Electronic Acknowledgement Receipt

EFS ID:	29110612
Application Number:	62500999
International Application Number:	
Confirmation Number:	1046
Title of Invention:	Applying Pragmatics Principles for Interaction with Visual Analytics
First Named Inventor/Applicant Name:	Md Enamul Hoque Prince
Customer Number:	24341
Filer:	David Vincent Sanker/Kari Aguiar
Filer Authorized By:	David Vincent Sanker
Attorney Docket Number:	061127-5058-P2
Receipt Date:	03-MAY-2017
Filing Date:	
Time Stamp:	19:56:45
Application Type:	Provisional

Payment information:

Submitted with Payment	yes
Payment Type	DA
Payment was successfully received in RAM	$260
RAM confirmation Number	050417INTEFSW00006044500310
Deposit Account	
Authorized User	

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		061127-5058-P2_Provisional_Application.pdf	2281187 887aeca0f44051deb04a7f53e25c597c50f4210b	yes	11

Multipart Description/PDF files in .zip description

Document Description	Start	End
Provisional Cover Sheet (SB16)	1	1
Specification	2	11

Warnings:

Information:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
2	Fee Worksheet (SB06)	fee-info.pdf	30174 188a37b9dab894ae33c7758560ede43774613ea8	no	2

Warnings:

Information:

Total Files Size (in bytes):	2311361

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

<u>New Applications Under 35 U.S.C. 111</u>
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

<u>National Stage of an International Application under 35 U.S.C. 371</u>
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

<u>New International Application Filed with the USPTO as a Receiving Office</u>
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

Electronically filed May 3, 2017

PROVISIONAL APPLICATION FOR PATENT COVER SHEET

This is a request for filing a PROVISIONAL APPLICATION FOR PATENT under 37 CFR 1.53(c).

Docket No.	061127-5058-P2

INVENTOR(s)

GIVEN NAME (first and middle, if any)	LAST NAME	RESIDENCE (City and either State or Foreign Country)
Md Enamul Hoque	Prince	Vancouver, British Columbia, Canada
Vidya Raghavan	Setlur	Portola Valley, California
Melanie Karla	Tory	Palo Alto, California
Isaac James	Dykeman	Bethesda, Maryland

☐ Additional inventors are being named on separately numbered sheets attached hereto.

TITLE OF THE INVENTION (500 characters max):

APPLYING PRAGMATICS PRINCIPLES FOR INTERACTION WITH VISUAL ANALYTICS

CORRESPONDENCE ADDRESS:

MORGAN, LEWIS & BOCKIUS LLP
Customer Number: **24341**

ENCLOSED APPLICATION PARTS (*check all that apply*)

☐ Application Data Sheet. See 37 CFR 1.76
☐ CD(s), number of CDs
☒ Specification *Number of pages* 10
☐ Other (specify)
☐ Drawing(s) *Number of sheets*

Fees Dues: Filing Fee of $260 ($130 for small entity). If the specification and drawings exceed 100 sheets of paper, an application size fee is also due, which is $400 ($200 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).

METHOD OF PAYMENT (*check one*)

☐ Applicant claims small entity status, see 37 CFR §1.27

$260
Total Fee Amount ($)

☒ The Director is hereby authorized to charge the filing fee and application size fee (if applicable or credit any overpayment to Deposit Account Number 50-0310 (order no. 061127-5058-P2).

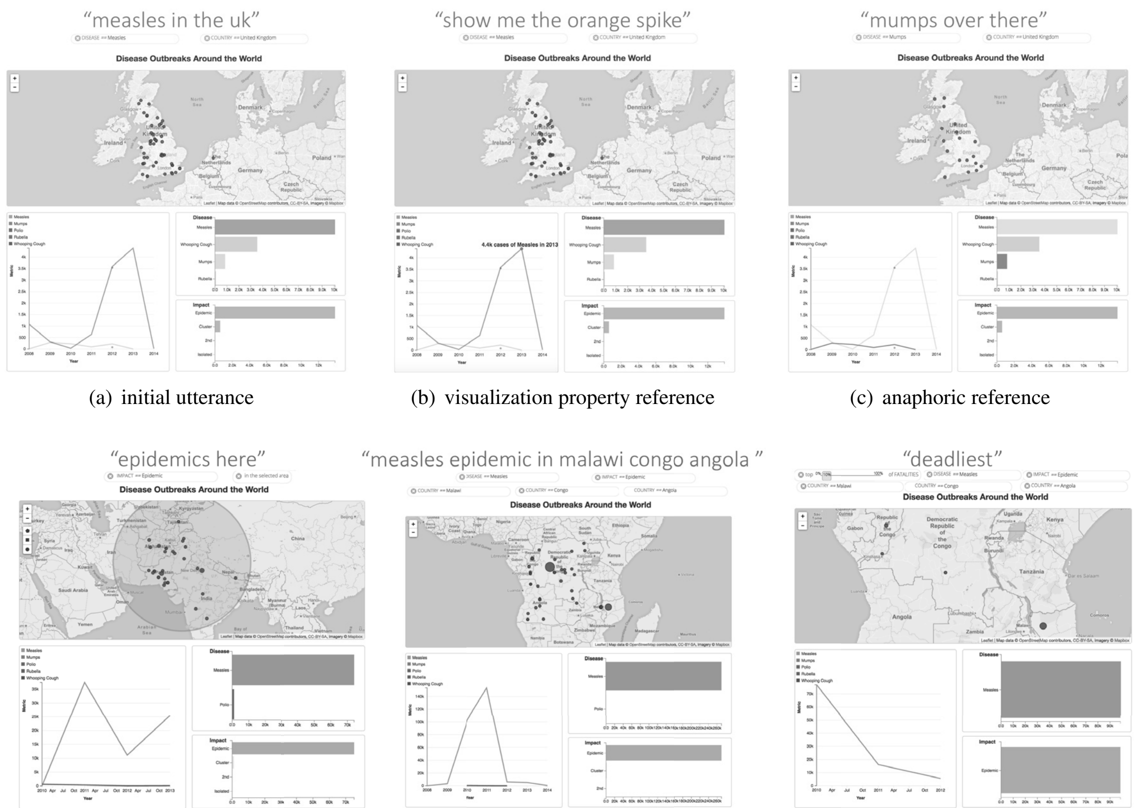
The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government. ☒ No.
☐ Yes, the name of the U.S. Government agency and the Government contract number are:

Signature: / David V. Sanker /
Date: May 3, 2017

David V. Sanker
Reg. No.: 56,242
Morgan, Lewis & Bockius LLP
1400 Page Mill Road
Palo Alto, CA 94304
Tel: 650-843-4000

Applying Pragmatics Principles for Interaction with Visual Analytics

Enamul Hoque, Vidya Setlur, Melanie Tory, and Isaac Dykeman



(a) initial utterance (b) visualization property reference (c) anaphoric reference

(d) deictic reference (e) conjunction (f) lexical cohesion

Fig. 1. Example results of various forms of natural language interactions with a dashboard using principles from pragmatic conversation structure. Starting with an initial utterance (a), our system *Evizeon* supports references to properties in a visualization (b), within the text (c), and through multi-modal interaction (d). Other forms of interaction include support for longer compound queries (e) and lexical cohesion (f), where a user may use other semantically similar words to describe attributes in the visualization, such as ‘deadliest’ for the data attribute ‘fatalities.’

Abstract—Interactive visual data analysis is most productive when users can focus on answering the questions they have about their data, rather than focusing on how to operate the interface to the analysis tool. One viable approach to engaging users in interactive conversations with their data is a natural language interface to visualizations. These interfaces have the potential to be both more expressive and more accessible than other interaction paradigms. We explore how principles from language pragmatics can be applied to the flow of visual analytical conversations, using natural language as an input modality. We evaluate the effectiveness of pragmatics support in our system, *Evizeon* and present design considerations for conversation interfaces to visual analytics tools.

Index Terms—natural language, interaction, language pragmatics, visual analytics, ambiguity, feedback

1 Introduction

A well accepted principle in visual analytics is the need to support interactive exploration and iterative view refinement. A single static visualization is rarely sufficient except in the simplest of investigative tasks. The user often needs to interact with their data, iteratively evolving both the questions and the visualization design. Our research explores *natural language interaction* as a complementary input modality to traditional mouse and touch based interaction for visual analytics.

