

Natural Language Queries for Producing Data Visualizations

Melissa Bica, Ben Cordova, Alex Gendreau, Kevin Holligan

Department of Computer Science

University of Colorado Boulder

{Melissa.Bica, Benjamin.Cordova, Alexandra.Gendreau, Kevin.Holligan}@colorado.edu

ABSTRACT

Tools exist to enable users to filter and visualize their data based on queries; however, these queries often use unfamiliar and unnatural language, making it difficult to see the relation between the query and the resulting visualization. We have designed a tool in which a user can choose from a bank of natural language keywords, which includes data-specific words as well as words used for grammatical correctness, to build a natural language research question or query and be presented with data visualizations that aim to answer the query. We followed a user-centered design process, iterating upon our design based on results from thinking aloud user testing. In this paper, we present relevant work which motivated the project, key requirements and challenges, and our evaluation of how our final design addressed the challenges found via user testing.

Author Keywords

Natural language processing; data visualizations; natural user interfaces; user study.

ACM Classification Keywords

H.5.2 Information interfaces and presentation (e.g., HCI): User-centered design.

INTRODUCTION

We developed an interface which uses natural language processing (NLP) of users' queries to generate data visualizations. The interface takes a natural language query as input and provides the user with a series of potential data visualizations as output. For example, the statement, "What is the number of users over time?" could result in output options of line graphs, bar charts, and other, more complex, forms of data visualization. The interface was designed with two groups of potential users in mind: academic researchers and business/financial analysts. Our system is built to accommodate queries based on a broad range of

research questions, including but not limited to, deductive questions that intend to prove existing knowledge (e.g. "How many people...") and inductive questions that attempt to guide thesis formation (e.g. "How do people make use of...") [3].

REQUIREMENTS

Much work has been done in this area, but fully connecting natural language queries with data visualizations is a developing field. The key challenges to this are creating an intuitive user interface that is not only easy to use, but allows the user to build queries that represent truly natural language. We have built on the current work by designing a more realistic natural language interface which handles data-specific queries. We also incorporated verbal descriptions of data visualizations into the final output to illuminate the relationship between the natural language query and the visualization.

Creating Natural Language Queries

As this system takes a query and interprets the results in a data visualization, the application needs to be able to parse and interpret user input. There are multiple methods of accepting input, including free text fields, drop down boxes as used in databases, and pre-selected keywords that you can drag and drop, among others. Each method of input has strengths and weaknesses in regards to its developmental complexity and freedom of user input. Free text fields, which present the most effective method of allowing the user to describe their own natural language query (as described in the next challenge), also present the most difficult implementation in terms of parsing and validating the input.

On the other end of the spectrum, drag-and-drop keywords and drop-down boxes present a prefixed method of controlling the user input, but also present challenges in how the user interprets the meaning of the keyword and whether that interpretation aligns with their understanding, or method, of how to construct a query. Catalina Hallet explores this challenge in regards to using NLP in database queries and found that using drop down boxes eliminates parsing and translation difficulties found in free text fields while still providing the full range of queries that can be posed to a given database [1].

Describing Visualizations with Natural Language

Another primary challenge in creating this type of system is representing natural language in a way that is meaningful

Paste the appropriate copyright/license statement here. ACM now supports three different publication options:

- ACM copyright: ACM holds the copyright on the work. This is the historical approach.
- License: The author(s) retain copyright, but ACM receives an exclusive publication license.
- Open Access: The author(s) wish to pay for the work to be open access. The additional fee must be paid to ACM.

This text field is large enough to hold the appropriate release statement assuming it is single-spaced in Times New Roman 8-point font. Please do not change or modify the size of this text box.

Each submission will be assigned a DOI string to be included here.

and accurate to all users. Different people use language in different ways, and as a study by Metoyer et al. showed, this is especially true when people use language to describe data visualizations to others [2]. Our intent was to provide users with a way to use keywords to build queries that are not only grammatically correct, but also accurate calls to the database to produce visualizations that answer their intended question. In other words, we wanted to lessen the distance between how a person thinks about a research question in natural language terms and how a database interprets actual queries.

To resolve this, we had to consider both the choice of vocabulary available to users to build the query as well as how the system backend would interpret the query. The first issue is discussed in work by Vuillemot and Akmanalp, in which they found that “the development of a vocabulary for visualizations is very important to both correctly processing natural language queries of visualizations as well as generating verbal description to aid in the interpretation of the visualization...” [4]. We had to consider how many words would be available (such as all the words in the English dictionary), what categories or classes of words (such as nouns, verbs, data-specific words), and whether we wanted users to be able to create complete, grammatically correct sentences/questions or sets of keywords (such as, “What is the number of users over time” as opposed to “number users over time”).

