Visualization block,” and the “Graph Classifier block,” which converts data into visualizations. The goal of the Dynamic, Recursive Context block is to extend the context lifespan, dynamically, based on user requests in a conversation.

That is, every conversation has a context and, in a typical conversation, a context is associated with a lifespan, which is the number of follow-up questions that a user can ask in relation to an original question. Traditional conversational software offers context lifespan to be set as a numeric parameter and expect that the designer of the conversational interface will know how long a context will likely last. For example, with certain conventional conversational software, contexts may have a default lifespan of two requests. Intents that renew the context will reset the counter and clock to give an additional five requests and ten minutes. This approach works in 'happy scenarios', such as when the conversation is flowing linearly and the user is able to achieve his/her goal with only a few tries. The problem with this approach is that, in a real conversational scenario of decision making, it is difficult to find an optimal context lifespan and pre-set it into an intent. A parameterized approach can make the conversation unpredictable after a few requests.

For example, in such arrangements, if the parameter is set too low, then the initial context is forgotten within just a few interactions. If the parameter is set too high, then context may be maintained after the user no longer cares about the questions. In the example shown in FIG. 2, below, the user is penalized for asking the wrong sequence of questions.

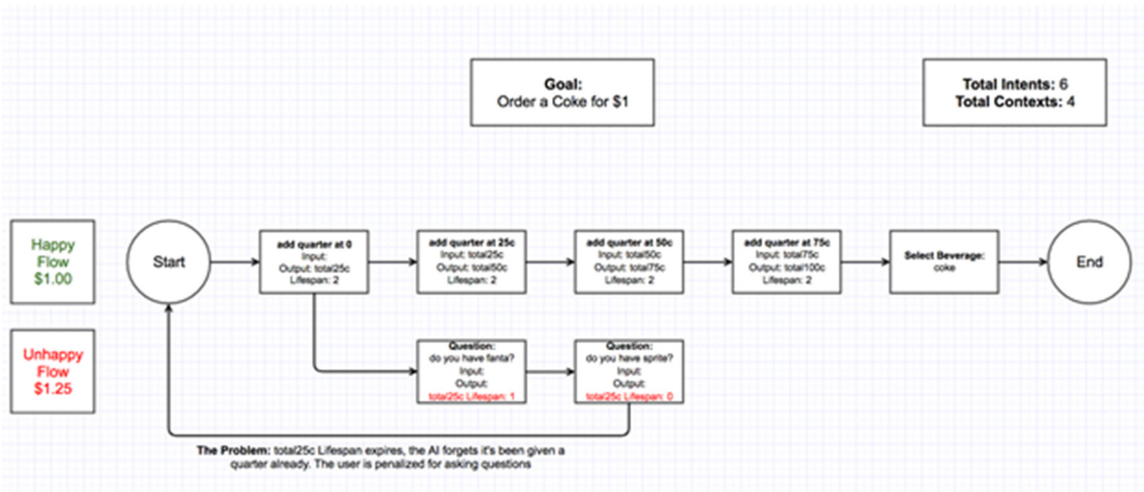

FIG. 2

As noted above, the techniques presented herein introduce the concept of a 'Recursive Context' block that maintains or discards context based on user interactions. The lifespan of the context is set to only 1 interaction, with context being refreshed by each additional question in the same vein. This allows users to ask questions indefinitely without context expiring at an arbitrary point. Context is only reset when users ask a new question in a different vein, instead of refining their previous question. This is supplemented by backend memory that stores parameters from questions and guards against parameters overwriting themselves.

As shown in FIG. 3, below, for the example stated above, the techniques presented herein retain the original context regardless of the sequence in which user conversation flows.