Direct manipulation is an effective interaction technique when one can easily point to the objects of interest (*e.g.*, lassoing a cluster of points). However, mouse and touch interaction can be inefficient when the interface requires many steps to complete a task. Moreover, this form of interaction gets in the way when users cannot translate their

- *Enamul Hoque is with University of British Columbia. This work was done when Enamul was an intern at Tableau Research. E-mail: enamul@cs.ubc.ca.*
- *Vidya Setlur is with Tableau Research. E-mail: vsetlur@tableau.com.*
- *Melanie Tory is with Tableau Research. E-mail: mtory@tableau.com.*
- *Isaac Dykeman is with Rice University. This work was done when Isaac was an intern at Tableau Research. E-mail: ijdykeman@gmail.com.*

data related questions into data attributes or visual variables to manipulate [18]. In contrast, natural language (NL) interaction can offer numerous advantages in terms of ease of use, convenience, and accessibility to novice users, facilitating the flow of analysis for novices and experts alike [18,35]. These make compelling arguments to investigate NL as a complementary interaction modality for visual analytics.

However, NL interaction techniques for visualization are in their infancy, and existing tools (*e.g.*, DataTone [16], Articulate [25]) largely follow a single query - response paradigm, with some facility to correct system misunderstandings. While promising, these systems do not really support a "cycle of visual analysis"; an interface that requires isolated independent queries cannot be expected to support the fluid iterative exploration and refinement that we expect in visual analytics. Our work introduces new techniques for NL interaction, based on a *conversational interaction model*.

Figure 1 shows results of natural language interaction using a conversational approach, where the user has a back-and-forth exchange with our system. The first query in Figure 1a, "measles in the uk" causes all charts to highlight or filter to cases of measles in the United Kingdom. Here 'uk' is not in the dataset, but is interpreted as an abbreviated place name and automatically matched to the data value 'United Kingdom' using a geographic corpus. The user then types "show me the orange spike" (Figure 1b); the system understands that this is a reference to the visual properties of the line chart and adds detail information to the spike in the line. In Figure 1c, the system interprets "mumps over there" as containing a reference to the previous location and a different value in the *disease* attribute. It retains the filter on 'United Kingdom' but updates *disease* from 'measles' to 'mumps'. "Epidemics here" (Figure 1d) is a reference to marks selected on the map with a mouse, so epidemic diseases in that selected region are highlighted. In Figure 1e, "measles epidemic in malawi congo angola" illustrates a conjunctive query involving multiple search criteria. Finally, the user asks for "deadliest" (Figure 1f). Here the word 'deadliest', even though not present anywhere in the dataset, is matched to top 10% of *fatalities* using a semantic similarity match with external knowledge corpora. Throughout this exchange, the user has been able to build on their prior queries and adapt the current system state, rather than starting over each time with a fully qualified input statement.

The interaction sequence in Figure 1 involves a back-and-forth information exchange akin to human conversations. In human conversation, a turn is identified as a basic unit of dialog, denoted as an *utterance* [20]. Human utterances are very often incomplete or imprecise, relying on the listener to interpret using their contextual knowledge (speaker, topic, time, location, past utterances, etc.). These tendencies carry over into interactions with a visualization, where it is known that people use ambiguous language and partial specification, and may refer to items in their past statements [16,18,35].

The importance of supporting idiosyncrasies of human language became blatantly apparent to us in our own previous work on *Eviza* [35], an early natural language interface to visualizations. Based on the experience of users in our study, plus an analysis of over 3000 example input utterances, we identified the need to support synonyms, compound queries with multiple criteria, queries that depend on prior queries, pronouns, references to visual properties like mark size and color, and multimodal input. Eviza supported only a few of these language characteristics and only with a very simple model. For example, it supported synonyms only through stemming or partial string matching. Similarly, people appreciated being able to enter follow-on queries, but system behavior in these situations was often unexpected, pointing to a need to better understand when the system should remember information from past queries and when it should start fresh.

An effective NL system needs to infer semantics and missing details by using contextual knowledge, providing feedback so the user knows how the system has interpreted their statements. In visual analytics, relevant context that the system needs to track primarily consists of understanding the semantic properties of the data set (attributes and values currently in play) and characteristics of the visualization (visual properties and data encodings). Our work exploits an understanding of human conversations as a basis for algorithms that can infer such contextual information. *Pragmatics* is a term used in linguistics to determine reference of utterances through context of use [19]. We must interpret input in the *context* of the current system state and the user's recent interactions. Furthermore, we must consider all possible contexts that the user might intend and enable the user to correct misinterpretations. To realize a pragmatic language for interacting with visual analytics, we utilize and extend a model commonly used in linguistic conversational structure [20]. We leverage relationships between entities and analytical actions that exist between utterances in conversation.

Inspired by Pokémon's etymology for name origins [5], we call our new system *Evizeon*, with 'eon' meaning evolution. Like Eviza, Evizeon enables natural language interaction with visualizations. However, Evizeon introduces a series of techniques to support conversational pragmatics, deeply enriching the interaction experience.

1.1 Contributions

Realizing interactive conversations like in Figure 1 requires understanding language pragmatics and adapting conversational interfaces to data analytics. Unlike general search interfaces, visual analytics tools can take advantage of their knowledge of data attributes, values, and data related expressions to do a better job of inferring a user's meaning. Towards this end, the contributions of our paper are as follows:

- First, we introduce a theoretical framework based on pragmatics that can be used to improve natural language interaction with visual analytics. We propose an extension to the centering approach employed in pragmatics theory, to support inter-sentential transitional states of continuing, retaining, and shifting the context of the data attributes in play.
- Second, we demonstrate techniques for deducing the grammatical and lexical structure of utterances and their context. Based on our framework, we support various pragmatic forms of natural language interaction with visual analytics. These include understanding incomplete utterances; referring to entities within the utterances and visualization properties; supporting long, compound utterances; identifying synonyms and related concepts; and 'repairing' responses to previous utterances.
- Third, we provide appropriate visualization responses either within an existing visualization or when necessary, by creating new visualizations. We support pragmatic ambiguity through targeted textual feedback and ambiguity widgets.
- We then validate the usefulness of language pragmatics in visual analytic conversations with a user study.

2 Related Work

Designing natural language interfaces can be challenging as they successfully need to interpret unconstrained input [36]. Such systems must be able to process utterances, often using deep expert modeling to extract information necessary for an appropriate interpretation. When users stray outside the supported domain, the system must still be able to respond to a broad range of plausible inputs to maintain an optimal user experience. The Persona project was one such interface, but with a limited task domain of selecting music from a database based on a few hundred variations of a dozen or so basic requests [10]. There has also been a body of research focusing on conversational interfaces, deducing human intent through gaze, turn-taking and dialog structure [9,12,15].

More recently, natural language interfaces for data analysis have emerged as a promising new way of interacting with data and performing analytics. This approach is promising in maintaining conversational flow, as users may be able to express their questions more easily in natural language rather than translating them to system commands. However, existing commercial systems [3,4,8] have fundamental limitations. Most return a minimally interactive visualization in response to queries, meaning the answer needs to be exactly correct rather than approximate. Many require experts to perform modeling before the systems are effective. None are richly integrated with a self-service analysis tool in a manner that allows natural language interactions to become part of a richer visual cycle of analysis. Research systems have similar limitations. RIA explored geo-referenced data on a map with simple queries [38]. Articulate generated visualizations based on simple natural language queries with limited pragmatics and feedback [25].

Studies show that systems where users are expected to always employ syntactically and semantically complete utterances can often be frustrating [11]. Constraining human-system communication to only a subset of utterances reduces users' attention to analytical goals by forcing them to concentrate on the preciseness of the utterances. DataTone [16] improved analysis flow by guessing the user's intent, producing a chart according to that guess, and then providing ambiguity widgets through which the user could adjust settings if the system's guess was incorrect. Eviza [35] was a first step towards supporting simple pragmatics in analytical interaction. The system used contextual inferencing for supporting pragmatics, wherein context established by the preceding dialog is used to create a complete utterance [32]. A related system, Analyza [14], similarly enabled follow-up data queries, but without a visualization focus. These systems recognized the importance of providing feedback on how the system interprets queries and enabling users to correct misunderstandings.

While the use of pragmatics helps with analytical flow, investigations into this approach so far have been very preliminary. To be truly interactive, these systems need richer support for understanding queries based on syntactic and semantic language structure, particularly tied to the analytical properties of the questions. We also need better criteria for deciding when to remember information from prior queries, and enhanced flexibility for users to correct poor system choices. In this work, we explore how a pragmatics-based approach can enable flexible interactions with data that support the flow of visual analysis.

3 Pragmatics

Interaction with visual analysis is most effective when users can focus on answering the questions they have about their data, rather than focusing on how to operate the interface to the analysis tool. Pragmatics is particularly important for visual analysis flow, where questions and insights often emerge from previous questions and patterns of data that a person sees. We apply principles of pragmatics by modeling the interaction behavior as a conversation.