DESIGN

Our final design is a three-paneled interface that contains a panel with a set of keywords, a query builder input field in which keywords can be dragged-and-dropped, and a panel that displays visualizations in a carousel when a valid query

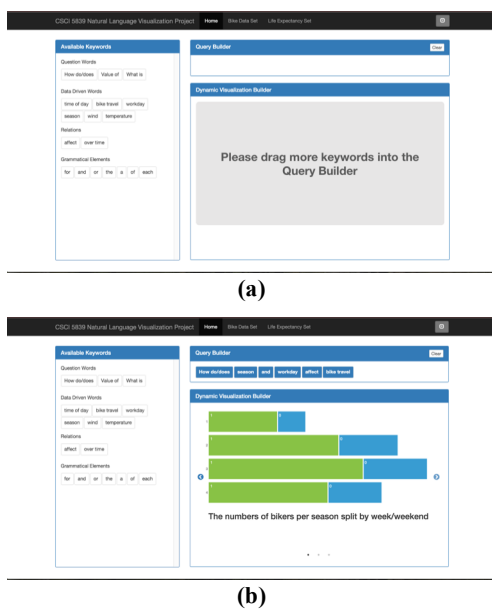


Figure 1. The user interface for our system, shown as is when a user opens it for the first time in (a) and after completing a query and getting data visualizations as output in (b).

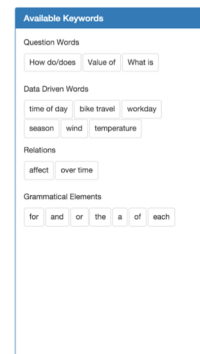


Figure 2. The Available Keywords panel with four groupings of drag-and-drop keywords.

is present (Figure 1).

In the leftmost panel of our design (Figure 2) are available keywords that can be dragged and dropped into the Query Builder (Figure 3). The keywords are categorized into four groups: Question Words, Data Driven Words, Relations and Grammatical Elements. The keywords represent elements of a query that relate to a different sets of images. For example, “over time” signifies that time series visualization should be displayed, and “temperature” references the content of the data. Feedback is provided to the user to help indicate that dragging is the correct action as the cursor changes to a hand/grab icon when hovering over any keyword in the set.

The upper right panel is the Query Builder (Figure 3). This panel has an input field that allows for the placement of dragged and dropped keywords from the keyword panel. The keywords can be dragged and dropped anywhere within the bounds of the input field. The input panel provides feedback to help prevent user error by displaying the text, “Drag keyword here” any time the user has a keyword selected. Additionally, a gray box appears in the field to help indicate to the user where to place the next keyword (albeit, the keyword can be placed at any position in the query). Keywords can also be deleted from the query by dragging and releasing them outside the input field, or simply by clicking the “Clear” button which resets the entire field. Upon completion of a successful query, the keywords that were used to display the generated

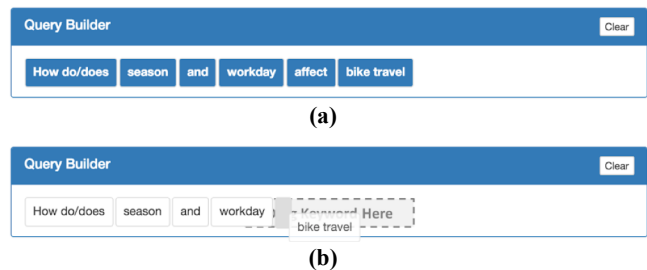


Figure 3. The Query Builder panel, shown while the user is constructing the query in (a) and after the user has completed constructing a valid query, with the relevant keywords in the resulting visualization highlighted in (b).

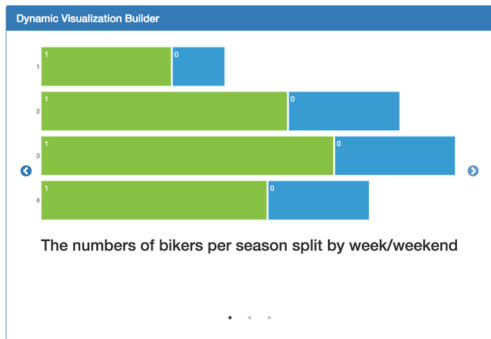


Figure 4. The Dynamic Visualization Builder panel after a valid query has been constructed. This panel uses a carousel to display multiple results, allowing the user to view the other visualizations by clicking on either the left or right arrows or the small dots at the bottom. In this example there are 3 resulting data visualizations, represented by 3 dots.

visualization change color from a white background with black lettering to a blue background with white lettering. This feedback allows for an iterative design by informing the user which keywords influence the generated visualization.