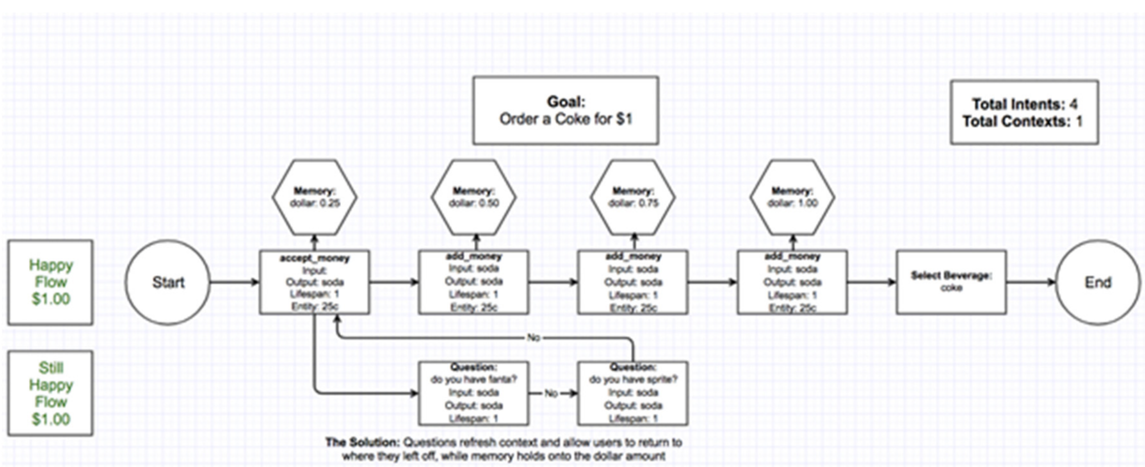

FIG. 3

As noted above, the techniques presented herein also include a "Next Best Question" feature. That is, a successful data exploration starts with asking the right series of questions. The goal of this "Next Best Question" feature is to enable users to ask the most meaningful sequence of questions from data.

The problem with pre-built visualizations is that it forces the users to think about a subject in a linear, pre-determined hierarchy. That implies that only a pre-determined set of questions could be answered, in a pre-determined sequence. Consider an example with a "Superstore Dataset" shown below:

	Sales	**Region**	**Product**
1	$400,000	North	Office Supplies
2	$350,000	South	Technology
3	$700,000	East	Furniture
4	$400,000	North	Office Supplies
5	$350,000	South	Technology
6	$700,000	East	Furniture
7	$600,000	West	Books

A pre-built visualization will either aggregate Sales by Region first, and then allow user to drill-down by Product, or vice versa, as shown below in FIG. 4.

FIG. 4

In the above example: for 2 categories, there are 2 sequences possible; for 3 categories, there are 6 sequences possible; and for 4 categories, there are 24 sequences possible. The number of combinations quickly grows to **n! (n-factorial)** for a n-dimensional space. Therefore, anytime the number of dimensions exceeds 3, the complexity of pre-creating the combinations rises exponentially. FIG. 5, below, shows how the data needs of different user groups could be very different from each other. Boxing them in a single dashboard might help a certain user group, but could be very limiting for others.

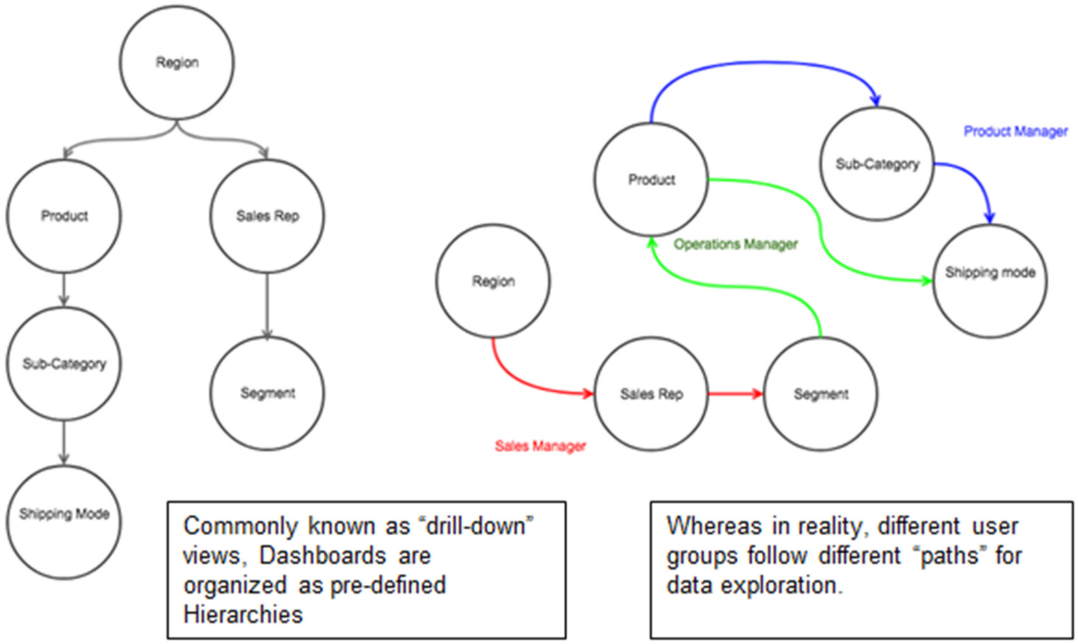

FIG. 5

The techniques presented herein address these limitations through a flexible approach for building successive data visualizations that could be triggered, for example. by one of the following scenarios:

- The user knows what to ask "show me"
- The user does not know what to ask "guide me"

In the first scenario in which the user knows what to ask, the user can specify the next “drill down” of data that he/she wishes to see in relation to an existing data visualization. The new view will be tailored to the user’s explicit request, based on prior context and new intent expressed by the user. In the second scenario in which the user doesn't know what to ask, the system can anticipate the user's needs and recommend a “next best question” to ask. As shown below in FIG. 6, in the world of data exploration, the next question depends on three (3) key factors, namely:

- Users General Preferences
- Recently asked questions
- Answer to the last question

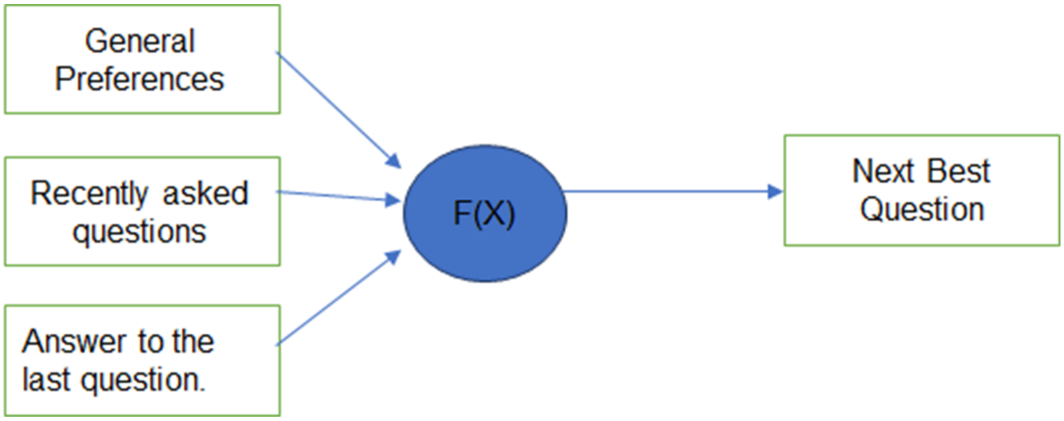

FIG. 6

For example, a first user, referred to as "John," occasionally attends musical concerts. He may have the following conversation, with questions and responses:

- "How is the weather in San Francisco today?"
  - It is cloudy and 67 degrees F.
- What will the weather be like over the weekend?
  - It will be sunny and 72 degrees F this weekend."

For example, a first user, referred to as "John," occasionally attends musical concerts. He may have the following conversation, with questions and responses:

In this example, the "Next Best Question" (predicted) could be:

- "What musical concerts are happening in San Francisco this weekend?"

The above example is represented in FIG. 7, below, where a meaningful dialogue involves long term preferences, recently asked questions, and answer to the last question