Conversations are more than mere sequences of utterances. For a sequence of utterances to be a conversation, it must exhibit *coherence*. Coherence is a semantic property of conversation, based on the interpretation of each individual utterance relative to the interpretation of other utterances [37]. In order to correctly interpret a set of utterances, we utilize and extend a model commonly used for discourse structure called *conversational centering* [20]. In this model, utterances are divided into constituent discourse segments, embedding relationships that may hold between two segments. A center refers to those entities serving to link that utterance to other utterances in the discourse. Consider a discourse segment DS with utterances $U_1 \ldots U_m$. Each utterance U_n ($1 \leq n < m$) in DS is assigned a set of forward-looking centers, $C_f(U_n, DS)$ referring to the current focus of the conversation; each utterance other than the segment's initial utterance, is assigned a set of backward-looking centers, $C_b(U_n, DS)$. The set of backward-looking centers of a new utterance U_{n+1} is $C_b(U_{n+1}, DS)$, which is equal to the forward-looking centers of U_n (*i.e.*, $C_f(U_n, DS)$). In the context of visual analytic conversations, forward and backward-looking centers consist of data attributes and values, visual properties, and analytical actions (*e.g.*, filter, highlight).

Each discourse segment exhibits both global coherence *i.e.*, the global context of the entire conversation, usually referring to a topic or subject of the conversation, and local coherence *i.e.*, coherence amongst the utterances within that conversation. Local coherence refers to inferring a sequence of utterances within a local context through transitional states of *continuing*, *retaining*, and *replacing* between $C_f(U_n, DS)$ and $C_b(U_n, DS)$. We extend this conversational centering theory for visual analytical conversation by introducing a set of rules for each of these local coherence constructs.

Given an utterance U_n, Evizeon responds by executing a series of analytical functions derived from the forward-looking centers $C_f(U_n, DS)$. Here, an analytical function $F(X, op, v)$ consists of a variable X which can be an attribute or a visualization property, an operator op, and a value v (usually a constant). For example, when the user says "measles in the uk," the system creates two functions namely `F_CAT(diseases, ==, measles)` and `F_CAT(country, ==, uk)`. As the user provides a new utterance U_{n+1}, the system first creates a set of temporary centers $C_{temp}(U_{n+1}, DS)$ from U_{n+1} without considering any previous context. We then apply the following set of rules to create a set of forward-looking centers, $C_f(U_{n+1}, DS)$ based on some set operations between $C_b(U_{n+1}, DS)$ and $C_{temp}(U_{n+1}, DS)$. These forward-looking centers are then used by Evizeon to respond to the user utterance:

Continue: This is a transition that continues the context from the backward-looking center to the forward-looking one. Hence, $C_b(U_{n+1}, DS) \in C_f(U_{n+1}, DS)$, along with other entities.

This transition occurs when a variable X is in $C_{temp}(U_{n+1})$ but not in $C_b(U_{n+1}, DS)$. In this case, the system performs the following union operation: $C_f(U_{n+1}, DS) = C_b(U_{n+1}, DS) \cup C_{temp}(U_{n+1}, DS)$.

Retain: This transition retains the context from the backward-looking center in the forward-looking one *without* adding additional entities to the forward-looking one. $C_b(U_{n+1}, DS) = C_f(U_{n+1}, DS)$.

This transition triggers when the variable X is in $C_b(U_{n+1}, DS)$ but not in $C_{temp}(U_{n+1}, DS)$.

Shift: In this transition, the context shifts from the previous one, with $C_f(U_{n+1}, DS) \neq C_b(U_{n+1}, DS)$.

This transition occurs when the variable X is in both $C_b(U_{n+1}, DS)$ and $C_{temp}(U_{n+1}, DS)$ but the corresponding values are different. In this case, the system replaces all the backward-centers $C_b(U_{n+1}, DS)$ containing X with $C_{temp}(U_{n+1}, DS)$. This transition also occurs when a filter constraint is removed; *e.g.*, removing a widget for *measles* shifts the `disease` variable from *measles* to *all diseases*.

Referring to Figure 1, we illustrate the use of these different types of transition rules in this analytical conversation snippet between a user, Sara and our system, Evizeon:

SARA: measles in the uk. C_f={measles, uk}
EVIZEON: [*Applies categorical and spatial filters showing measles in the UK.*]
SARA: show me the orange spike. **CONTINUE**
C_b={measles, uk}, C_f={measles, uk, orange spike}
EVIZEON: [*Highlights spike in the line for measles in the line chart.*]
SARA: mumps over there.
RETAIN C_b={uk}, C_f={uk}
SHIFT C_b={measles, orange spike}, C_f={mumps}
EVIZEON: [*Retains spatial filter for UK, updates categorical filter to mumps, removes highlighted spike in the line for measles.*]
SARA: measles epidemic in malawi congo angola.
SHIFT C_b={mumps}, C_f={measles, epidemic}
SHIFT C_b={uk}, C_f={malawi, congo, angola}
EVIZEON: [*Applies categorical filter for measles epidemics, applies new spatial filter on Malawi, Congo, and Angola, replacing UK.*]

The global coherence of the analytical conversation is updated when the user either moves to a different dataset or resets the visualization, clearing all previous states. We model the global and local coherences of the conversation states with a finite state machine; updating each state with intentional relationships based on these transitional behaviors [33].

4 Forms of Pragmatic Interaction

Conversation centering posits that utterances display connectedness between them. The manner in which these utterances link up with each other to form a conversation is *cohesion*. Cohesion comes about as a result of the combination of both lexical and grammatical structures in the constituent phrases. Identifying phrase structure is a logical starting point to resolve that utterance into one or more analytical functions applied to the visualization. A probabilistic grammar is applied to provide a structural description of the input queries, similar to the approach in Eviza [35]. We deduce additional syntactic structure by employing a Part-Of-Speech (POS) Tagger [27]. Entities from the parsed output are resolved to corresponding categorical and ordered data attributes [31]. By applying our framework on conversation structure, we demonstrate techniques for supporting various forms of pragmatics in analytical conversation [21].

4.1 Ellipsis

Incomplete utterances are common in conversation. Ellipses are syntactically incomplete sentence fragments that exclude one or more linguistic elements. Often, these utterances cannot be understood in isolation, but rather with previously established context.

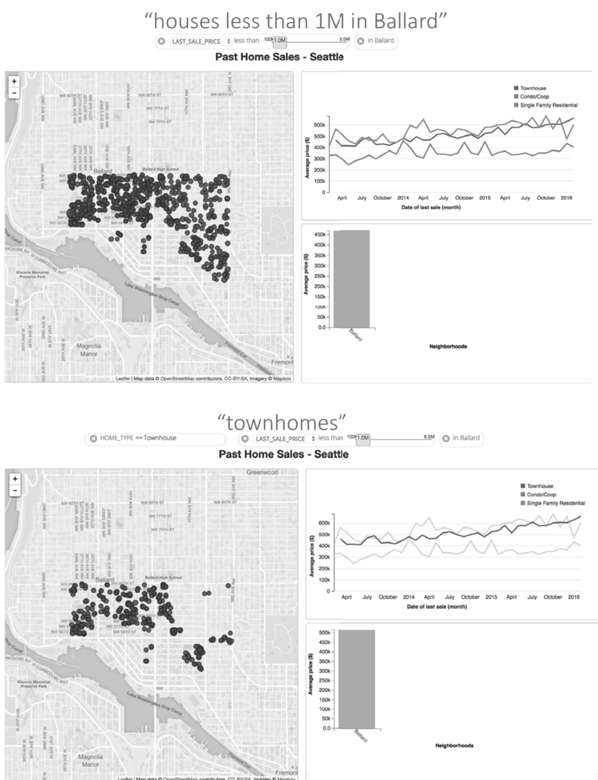


Fig. 2. Results of an ellipsis implementation in Evizeon. Here, the incomplete utterance "townhomes" is parsed with context from the previous utterance to show townhomes under $1M in the Ballard neighborhood.

The conversation below shows how an incomplete utterance "townhomes" is understood in the context of the previous utterance, and is shown in Figure 2.

JOHN: houses less than 1M in Ballard.
C_f={houses, ballard, 1M}
EVIZEON: [*Applies numerical and spatial filters showing houses under $1M in Ballard.*]
JOHN: townhomes.
RETAIN C_b={1M, ballard}, C_f={1M, ballard}
SHIFT C_b={houses}, C_f={townhomes}
EVIZEON: [*Retains numerical and spatial filter for Ballard, applies categorical filter on home_type to show only townhomes.*]

The system applies the rules described in Section 3 to interpret the incomplete utterance "townhomes." The omitted criteria 'less than 1M in Ballard' are retained and the value 'townhomes' replaces 'houses.'