The third and final panel (Figure 4), located in the lower right of the application, is used to display data visualizations as output when the user has generated a valid, complete query. The field also provides feedback to the user by stating to drag keywords into the query builder. After the user successfully places one keyword, the feedback changes to “Please drag more keywords into the Query Builder,” (Figure 1a.) to inform the user that the query is still incomplete. When output is displayed, if there is more than one visualization for the query, arrow keys are displayed to the left and right of the visualization and a carousel appears underneath the visualization (Figure 4). These both indicate to the user how to access the visualization options that are generated as a result of the query.

EVALUATION

Addressing Challenges of Creating Natural Language Queries

This challenge was in regards to manipulating the tool and the methods of user interaction. We opted for drag-and-drop as the method of entering keywords into the query builder. This method limited the full range of queries a user could build as the keywords were pre-populated, either from a global keyword set or a subset specific to the user-uploaded data. A majority of our users found the drag and drop feature intuitive and easy to use. One user commented, “Once you’ve done the first one, it’s really easy to figure out how to build any other query.” As such, we ensured that we made the discovery process for the first query as easy and streamlined as possible; we provided multiple instances of feedback to indicate how the user should interact with

the keywords and where they should be placed, such as “Drag keyword here” in the Query Builder.

However, one of our six users indicated that they would prefer to double click a keyword into the query builder, or at least would like the option to do so, as it is faster than dragging and dropping. Additionally, we made sure that our buttons and headings were labeled in a way that made the object functionality obvious to the user. In our final round of testing, six out of six users were able to correctly identify and use the “Clear” button in order to reset the query builder field.

Addressing Challenges of Describing Visualizations with Natural Language

This challenge was in regards to the difficulties in coming up with an appropriate bank of keywords that any user would be able to use to build a natural language sentence. Our final design has four classes or groupings of keywords in the Available Keywords panel: Question Words, Data-Driven Words, Relations, and Grammatical Elements. From our thinking aloud user testing results, we determined that these groupings were helpful for users for building queries. One user commented that she especially liked the keywords in Relations, and several users liked that the Question Words provided good starting keywords for queries. Another user commented that he liked putting in words like “the,” even though he wasn’t sure if it actually made a difference in the query processing, but it made the query more readable. In general, in our final round of user testing, we received very few complaints from users about not being able to construct grammatically correct queries, and those who did have concerns about this were eventually able to find the specific words they needed to make the query grammatically correct.

On the other hand, a few users commented that “time” was not a relational word in all cases, and that it should actually be placed under Data Driven Words. Similarly, users did not like having to use “over” and “time” separately, and suggested combining those as one keyword under Relations. These data show that we were successful overall in our choice of keyword classes and availability of sufficient keywords for grammatical accuracy, but that we could have improved our design by adjusting some of the individual words to better reflect the classes they were contained in.

CONCLUSIONS

In this paper, we have demonstrated our design of a tool for a user to build natural language queries in order to get resulting data visualizations. We outlined our user-centered design process and how results from thinking aloud user testing influenced our final design. Our evaluation of the final design is that we successfully created a natural and intuitive user interface to accomplish the specified tasks. While some details could be tweaked in response to a few users’ comments, overall we received much positive feedback and are confident in the user-centered design.

REFERENCES

1. Catalina Hallett. 2006. Generic querying of relational databases using natural language generation techniques. In *Proceedings of the fourth international natural language generation conference*. Association for Computational Linguistics.
2. Ronald Metoyer, Bongshin Lee, Nathalie Henry Riche, and Mary Czerwinski. 2012. Understanding the verbal language and structure of end-user descriptions of data visualizations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 1659-1662. DOI=<http://dx.doi.org/10.1145/2207676.2208292>
3. Leysia Palen. 2014. Empirical Epistemologies Applied to Human-Centered Computing Research: A One Page Guide. University of Colorado Boulder, Nov 16, 2014.
4. Romain Vuillemot and Mehmet Akmanalp. 2015. Towards Text Search for Information Visualization Retrieval. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '15)*. ACM, New York, NY, USA, 1821-1826. DOI=<http://dx.doi.org/10.1145/2702613.2732753>