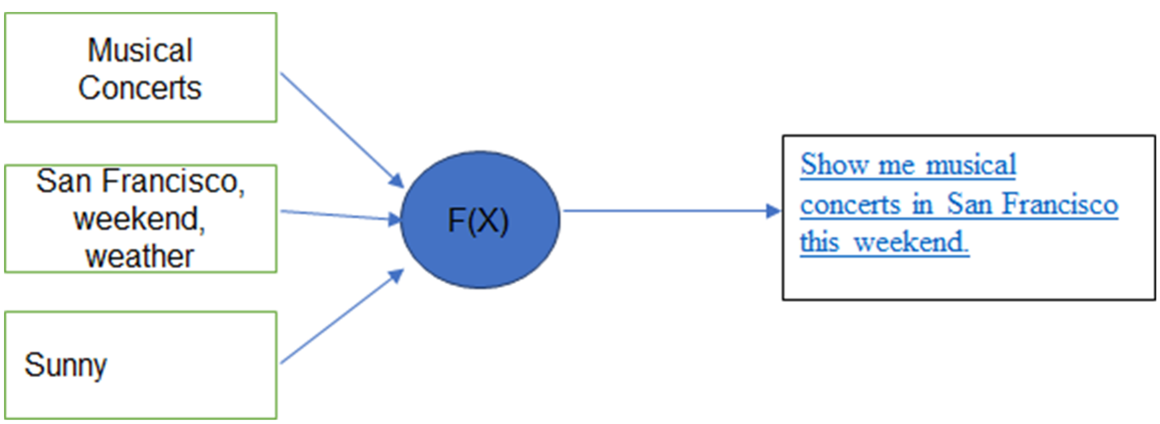

FIG. 7

In accordance with the techniques presented herein, a Top-N Sequential Recommendation algorithm via Convolution Sequence Embedding may be employed. The proposed algorithm differs from traditional approaches of recommendations in the following ways:

First, this approach provides a unified and flexible network structure for capturing many important features of sequential recommendation, i.e., point-level and union-level sequential patterns, skip behaviors, and long term user preferences. This is generally shown below in FIG. 8.

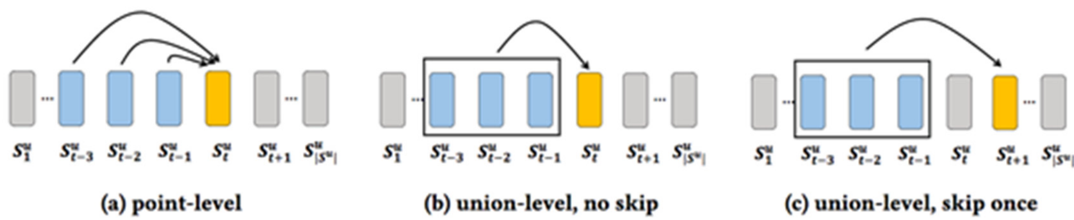

FIG. 8

Second, traditional recommendation systems, e.g., top-N recommendation, recommend the items based on a users general, long-term preferences without paying attention to context (i.e., how recent the particular items may be). Third, the algorithm incorporates the Convolution Neural Network (CNN) to learn sequential features, and Latent Factor Model (LFM) to learn user-specific features.

Fourth, the CNN is trained for each user (u) through extraction of every L successive questions as input, as well as their next T questions as the targets from the user's sequence S. This is implemented via a sliding a window of size L+T over the user's sequence, and each window generates a training instance for u denoted by a triplet (u, previous L questions, next T questions). This approach enables the use of convolution filters to search for sequential patterns, while borrowing from the idea of using CNN in text classification. This approach regards the L x d matrix E as the "image" of the previous L items in the latent space and regard sequential patterns as local features of this "image."

Fifth, the algorithm has the ability to provide a "Next Best Question" recommendation. For example, the algorithm may recommend the N items that have the highest values in the output layer y.

The techniques presented herein employ a Graph Classifier that uses generative models. These aspects are designed to learn rules for data visualizations that best fit the underlying data.

More specifically, different data characteristics require different visualization techniques for effectively studying the underlying patterns in the data. Deciding which chart to use for which data is hard for non-technical, business users and therefore they need an alternate approach. With the growing advancements in data sources, volume, and AI/ML, the traditional approach for hard-coding these rules is not scalable. The following drawbacks exist:

1. Data is dynamic, and the rules are too static.
2. Visualizations by themselves don't explain the insights. There is a need to layer visualizations on top of each other to make better sense of data. For example, a time series graph of 5 variables can be drawn on 5 line curves but there is no annotation to explain the correlation between the peaks and valleys. Business users need those insights to make sense of data. The techniques presented herein propose a solution to generate such overlaying visualizations dynamically at request, in context using generative models.
3. In fields such as IoT, data is largely multi-dimensional. As the dimensions grow the pre-existing visualizations quickly become irrelevant.

FIG. 9, below, illustrates an example correlation graph.

FIG. 9

The techniques presented herein address the above issues through the use of Generative Models that are composed of encoder-decoder layers of re-current neural networks (RNNs). Generative models help in the following ways:

1. They can help decide what chart to use.
2. They evolve as the data evolves, in volume and in dimensionality.
3. They can help add layers on existing visualization to help make sense of data.