4.2 Referencing

Referring expressions help to unify the text and create economy, preventing unnecessary repetition. Halliday and Hassan state that referencing is a conversation form, which instead of being interpreted semantically in its own right, makes reference to something else for its interpretation [21].

When the interpretation is within the text, this is known as *anaphoric* referencing. In visual analytics interaction, the reference pertains to data attributes and analytical functions. The conversation below shows how John references the year '2015' when asking for prices in the year 2014. The system computes the date for 'previous' using a temporal function DATECALC (Figure 3).

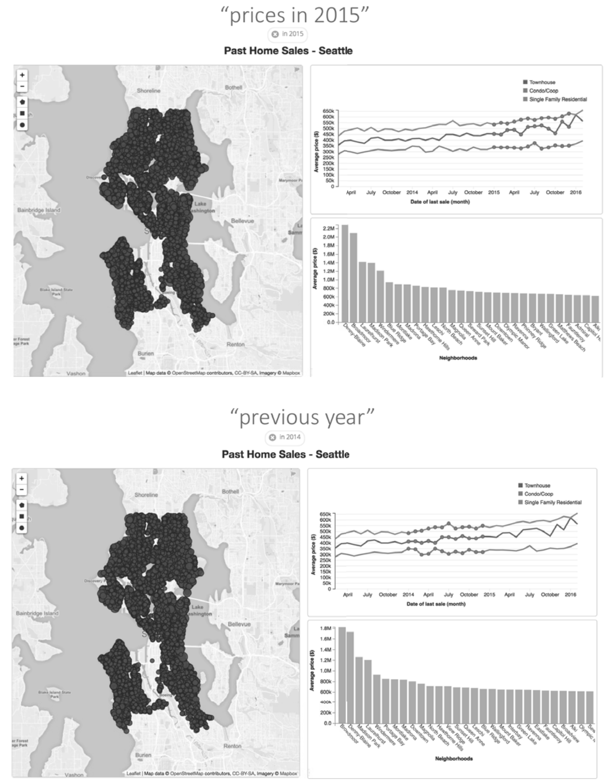


Fig. 3. Results of an anaphoric reference implementation in Evizeon, where the user refers to the year '2014' with the utterance "previous year."

JOHN: prices in 2015. C_f={prices, 2015}
EVIZEON: [*Applies temporal filter showing home prices in the year 2015.*]
JOHN: previous year.
SHIFT C_b={2015},
C_f={DATECALC('year', -1, C_b={2015})$\rightarrow$2014}
EVIZEON: [*Retains a reference to 2015 to update the temporal filter to 2014.*]

We first identify the anaphora in the given utterance, such as 'that', 'those', 'them', 'ones', 'previous', 'next.' We then look at the phrasal chunk containing that reference to identify what entities it is referring to. Next, we search through the backward-looking centers $C_b(U_n, DS)$ to find such entities and replace the anaphoric reference with these entities. After an anaphoric resolution is performed, we apply the rules for updating the forward-looking centers as described in Section 3. For instance, in the above conversation Evizeon identifies that 'previous' is followed by 'year', therefore it finds the value of year in $C_b(U_n, DS)$. Consider another example, "Show fremont, queen anne, and ballard" followed by "condos in *those* districts"; here *those* is referring to some values (*i.e.*, fremont, queen anne, and ballard) of the attribute `neighbourhood` as indicated by the word `districts`.

Note that the references may not always refer to values of a data attribute; they may refer to actions that need to be executed by the system. For instance, consider the utterance "filter out ballard" followed by "do *that* to fremont." Here, *that* is not immediate followed any noun but immediately preceded by a verb word 'do' from which we look at the action mentioned in the previous utterance *i.e.*, 'filter out.'

Another form of referencing lies *outside* the text, and in the context of the visualization. Here, the forward-looking center C_f references context within the visualization as opposed to text in the backward-looking center C_b.

This form of indirect referencing is of two types: (1) A *deictic* reference refers to some object in the environment, usually by pointing. We support deictic references by enabling multimodal interaction (mouse + speech/text), as shown in Figure 1d. (2) A *visualization property* reference uses properties in the visualization such as mark properties, text in labels, axes and titles (Figure 1b).

4.3 Conjunctions

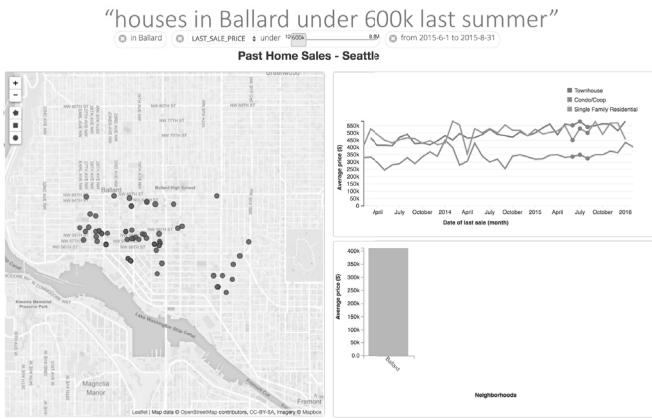


Fig. 4. Evizeon showing the results of an utterance with implicit conjunctions between various attributes for houses in Ballard.

Conjunctions in utterances communicate a range of relations between phrase fragments called conjuncts. In a conversation, people tend to iteratively build a compound query by adding multiple conjuncts as seen in Figure 1d. They often avoid explicit use of conjunctions and connectors, such as 'and', 'or', 'also' between the phrases [24]. For instance, consider this conversation construct for "houses in Ballard under 600k last summer" (Figure 4):

SARA: houses in Ballard. C_f={houses, ballard}
EVIZEON: [*Applies categorical and spatial filters showing houses in Ballard.*]
SARA: houses in Ballard under 600k last summer. **CONTINUE** C_b={houses, ballard}, C_f={houses, ballard, $<600k$, last summer}
EVIZEON: [*Further refines the current visualization by applying a numerical filter on house price and a temporal filter to show the past summer.*]

4.3.1 Linearization

Finding implicit data coherence among the conjuncts, can be a challenging task. In the example above, all the conjuncts refer to the same entity 'houses in Ballard.' However, there are cases where conjuncts map to *different* entities, for instance "houses in Ballard under 600k condos in South Lake Union." The system needs to determine how individual conjuncts resolve to the same or different entities. We devised a rule-based technique that takes a potentially long utterance with possibly implicit conjunctions and translates into a set of analytical functions chained together by logical operators. Evizeon then executes these analytical functions in response to the user utterance.

Multiple conjuncts within these compound utterances need to be resolved to correctly invoke one or more corresponding analytical functions; A process called linearization [23]. As mentioned earlier, an analytical function `F(X, op, v)` consists of a variable X (*e.g.*, an attribute), an operator op, and a value v. Each attribute can be of two types: `categorical` and `ordered` [31]. The `ordered` data type is further categorized into `ordinal` and `quantitative`. The linearization

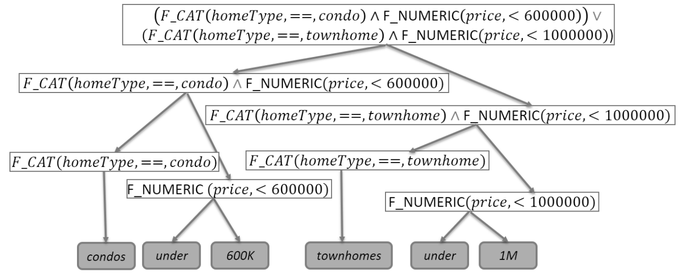


Fig. 5. A parse tree output from Evizeon's parser, showing how the system iteratively connects the analytical functions of adjacent nodes in the parse tree.

process considers the types of attributes and operators to combine analytical functions using logical operators (*i.e.*, $\wedge$, $\vee$) as described below:

Applying the $\vee$ operator: When two or more adjacent conjuncts share an attribute and if that attribute's data type is `categorical`, then these conjuncts are connected by $\vee$. Similarly, if that shared attribute is `ordered` and the function's operator is $==$, we apply $\vee$. Notice that in such cases, $\vee$ is logically appropriate because applying $\wedge$ would not match to any item in the data table.

For example, if the utterance is "show me condos and townhomes," then the system generates the following combination of analytical functions: (`F_CAT(homeType, ==, condo)` $\vee$ `F_Cat(homeType, ==, townhome)`). Here, both 'condo' and 'townhome' belong to the same `categorical` attribute, *i.e.*, *homeType*. Applying $\wedge$ operator would not make sense here because a particular house (item) cannot be both 'condo' and 'townhome' at the same time. Similarly, if the user utters "2 3 bedroom houses", the system generates (`F_ORDINAL(bed, ==, 2)` $\vee$ `F_ORDINAL(bed, ==, 3)`).

The $\vee$ operator is also appropriate if attribute type is `ordered` and involves the condition $X<v_1$ and $X>v_2$, where $v_1<v_2$. For instance, if the utterance is "before 2013 and after 2014", then the $\vee$ operator will be used between the two conjuncts. Again, here applying the $\wedge$ operator would result in matching no item in the data table.

Applying the $\wedge$ operator: The $\wedge$ operator is appropriate if attribute type is `ordered` and involves the condition $X>v_1$ and $X<v_2$, where $v_1<v_2$. For example, "houses over 400k and under 700k" resolves to (`F_NUMERIC(price, >, 400000)` $\wedge$ `F_NUMERIC(price, <, 700000)`). "Beds between 2 to 4" resolves to (`F_ORDINAL(beds, >=, 2)` $\wedge$ `F_NUMERIC(beds, <=, 4)`). Notice that applying $\vee$ operator would result in matching to all items in the data table, which would not make sense.

Finally, the $\wedge$ operator is applied when there is no common attribute between two conjuncts. For example, the utterance "price under 600k with 2 beds" resolves to (`F_ORDINAL(beds, ==, 2)` $\wedge$ `F_NUMERIC(price, <=, 600000)`).

In order to generate the analytical function representation of the whole utterance, we traverse the corresponding parse tree (generated by the parser described in [35]) in post-order and apply the above two rules iteratively on the phrases as illustrated in Figure 5. Here, the system takes the utterance "condos under 600K townhomes under 1M" as input, and iteratively applies the above rules to generate the chain of analytical functions.

4.4 Lexical Cohesion

The previous three types of pragmatics - ellipsis, referencing, and conjunction, provide *grammatical cohesion* to the conversation. In addition to these grammatical constructs, people often find ways for expressing concepts through related word meanings, *i.e.*, senses in conversation, a term called *lexical cohesion* [30]. These word senses can be as simple as spelling, stemming and plurality variations (*e.g.*, 'profit' and 'profits'), synonyms (*e.g.*, 'country' and 'nation'), to related or co-occurring terms (*e.g.*, 'violence' and 'crime').

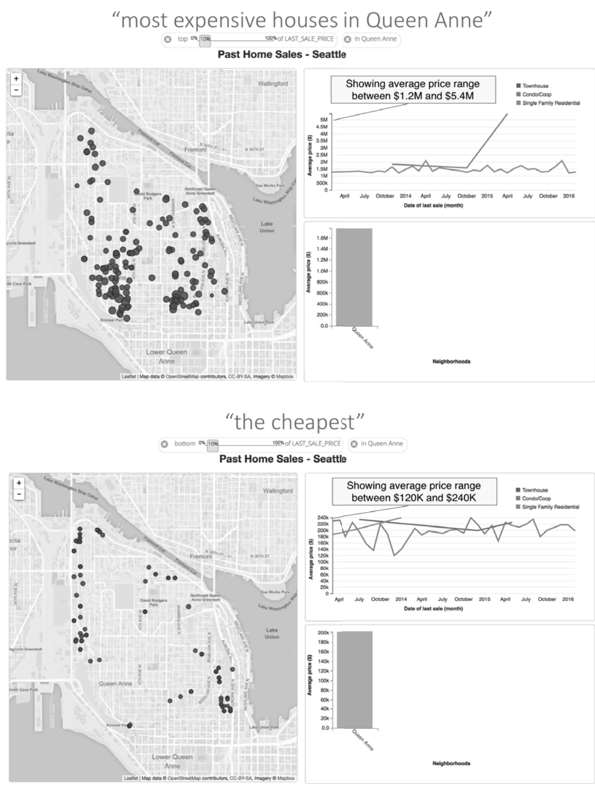


Fig. 6. An implementation of lexical cohesion in Evizeon, where ‘most expensive’ is mapped to the analytical function `Top-N(sale_price)` of houses, and ‘cheapest’ is mapped to `Bottom-N(sale_price)`. Price ranges annotated in yellow for clarity.

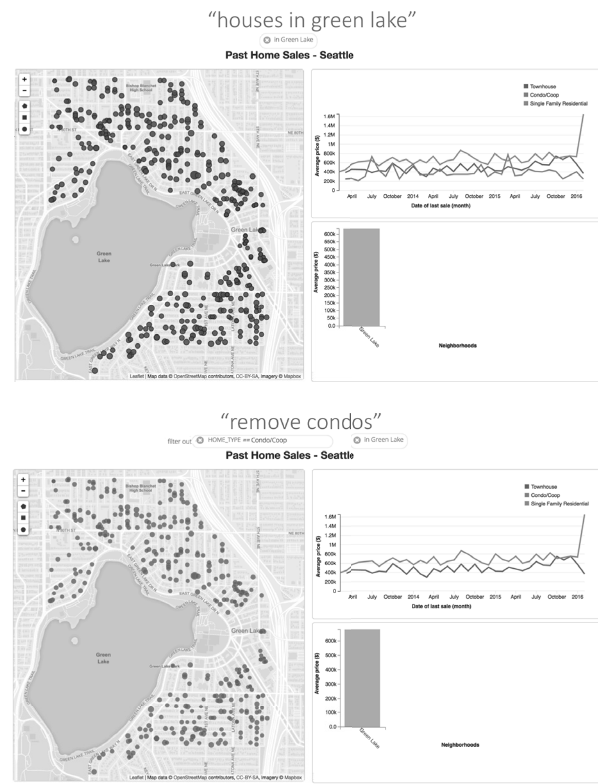


Fig. 7. Evizeon parses the repair utterance “remove condos” and updates the results from the previous figure, filtering out condos.

Often word senses are related to each other within a semantic context [29]. We identify attribute word senses by employing the *word2vec* model containing learned vector representations of large text corpora [28]. We compute word vectors using a recurrent neural network [22]. The semantic relatedness S_{rel} between a word w_i in a given utterance and a data attribute d_j, is the maximum value of a score computed as follows:

$$S_{rel}(w_i, d_j) = max_{m,n} \lambda \cos(v_{w_i}, v_{d_j}) + (1-\lambda) \frac{1}{dist(S_{i,m}, S_{j,n})} \quad (1)$$

where $dist(S_{i,m}, s_{j,n})$ is the Wu-Palmer distance [39] between the two senses $S_{i,m}, S_{j,n}$. v_{w_i}, v_{d_j} are the vector representations of w_i and that d_j respectively. λ is a weighting factor applied to a pairwise *cosine* distance between the vectors.

Other natural language queries that users might ask are “show me the *cheapest* houses near Ballard” or “where are the *mansions* in South Lake Union?” We not only have to compute semantic relatedness between these terms and data attributes, but also compute the type of *analytical function* associated with each term. We consider the corresponding dictionary definitions [1] as additional features to these word vectors, and check if the definitions contain quantitative adjectives such as ‘less’, ‘more’, ‘low’, ‘high’ using a POS tagger. Appropriate analytical functions are then mapped to these adjectives. For example, Figure 6 shows ‘most expensive’ mapping to `Top-N(sale_price)`, ‘cheapest’ to `Bottom-N(sale_price)`. Similarly, Figure 1f shows ‘deadliest’ mapped to the `Top-N` values of the attribute ‘fatalities.’

4.5 Repair Utterances

In a natural conversational flow, it is quite common for people to correct or clarify a previous utterance. In addition to widgets, we also support the use of follow-up *repair utterances* to modify or ‘repair’ a potentially ambiguous utterance or to change the default behavior of how the results are presented to the user. For instance, if the user would like to update the default behavior of the system, such as highlighting for selection, she can use utterances like “no, filter instead”, or she can update attributes (*e.g.*, “get rid of condo” or “change from condo to townhomes”) as shown in Figure 7.

5 Response and Feedback

To support a conversation, the visualizations need to provide cohesive and relevant responses to various utterances. Sometimes the system needs to respond by changing the visual encoding of existing visualizations, while in other cases it is necessary to create a new chart to support the visual analytical conversation more effectively. In addition to appropriate visualization responses, it is critical to help the user understand how the system has interpreted her utterance by producing appropriate feedback and allowing her to rectify the interpretation if necessary. We devised a feedback mechanism that helps the user to interpret the system’s response and subsequently modify the actions made by Evizeon through some interface controls.

5.1 Responses Within Existing Visualizations

In a traditional dashboard, users can interact by selecting items or attributes in a visualization that are highlighted to provide immediate visual feedback [31]. Simultaneously, other charts are updated by highlighting/filtering out items. In a natural language interface, instead of making explicit selection by mouse/keyboard, the user mentions different attributes and values, making it a non-trivial task of deciding how each view within a dashboard should respond to the utterance. Previous natural language interfaces (*e.g.*, Eviza [35]) have not explored this problem as they have only focused on a single visualization.

To decide how the views (V) in a dashboard should respond to the utterance, our approach is as follows. If the items in the results set retrieved by applying the analytical functions are directly encoded in a chart, then Evizeon *highlights* these items. In Figure 1a, the map chart highlights the items that match the criteria “measles in the uk.”

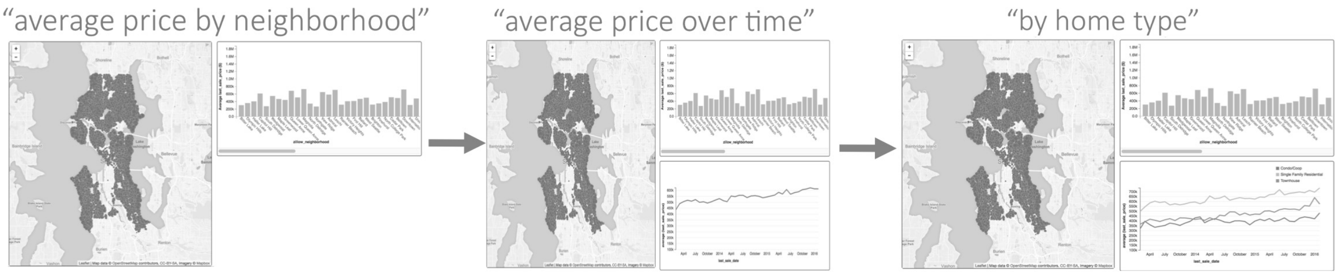


Fig. 8. Evizeon generates new visualizations as part of a coordinated dashboard in response to various input utterances.

However, for a secondary chart that applies further data transformation on the result set (*e.g.*, the line chart and two bar charts), the following rules are applied: The system first creates a list of all attributes $\{X_1, X_2, ...X_m\}$ from the forward looking centers $C_f(U_{n+1}, DS)$. It then invokes the visualization manager to determine if any of these attributes are encoded as dimensions of that visualization V directly (*i.e.*, without using any aggregate functions such as `count` or `average`). If that is the case, it *highlights* the marks related to the corresponding criteria. For example, in Figure 1a , Evizeon highlights the series in the line chart and the bar in the barchart representing 'measles.' However, the bar chart on impact (lower right) cannot highlight any mark because it does not encode any attribute in $\{X_1, X_2, ...X_m\}$. Therefore, it *filters out* the results that do not match the criteria "measles in the uk" and updates the chart accordingly. Note that users can change the default behavior by explicitly expressing the choice about whether to filter vs. highlight (*e.g.*, 'exclude', 'remove', 'filter only').

5.2 Creating New Visualizations

During visual analysis flow, there may be situations where the existing visualization cannot meet the evolving information needs of the user. This scenario could arise when a particular data attribute cannot be encoded effectively in the existing visualization (*e.g.*, time values in a map), warranting the need for creating a new visualization as a response. We draw inspiration from work that connects the visualization and language specification [26, 34]. Our current implementation supports the creation of four different types of visualizations (*i.e.*, bar chart, line chart, map chart, and scatterplot). Figure 8 shows how a dashboard is progressively constructed based on the input utterances.

The underlying algorithm for creating or changing an existing visualization works in following steps: First, the system determines if the creation of a new visualization or change of an existing one is necessary. Evizeon analyzes the attributes specified in the forward-looking centers $C_f(U_{n+1}, DS)$, and searches for any current visualization that encodes these data properties. If there is no match with the specification of existing visualizations, the system generates the corresponding new specification consisting of attributes and aggregation types. We employ a simplified version of the automatic presentation algorithm described in [26] to decide the type of chart generated based on this specification. Finally, the new chart is positioned using a two-dimensional grid-based layout algorithm, automatically coordinated with other views. This updated dashboard responds to subsequent utterances through actions like highlighting or filtering.

5.3 Ambiguity Handling

The interactive dialog provides many new challenges for natural language understanding systems. One of the most critical challenges is simply determining the intent of the utterance. Evizeon automatically attempts to resolve various forms of syntactic, lexical and semantic ambiguities. These resolutions are expressed in the form of widgets and feedback to help the user understand the system's intent and the provenance of how the utterance was interpreted. By manipulating these widgets and viewing the feedback of what results are shown in the visualization, the user can instantiate a follow-up repair utterance to override or clarify the system decisions made.

Widgets are identified from the analytical functions derived from an utterance. An important design consideration here is how can we organize and present the widgets in an intuitive way so that the user can understand how the system interprets her utterance and subsequently modify the interpretation using these widgets. For this purpose, we take the original utterance and order the widgets in the same sequence as the corresponding query terms. We achieve this by using the library Sparklificator that facilitates the placement of small word-scale visualization within text in a compact way [17]. In addition, we provide a set of interactions to users including the ability to manipulate and remove a widget to modify the query and resolve ambiguous ones.

Figure 9 shows how Evizeon presents the widgets for the utterance "condo near Ballard under 1.2M." Here the first term 'condo' was resolved to the widget representing the criteria 'HOME_TYPE equals Condo/coop'. Then, the second widget conveys the fuzzy distance represented by 'near Ballard.' Finally, since 'under 1.2M' does not explicitly mention any attribute, the system determines whether the value 1200000 is within the range of minimum and maximum values of any `numeric` attribute in the data. If such an attribute exists (`LAST_SALE_PRICE` in this case), the system conveys that to the user and then allows her to change the attribute using the drop-down menu.

5.4 Textual Feedback

In addition to ambiguity handling, Evizeon also provides feedback and meaningful hints to modify the text, when it fails to completely understand the query. For instance, if the system cannot successfully parse the given utterance, it first attempts to automatically correct the misspelled terms by comparing the tokens with the attributes, cell values, and related keywords in the current dataset using fuzzy string matching [2]. When, the user forms a query that is partially recognized the system prunes the unrecognized terms from the corresponding parse tree and then shows the results based on the tokens that are understood. Figure 10 presents examples of different possible cases and the corresponding feedback generated by Evizeon.

6 Evaluation

We conducted a study to evaluate the effectiveness of our pragmatics handling approach. Specifically, we aimed to: (1) verify the usefulness of our pragmatics support and feedback mechanisms, (2) identify additional user behavior that our system should handle, and (3) explore individual differences. We decided that an observational study would best enable us to explore pragmatics challenges in an open-ended way. We considered a comparative experiment; however, our prior study of Eviza [35], the most appropriate system for comparison, had already revealed its shortcomings with respect to pragmatics. Specifically, the system often remembered things when participants meant to start over, or forgot things things they hoped it would retain; moreover, we had no good model to understand why.

6.1 Method

6.1.1 Participants

We recruited 21 volunteers (17 males, 4 females, age ranges: 24-35 to 65+) from our institution. All had prior experience with visualization

Fig. 9. Given the utterance "condo near Ballard under 1.2M", the system conveys how it interprets this query by presenting three different widgets.

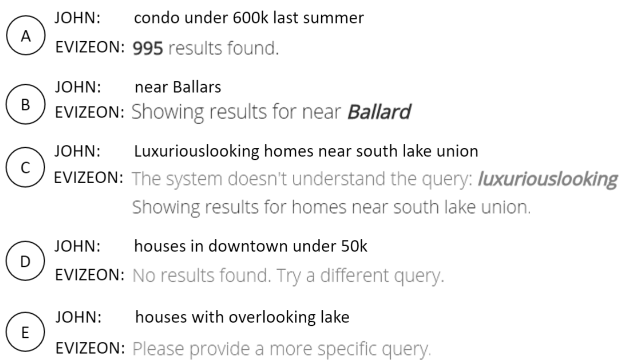


Fig. 10. Examples of feedback implemented in Evizeon: A) The utterance was understood and results were found, B) Not successfully understood, but the system suggested an alternative query, C) The utterance was partially understood and the unrecognized terms were highlighted, D) The utterance was understood but no results were found. E) The utterance was not understood.

and all but 5 regularly used NL interfaces such as Siri and Alexa. All were fluent in English (18 native speakers).

6.1.2 Tasks

We had two types of tasks: target-criteria tasks and open-ended exploration. Each participant completed one task type with one visualization dashboard (either past housing sales or disease outbreaks).

For the target-criteria tasks, we provided target data values and asked participants to manipulate the visualization (through NL or ambiguity widget input) to reveal those data. These were designed as groups of related criteria (3 sets of 4 tasks), to force participants through the various transition types – continuing, retaining, and shifting. We identified four task types: (1) **Add**: new attribute(s) are added to the context (continuing), (2) **Change**: value(s) of existing attribute(s) are changed (shifting), (3) **Remove**: some attribute(s) are removed, retaining others, and (4) **Mixed**: a mixture of (1-3). Each set started with a new query (following a global reset) and each of the remaining tasks transitioned the criteria through one of the task types. To avoid priming participants with specific wording, target criteria were presented as a visual representation (available at [6, 7] and in supplemental material).

Our goal for the open-ended tasks was to observe how people would use pragmatics in an unscripted interaction. For the housing dashboard, participants were asked to imagine a fictional home-buying scenario and then use the visualization to identify suitable neighborhoods. For the outbreaks dashboard, they were simply asked to explore the data.

6.1.3 Procedure and Apparatus

We began with a short introduction to the goals of the study and demonstrated the possible interactions. Participants were instructed to phrase their queries however felt most natural and to interact however felt appropriate. We asked participants to think aloud and to tell us whenever the system did something unexpected. We discussed the participant's reactions to system behavior throughout the session. The session concluded with a semi-structured interview. Three participants completed the study in person and the remainder via videoconference with a shared screen. All sessions took approximately 30 minutes and were recorded. Queries were typed rather than spoken.

6.2 Results

Overall, participants were very positive about NL interaction and identified many benefits. Evizeon allowed participants to focus on their questions rather than how to express them, could react without fully understanding ("*I love this because...it didn't fully understand my query, but what it did get, it actually answered my question!*" [P9]), helped them learn ("*[with the widgets], I have actually exposed and discovered the syntax of this dataset*"), and could save time ("*I can totally see people building a dashboard quick and dirty...literally with 3 questions you've built me a dashboard...you've saved me like 150 clicks right there!*" [P19]). They all appreciated the conversational model enabled by our pragmatics techniques, "*I like that I can just think of a question...and then it allowed me to stay in my workflow and ask additional questions*" [P21] and the ambiguity widgets, "*they're very helpful. First of all because it gives me a way of checking that it got what I want, and secondly it gives me a second way of interacting*" [P16].

6.2.1 Target Criteria Tasks

We observed two distinct strategies that participants took to transition their context in the target criteria tasks:

Edit-in-place: In this strategy, the user repeatedly edits the queries in place with fully qualified utterances, "*I like to know what query I had. I'm actually able to go in and change, like the number of bedrooms, just by editing it. I really like that*" [P21]. An example sequence of utterances for this strategy might be: (1) "single family houses in fremont," (2) "single family houses in fremont and ballard under 900k," (3) "single family houses under 900k."

Replace: In this strategy, the user replaces their queries with terse utterances, often involving ellipsis, while continuing the same train of thought. Example sequence: (1) "single family houses in fremont," (2) "and ballard," (3) "under 900k," (4) "all neighborhoods."

Of the 12 participants who did target criteria tasks, 8 used the edit-in-place strategy and 3 used replace. One participant used a reset strategy, resetting the view to the default state between most tasks.

		Widget input		Text input		Pragmatics use		
	Task	Remove	Adjust	Edit	New	Cmpd	Ellipsis	Reference
Replace	Add	0	0	33	100	100	100	33
	Change	33	33	100	100	67	100	0
	Remove	33	0	0	67	0	33	33
	Mixed	33	0	33	100	67	67	0
Edit-in-place	Add	25	0	63	38	100	0	0
	Change	38	50	100	0	100	0	0
	Remove	25	0	100	13	25	0	0
	Mixed	38	0	88	13	88	0	0

Table 1. Percent of participants in edit-in-place and replace groups who used each interaction type as a primary strategy. Replace participants typed completely new text more often and used more interaction types including ellipsis. Widget manipulation was similar across the two groups, with widget adjustment used only for value change tasks.

Table 1 illustrates how the interaction strategies differed. Only one participant used referencing (specifically, anaphoric); perhaps it was unnecessary or not obvious that it was possible. Both groups occasionally used lexical cohesion (e.g., "SFH" for single family residential or "sold in" for last_sale_date). Evizeon generally worked well for Add and Change tasks: pragmatics behaved as expected and widgets provided effective feedback on how the input had been interpreted. Participants had the most problems on Remove tasks; Evizeon neglected to support text-based removal of items from the context so participants often used widgets as a backup. Table 1 shows that 100% of edit-in-place and 67% of remove participants attempted text-based removal.

Most interesting were the different expectations on how widgets and text should interact, which most often led to trouble on removal tasks. Participants using the edit-in-place strategy all expected the widgets and the current text query to be exactly in sync, both fully documenting the global context state. Therefore, their text-based approach to removal was to delete the text for unwanted items (e.g., last query in the sequence above). Evizeon's failure to remove constraints following such a query was confusing to these participants. Moreover, they also expected that deleting a widget would remove the corresponding text, e.g., "*the only shortcoming was if I made a change in the [widgets], that it would be reflected back in what the text is, so I'm constantly sure what is in control*" [P13] . The lack of synchrony sometimes led to errors; for example, P13 removed a [fatalities = 0] widget, then edited his query text for a different purpose, but the [fatalities = 0] constraint spuriously returned because its text was still present.

Participants using the replace strategy had difficulty conveying whether their queries should be interpreted as new versus continuing the current context. The participant using the reset strategy did so for precisely this reason, "*It's a very funny thing – sometimes you want it to remember and sometimes you don't...And you're very sure about which one you want it to be, but there's nothing about your question that indicates why it would be so.*" P15 often explicitly noted when he wanted the system to maintain context, which he indicated by starting a query with "*now*." Similarly, many participants used keywords to indicate broadening of the context. For example, when moving from a query about measles to any diseases in the same places, one participant typed, "*all diseases there*" (also an example of anaphoric referencing). Another participant used "*worldwide*" in an attempt to remove a country constraint. P16 said that her phrase length should be interpreted as a differentiator, saying, "*it was not in sync with me about when it should drop old stuff and when it should keep old stuff. If I had a more complete sentence, then I meant to wipe the slate clean*."

6.2.2 Open-Ended Tasks

Open-ended tasks demonstrated that the pragmatics functionality was critical. We observed distinct discourse segments containing sets of related utterances for each line of inquiry and 35% of utterances depended on earlier ones in the segment. Maximum segment length ranged from $5-18$ (*avg.*=8.6) and 8 out of 9 participants used utterances that required system memory. Maximum *memory depth* (prior steps necessary to capture all relevant context) ranged from $0-15$ (*avg.*=4.4). References were rarely used (5 instances). Comments included ("*Ha, that's awesome...it is able to take just these additional little snippets*" [P9] and "*I did not think it would understand the word expensive...that blows my mind a little bit!*" [P9]).

Interestingly, none of these participants exhibited the edit-in-place strategy. Four used replace, 2 had an indiscernible strategy, and the remaining 3 used a mixture of replace and edit-in-place. For example, P19 frequently used short follow-on queries to continue his current train of thought, but typed a longer sentence (often without an explicit reset) when starting a new line of inquiry.

The open-ended tasks also revealed a number of system shortcomings that led to unsuccessful queries and user frustration. Unsuccessful queries most often asked for unsupported functionality. For instance, P8 focused on time-related questions, for which we had minimal analytics support. Domain-specific language was an interesting challenge; for example, the system could not relate disease *prevalence* to the numeric attribute *cases*. Another interesting case was the word *fatalities*, the name of a numeric attribute in our outbreaks data. P18 frequently tried to use this word without any numeric constraints because the word itself implies a non-zero state (e.g., "countries with fatalities" rather than the more complete "countries with fatalities > 0"). P9 requested an ability for users to teach the system their own language; he particularly wanted to ask Evizeon about Seattle's "crustiest" neighborhoods.

7 Discussion and Future Work

Our study results suggest that pragmatic approaches to natural language input present a promising approach for engaging people in the flow of visual analysis [13]. Even P7, who was a database expert familiar with SQL, stated that she preferred natural language interaction because she could just focus on her questions rather than how to express them. We were pleased to see 3 participants really get into the task; these participants began telling us all their insights about the data rather than what we had asked for – their experience with the interface.

Presence of edit-in-place and replace strategies presents an interesting interface design challenge, as the two strategies require different interface behavior. Edit-in-place depends on continual presence of an editable text phrase representing the full context, whereas such text is likely to get in the way for the replace strategy. Future work could explore interfaces that accommodate both user strategies as well as interfaces that can learn from and adapt to the strategy. The difference in strategies between open-ended and target criteria tasks also merits further investigation. The lack of any pure edit-in-place approaches for the open-ended task could reflect either individual variation or the task intervention itself. Target-criteria tasks had the advantage of ensuring participants experienced all transition types, but represented an artificial data analysis experience, possibly influencing strategy.

The pragmatics techniques however, do have some limitations. Our rules for understanding how to transition the context appear to be generally correct, but incomplete. Observations of several participants suggest that there may be additional cues we can exploit to better understand user intent. In particular, future work can explore how query length or keywords like *now* might be used as predictors for continuing the prior context versus starting over.

The study also reveals the need for improved coordination between the input text and widgets. We would like to enhance the feedback mechanism further by introducing mixed-initiative approaches, where the user will be able to provide additional information to help the system learn more personalized context. For instance, if the user says "Show me houses near my kid's school" and the system fails to understand "kids's school," the user could point out that location in the map. Providing feedback mechanisms for specifying data aliases on the fly could also help improve the experience.

While the response and feedback generated by our system are useful to users, it would be worth exploring insights from the data itself. The ability to understand natural language seems to set an expectation of general intelligence; as stated by P14, "*I wanted it to do analysis for me...find something interesting...teach me something*." Spatial and temporal analytics were the most frequently requested features (e.g., temporal trend analysis and spatial aggregation at different levels of detail). Narrative summarization of the visualization responses in the textual feedback could further benefit the analytical process. Also, statistical analysis could enhance the utility of labels shown for utterances such as "show me the spike."

The synergies between pragmatics, discourse and visual analytical interaction is promising, with implications for several interesting research directions. Personalized pragmatics derived from historical user interactions and context could provide richer models for intent and flow. Facilitating domain knowledge and advanced analytical functions could help broaden the repertoire of utterances supported. The prevalence of analytical tools on mobile devices opens up new opportunities to support forms of pragmatic interaction unique to spoken dialog.

8 Conclusion

We present *Evizeon*, a system that implements various types of pragmatic interaction with visual analytics. Such interaction requires understanding the pragmatics of human language and intent, and adapting the context to the domain of data analytics. We extend the centering approach employed in pragmatics theory, to support transitional states of data attributes, values, and data related expressions unique to visual analytical flow. We demonstrate that such a system is useful when it not only parses the linguistic structure of the utterances, but also effectively addresses inevitable ambiguity through repair utterances, feedback and ambiguity widgets. While human-computer interaction by means of natural language may never match the nuances of human-human interaction, we believe that there are many promising research directions to get us closer to that goal; one conversation at a time.

REFERENCES

[1] Dictionary. http://www.dictionary.com.

[2] Fuse Library. http://fusejs.io/.

[3] IBM Watson Analytics. http://www.ibm.com/analytics/watson-analytics/.

[4] Microsoft Q & A. https://powerbi.microsoft.com/en-us/documentation/powerbi-service-q-and-a/.

[5] Pokémon Etymology. https://pokemondb.net/etymology.

[6] Study trials (housing). https://public.tableau.com/profile/melanie.tory#!/vizhome/Evizeonstudytrials-housingversion/Dashboard1.

[7] Study trials (outbreaks). https://public.tableau.com/profile/melanie.tory#!/vizhome/Evizeonstudytrials-outbreaksversion/Dashboard1.

[8] ThoughtSpot. http://www.thoughtspot.com/.

[9] G. Ball. Mixing scripted interaction with task-oriented language processing in a conversational interface. In *Proceedings of the 4th International Conference on Intelligent User Interfaces*, IUI 1999, pp. 101–103. ACM, New York, NY, USA, 1999.

[10] G. Bell, D. Ling, D. Kurlander, J. Muller, Pugh,D, and T. Skelly. *Lifelike Computer Characters: The Persona Project at Microsoft Research*, pp. 191–222. London:AAAI Press, 1997.

[11] J. G. Carbonell, W. M. Boggs, M. L. Mauldin, and P. G. Anick. The xcalibur project, a natural language interface to expert systems and data bases. *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, 1985.

[12] J. Cassell, T. Bickmore, M. Billinghurst, L. Campbell, K. Chang, H. Vilhjálmsson, and H. Yan. Embodiment in conversational interfaces: Rea. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI 1999, pp. 520–527. ACM, New York, NY, USA, 1999.

[13] M. Csikszentmihalyi. *Flow: The Psychology of Optimal Experience*. Harper Perennial, New York, NY, March 1991.

[14] K. Dhamdhere, K. S. McCurley, R. Nahmias, M. Sundararajan, and Q. Yan. Analyza: Exploring data with conversation. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*, IUI 2017, pp. 493–504, 2017.

[15] R. Fang, J. Y. Chai, and F. Ferreira. Between linguistic attention and gaze fixations inmultimodal conversational interfaces. In *Proceedings of the 2009 International Conference on Multimodal Interfaces*, ICMI-MLMI 2009, pp. 143–150. ACM, New York, NY, USA, 2009.

[16] T. Gao, M. Dontcheva, E. Adar, Z. Liu, and K. G. Karahalios. Datatone: Managing ambiguity in natural language interfaces for data visualization. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software Technology*, UIST 2015, pp. 489–500. ACM, New York, NY, USA, 2015.

[17] P. Goffin, W. Willett, J.-D. Fekete, and P. Isenberg. Exploring the placement and design of word-scale visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2291–2300, 2014.

[18] L. Grammel, M. Tory, and M.-A. Storey. How information visualization novices construct visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):943–952, 2010.

[19] G. M. Green. *Pragmatics and Natural Language Understanding*. Routledge, 2012.

[20] B. J. Grosz and C. L. Sidner. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204, July 1986.

[21] M. A. Halliday and R. Hasan. *Cohesion in English*. Longman, London, 1976.

[22] F. Hill, K. Cho, A. Korhonen, and Y. Bengio. Learning to understand phrases by embedding the dictionary. *Transactions of the Association for Computational Linguistics*, 4:17–30, 2016.

[23] L. Horn and G. Ward. *Handbook of Pragmatics*. Blackwell Handbooks in Linguistics. Wiley, 2004.

[24] A. Knott and T. Sanders. The classification of coherence relations and their linguistic markers: An exploration of two languages. *Journal of Pragmatics*, 30:135–175, 1997.

[25] A. Kumar, J. Aurisano, B. Di Eugenio, A. Johnson, A. Gonzalez, and J. Leigh. Towards a dialogue system that supports rich visualizations of data. In *17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pp. 304–309, 2016.

[26] 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, Nov. 2007.

[27] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pp. 55–60, 2014.

[28] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, NIPS 2013, pp. 3111–3119. Curran Associates Inc., USA, 2013.

[29] G. A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, Nov. 1995.

[30] J. Morris and G. Hirst. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17(1):21–48, Mar. 1991.

[31] T. Munzner. *Visualization Analysis and Design*. CRC Press, 2014.

[32] T. Reinhart. *Pragmatics and Linguistics: An Analysis of Sentence Topics*. IU Linguistics Club publications. Reproduced by the Indiana University Linguistics Club, 1982.

[33] E. Roche and Y. Shabes, eds. *Finite-State Language Processing*. MIT Press, Cambridge, MA, USA, 1997.

[34] A. Satyanarayan, K. Wongsuphasawat, and J. Heer. Declarative interaction design for data visualization. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST 2014, pp. 669–678. ACM, New York, NY, USA, 2014.

[35] V. Setlur, S. E. Battersby, M. Tory, R. Gossweiler, and A. X. Chang. Eviza: A natural language interface for visual analysis. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST 2016, pp. 365–377. ACM, New York, NY, USA, 2016.

[36] T. Trower. Creating conversational interfaces for interactive software agents. In *Extended Abstracts on Human Factors in Computing Systems*, CHI EA 1997, pp. 198–199. ACM, New York, NY, USA, 1997.

[37] T. Van Dijk. *Text and Context: Explorations in the Semantics and Pragmatics of Discourse*. Longman Linguistics Library. Addison-Wesley Longman Limited, 1977.

[38] Z. Wen, M. X. Zhou, and V. Aggarwal. An optimization-based approach to dynamic visual context management. In *IEEE Symposium on Information Visualization*, pp. 187–194, 2005.

[39] Z. Wu and M. Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics*, ACL 1994, pp. 133–138. Association for Computational Linguistics, Stroudsburg, PA, USA, 1994.